

# DSG Support Multi Solution

## Evolution Of Internet

इंटरनेट दुनिया का सबसे बड़ा नेटवर्क है जो पूरी दुनिया में फैला रहता है. इंटरनेट को हिंदी में 'अंतरजाल' कहते हैं। सरे शब्दों में कहें तो, "इंटरनेट एक वैश्विक (global) नेटवर्क है जिसमें बहुत सारे कंप्यूटर आपस में एक दूसरे के साथ जुड़े रहते हैं. इंटरनेट का आविष्कार 1 जनवरी 1983 को Vint Cerf और Bob Kahn ने किया था. इसलिए Vint Cerf और Bob Kahn को 'इंटरनेट का पिता' भी कहा जाता है। भारत में पहली बार 14 अगस्त 1995 को इंटरनेट लांच हुआ था. तब उस समय इसकी स्पीड बहुत कम हुआ करती थी।

आज हम जिस इंटरनेट को जानते हैं, वह एक विशाल, परस्पर जुड़ा हुआ नेटवर्क है जिसने हमारे संवाद करने, काम करने, सीखने और बातचीत करने के तरीके को बदल दिया है। लेकिन हमेशा ऐसा नहीं था। इंटरनेट की उत्पत्ति 1950 के दशक में हुई थी जब दुनिया शीत युद्ध से जूझ रही थी, और शोधकर्ताओं ने एक ऐसी संचार प्रणाली बनाने की कोशिश की जो व्यवधानों का सामना कर सके। एक सैन्य परियोजना के रूप में अपनी विनम्र शुरुआत से लेकर आज वैश्विक वेब तक, इंटरनेट का इतिहास मानवीय सरलता, सहयोग और कनेक्टिविटी की अथक खोज और नवाचार, सहयोग और तेजी से तकनीकी प्रगति की एक आकर्षक यात्रा की कहानी है।

## इंटरनेट के इतिहास में

- **1969:** ARPANET का शुभारम्भ, पहला परिचालन पैकेट-स्विचिंग नेटवर्क।
- **1971:** रे टॉमलिंग्सन द्वारा पहला ईमेल भेजा गया।
- **1983:** ARPANET के लिए मानक प्रोटोकॉल के रूप में TCP/IP को अपनाया गया।
- **1989-1990:** टिम बर्नर्स-ली ने वर्ल्ड वाइड वेब का आविष्कार किया।
- **1993:** पहला ग्राफिकल वेब ब्राउज़र, मोजेक, जारी किया गया।
- **1998:** गूगल की स्थापना हुई, जिसने सर्च इंजन में क्रांति ला दी।
- **2007:** आईफोन ने व्यापक मोबाइल इंटरनेट पहुंच की शुरुआत की।
- **2010 का दशक:** क्लाउड कंप्यूटिंग, सोशल मीडिया और स्ट्रीमिंग सेवाओं।

## इंटरनेट का संपूर्ण इतिहास: वर्ष-दर-वर्ष विकास

यह वर्ष-दर-वर्ष मार्गदर्शिका युद्ध के दौरान विचार से लेकर 2024 के वर्तमान दिन तक इसके सम्पूर्ण विकास का पता लगाएगी, जिससे आपको इसकी गहरी समझ मिलेगी कि यह कैसे अस्तित्व में आया।

# DSG Support Multi Solution

1940 का दशक: प्रारंभिक नींव और द्वितीय विश्व युद्ध से संबंध

1945

- अमेरिकी इंजीनियर वन्नेवर बुश ने "जैसा हम सोच सकते हैं" शीर्षक से एक लेख प्रकाशित किया, जिसमें *मेमेक्स* नामक एक प्रणाली की कल्पना की गई थी, जो एक सैद्धांतिक उपकरण है जो किसी व्यक्ति की सभी पुस्तकों, अभिलेखों और संचार को संग्रहीत कर सकता है। इस दूरदर्शिता ने हाइपरटेक्स्ट और लिंकड डेटा के विचार के लिए वैचारिक आधार तैयार किया, जो भविष्य के इंटरनेट के महत्वपूर्ण तत्व हैं।
- **कहानी का संबंध** : बुश का काम द्वितीय विश्व युद्ध के दौरान अमेरिकी वैज्ञानिक अनुसंधान और विकास कार्यालय में उनके नेतृत्व से उभरा। युद्धकालीन प्रौद्योगिकी के समन्वय के उनके प्रयासों ने बड़े पैमाने पर सूचना साझा करने के विचारों को प्रेरित किया, जिसने अंततः इंटरनेट के विचार में योगदान दिया।

1950 का दशक: विचारधारा का बीज

1950 – 1958

- **शीत युद्ध का प्रभाव** : जैसे-जैसे अमेरिका और सोवियत संघ के बीच तनाव बढ़ता गया, अमेरिका में तकनीकी हीनता का डर बढ़ता गया। इस माहौल ने तकनीकी नवाचार को बढ़ावा दिया, खास तौर पर रक्षा और संचार के क्षेत्र में।
- **ARPA निर्माण की कहानी** : 1957 में, स्पुतनिक के सोवियत प्रक्षेपण ने अमेरिका को उन्नत अनुसंधान परियोजना एजेंसी (ARPA) बनाने के लिए प्रेरित किया, जिसका ध्यान अत्याधुनिक तकनीक पर था, जो अमेरिकी श्रेष्ठता सुनिश्चित कर सके।

1960 का दशक: नेटवर्किंग अवधारणाओं का जन्म

1962

- **जे.सी.आर. लिक्लिडर का विज़न** : मनोवैज्ञानिक और कंप्यूटर वैज्ञानिक जे.सी.आर. लिक्लिडर ने "इंटरगैलेक्टिक कंप्यूटर नेटवर्क" के बारे में लिखा था। उनके विचारों ने उस चीज़ की नींव रखी जो अंततः इंटरनेट बन गई, जिसमें कंप्यूटरों को एक दूसरे के साथ सहजता से संवाद करने की कल्पना की गई थी।

1965

- **पहला वाइड-एरिया नेटवर्क (WAN) प्रयोग** : एमआईटी में कार्यरत लॉरेंस रॉबर्ट्स ने दूर-दराज के सिस्टम के बीच डेटा संचार की क्षमता का प्रदर्शन करते हुए, एक टेलीफोन लाइन पर कंप्यूटरों को जोड़ा। यह बड़े पैमाने पर नेटवर्किंग की ओर पहला कदम था।

# DSG Support Multi Solution

1969

- **ARPANET का जन्म** : अमेरिकी रक्षा विभाग द्वारा वित्तपोषित, ARPANET दुनिया का पहला पैकेट-स्विचिंग नेटवर्क था। पहला संदेश, "LO" (जिसका अर्थ "LOGIN" था), UCLA और स्टैनफोर्ड रिसर्च इंस्टीट्यूट के बीच भेजा गया था, लेकिन बीच में ही क्रैश हो गया - एक साधारण लेकिन यादगार पल।

1970 का दशक: आधुनिक नेटवर्किंग की नींव

1971

- **पहला ईमेल** : ईमेल का आविष्कार **रे टॉमलिंग्सन** ने किया था , जिन्होंने कंप्यूटर नामों से उपयोगकर्ता नामों को अलग करने के लिए "@" प्रतीक पेश किया था। ईमेल तेजी से ARPANET की सबसे लोकप्रिय सेवा बन गई। रे टॉमलिंग्सन ने पहला ईमेल भेजा, जिसमें कंप्यूटर नामों से उपयोगकर्ता नामों को अलग करने के लिए "@" प्रतीक का चयन किया गया। यह नवाचार डिजिटल संचार की रीढ़ बन गया।

1973

- **अंतर्राष्ट्रीय विस्तार** : ARPANET वैश्विक हो गया जब नॉर्वे में एक नोड जोड़ा गया, जिसने पहला अंतर्राष्ट्रीय कनेक्शन चिह्नित किया और वैश्विक संचार नेटवर्क के लिए मंच तैयार किया।
- **नेटवर्किंग प्रोटोकॉल** की अवधारणा ने **रॉबर्ट काहन** और **विंटन सेर्फ** द्वारा TCP/IP (ट्रान्समिशन कंट्रोल प्रोटोकॉल/इंटरनेट प्रोटोकॉल) डिज़ाइन करने के साथ एक छलांग आगे बढ़ाई। TCP/IP बाद में इंटरनेट के लिए मानक प्रोटोकॉल बन गया।

1974

- **'इंटरनेट' शब्द** : विंटन सेर्फ और **बॉब काहन** ने टीसीपी (ट्रान्समिशन कंट्रोल प्रोटोकॉल) नामक एक नया प्रोटोकॉल प्रस्तावित किया। यह इस अवधि के दौरान था कि "इंटरनेट" शब्द का पहली बार इस्तेमाल किया गया था, जो उनके परस्पर जुड़े नेटवर्क के विचार को संदर्भित करता था।

1976

- महारानी एलिजाबेथ द्वितीय ने अपना पहला ईमेल भेजा, जिससे वे इस उभरती हुई तकनीक का उपयोग करके संचार करने वाली प्रथम राष्ट्राध्यक्षों में से एक बन गईं।

1978

- **टीसीपी/आईपी प्रोटोकॉल का जन्म** : सेर्फ और खान द्वारा विकसित टीसीपी/आईपी इंटरनेट के लिए प्राथमिक प्रोटोकॉल सूट बन गया, जिसने नेटवर्क संचार के लिए एक मानकीकृत विधि की स्थापना की।
- पहला स्पैम ईमेल **गैरी थुर्क** नामक एक विपणनकर्ता द्वारा ARPANET उपयोगकर्ताओं को भेजा गया था, जिसने डिजिटल मार्केटिंग की शुरुआत को चिह्नित किया (हालांकि यह अत्यधिक विवादास्पद था)।

# DSG Support Multi Solution

1980 का दशक: शैक्षणिक नेटवर्क से सार्वजनिक सेवा तक

1983

- **TCP/IP में परिवर्तन** : 1 जनवरी, 1983 को ARPANET ने पूरी तरह से TCP/IP पर स्विच कर दिया, जिससे आधुनिक इंटरनेट का आधिकारिक जन्म हुआ, जैसा कि हम जानते हैं। इस घटना को सभी इंटरनेट संचार की शुरुआत के रूप में मनाया जाता है।
- **डोमेन नाम प्रणाली (DNS)** : पॉल मोकापेट्रिस ने DNS का आविष्कार किया, तथा .com, .org, और .edu जैसे डोमेन पेश किए, जिससे इंटरनेट अधिक उपयोगकर्ता-अनुकूल बन गया।

1984

- **डोमेन नाम प्रणाली (DNS)** की शुरुआत की गई, जिससे “.com,” “.org,” और “.net” जैसे पहचाने जाने योग्य पते बनाए गए। इससे लोगों के लिए वेबसाइट तक पहुँचने का तरीका सरल हो गया, जिससे यह अधिक उपयोगकर्ता-अनुकूल बन गया।

1985

- **प्रथम पंजीकृत डोमेन नाम** : सिम्बोलिक्स.कॉम पहला पंजीकृत डोमेन नाम बन गया, जिसने डोमेन नाम के प्रयोग की शुरुआत को चिह्नित किया जो जल्द ही इंटरनेट नेविगेशन का केंद्र बन गया।

1989

- **वर्ल्ड वाइड वेब की परिकल्पना** : सर्न में काम करने वाले ब्रिटिश वैज्ञानिक टिम बर्नर्स-ली ने एक वितरित सूचना प्रणाली के लिए एक विचार प्रस्तावित किया जो वर्ल्ड वाइड वेब बन गया। उनका लक्ष्य हाइपरलिंक का उपयोग करके इंटरनेट पर दस्तावेजों को जोड़ने और उन तक पहुँचने का एक तरीका बनाना था।

1990 का दशक: वेब और वाणिज्यिक इंटरनेट का उदय

1990

- **WWW कोड जारी** : बर्नर्स-ली ने पहला वेब पेज एडिटर और ब्राउज़र लिखा, और WWW का आधिकारिक तौर पर जन्म हुआ। यह वेबसाइट और वेब पेजों की सुबह थी। बर्नर्स-ली ने पहला वेब ब्राउज़र और वेब सर्वर विकसित किया, जिससे **वर्ल्ड वाइड वेब** के उनके विचार को जीवन मिला। इसने वेब ब्राउज़िंग की शुरुआत को चिह्नित किया और वेबसाइटों और सामग्री साझा करने की नींव रखी।
- **ARPANET बंद** : ARPANET, जो अप्रचलित हो चुका था, को बंद कर दिया गया, तथा आधिकारिक तौर पर इसकी जगह आधुनिक इंटरनेट को दे दी गई।

1991

- **वर्ल्ड वाइड वेब को सार्वजनिक रूप से जारी किया गया**, जिससे लोगों को एक मानकीकृत प्रणाली के माध्यम से दस्तावेजों और वेबसाइटों को साझा करने और उन तक पहुंचने में मदद मिली।

# DSG Support Multi Solution

1993

- **मोज़ेक ब्राउज़र** : मार्क एंड्रीसेन और एरिक बीना ने मोज़ेक विकसित किया, जो पहला लोकप्रिय वेब ब्राउज़र था जो वेब पर एक ग्राफ़िकल इंटरफ़ेस लाया। इस नवाचार ने गैर-तकनीकी लोगों को आसानी से वेब का पता लगाने की अनुमति दी।
- मोज़ेक वेब ब्राउज़र की शुरुआत की गई, जिसने ग्राफ़िक्स और उपयोग में आसान इंटरफ़ेस को शामिल करके वेब को रोज़मर्रा के उपयोगकर्ताओं के लिए सुलभ बना दिया। इससे इंटरनेट के तेजी से विकास में मदद मिली।

1994

- **नेटस्केप का शुभारंभ** : नेटस्केप नेविगेटर पहला व्यापक रूप से प्रयुक्त वाणिज्यिक वेब ब्राउज़र बन गया और इसने इंटरनेट को रोज़मर्रा के उपयोगकर्ताओं के लिए अधिक सुलभ बनाने में बड़ी भूमिका निभाई।
- **अमेज़न और याहू की स्थापना हुई**, जो ई-कॉमर्स और ऑनलाइन खोज सेवाओं के उदय का प्रतीक है।

1995

- **इंटरनेट का व्यावसायीकरण** : नेटस्केप सार्वजनिक हो गया, जिससे माइक्रोसॉफ्ट के इंटरनेट एक्सप्लोरर के साथ "ब्राउज़र युद्ध" शुरू हो गया। **ईबे** भी लॉन्च हुआ, जिससे वेब की व्यावसायिक क्षमता और बढ़ गई।
- **वर्ल्ड वाइड वेब कंसोर्टियम (W3C)** : टिम बर्नर्स-ली ने वेब प्रोटोकॉल को मानकीकृत करने और एक स्वतंत्र और खुले मंच के रूप में इंटरनेट के विकास को सुनिश्चित करने के लिए W3C की स्थापना की।
- **जावा प्रोग्रामिंग भाषा** के आगमन से वेब पेजों पर गतिशील सामग्री का निर्माण संभव हो सका।

1998

- **गूगल की स्थापना** : लैरी पेज और सर्गेई ब्रिन ने गूगल की शुरुआत की, जिसने लोगों के जानकारी तक पहुँचने के तरीके को मौलिक रूप से बदल दिया। इसका सर्च एल्गोरिदम जल्द ही ऑनलाइन जानकारी खोजने का सबसे कारगर तरीका बन गया।

2000 का दशक: सोशल मीडिया और मोबाइल इंटरनेट का विस्फोट

2001

- **विकिपीडिया का शुभारंभ** : एक निःशुल्क, सहयोगात्मक ऑनलाइन विश्वकोश, विकिपीडिया इंटरनेट पर सबसे अधिक देखी जाने वाली वेबसाइटों में से एक बन गया, जिसने ज्ञान साझा करने के तरीके को बदल दिया।

2004

- **फेसबुक का जन्म** : मूल रूप से कॉलेज के छात्रों के लिए एक सामाजिक नेटवर्क, फेसबुक जिसे अब मेटा (मार्क जुकरबर्ग द्वारा स्थापित) के रूप में जाना जाता है, तेजी से विस्तारित होकर दुनिया के सबसे बड़े सामाजिक प्लेटफार्मों में से एक बन गया।

# DSG Support Multi Solution

2005

- **यूट्यूब की स्थापना** : वीडियो-शेयरिंग प्लेटफॉर्म यूट्यूब ने सामग्री निर्माण और उपभोग को बदल दिया, जिससे वीडियो ब्लॉगिंग (व्लॉगिंग) और वायरल सामग्री को बढ़ावा मिला।

2007

- **आईफोन का लॉन्च** : एप्पल के आईफोन ने मोबाइल इंटरनेट के युग की शुरुआत की। टचस्क्रीन और शक्तिशाली वेब ब्राउज़र के साथ, मोबाइल क्रांति ने दुनिया भर में इंटरनेट अपनाने में तेज़ी ला दी।

2008

- गूगल क्रोम ब्राउज़र का शुभारम्भ हुआ, जिसने तीव्र ब्राउज़िंग गति प्रदान की तथा वेब प्रदर्शन के लिए नए मानक स्थापित किए।

2010 का दशक: कनेक्टिविटी, ऐप्स और सामाजिक प्रभाव का युग

2010

- **इंस्टाग्राम लॉन्च** : फोटो और लघु वीडियो साझा करने पर केंद्रित इंस्टाग्राम ने सोशल मीडिया और प्रभावशाली संस्कृति के परिदृश्य को बदल दिया।

2011

- **स्नैपचैट** की स्थापना, क्षणिक विषय-वस्तु को प्रस्तुत करने तथा लोगों के ऑनलाइन संचार के तरीके को नया स्वरूप देने के लिए की गई थी।

2013

- **एडवर्ड स्नोडेन खुलासे** : पूर्व एनएसए ठेकेदार एडवर्ड स्नोडेन द्वारा लीक किए गए दस्तावेजों ने वैश्विक डिजिटल निगरानी की सीमा को उजागर कर दिया, जिससे ऑनलाइन गोपनीयता और सरकारी निगरानी के बारे में दुनिया भर में बहस छिड़ गई।

2015

- **इंटरनेट ऑफ थिंग्स (IoT)** : यह शब्द तब लोकप्रिय हुआ जब थर्मोस्टैट से लेकर कार तक के उपकरण इंटरनेट से जुड़ गए, जिससे रोजमर्रा की वस्तुएं "स्मार्ट" उपकरणों में बदल गईं।

2016

- **कृत्रिम बुद्धिमत्ता और इंटरनेट एकीकरण** : एआई प्रौद्योगिकियां, चैटबॉट और अमेज़ॉन के एलेक्सा और गूगल असिस्टेंट जैसे स्मार्ट सहायक रोजमर्रा के इंटरनेट उपयोग का हिस्सा बन गए हैं।

2018

- यूरोपीय संघ में **GDPR (जनरल डेटा प्रोटेक्शन रेगुलेशन)** लागू किया गया, जो गोपनीयता संबंधी चिंताओं को उजागर करता है और यह विनियमित करता है कि कंपनियां उपयोगकर्ता डेटा को कैसे संभालती हैं।

# DSG Support Multi Solution

2020 का दशक: दूरस्थ कार्य और AI एकीकरण का युग

2020

- **कोविड-19 महामारी** : महामारी के दौरान इंटरनेट और भी ज़रूरी हो गया, रिमोट वर्क, ऑनलाइन शिक्षा और वर्चुअल इवेंट आम बात हो गई। जूम जैसी सेवाओं में तेज़ी से वृद्धि देखी गई।

2021

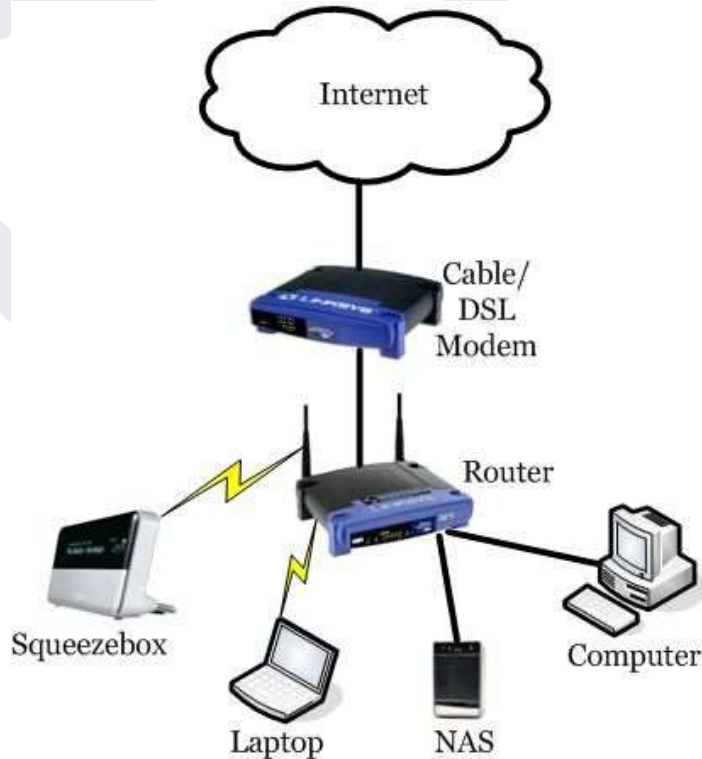
- **मेटावर्स अवधारणा ने गति पकड़ी, मेटा (पूर्व में फेसबुक)** जैसी प्रमुख तकनीकी कंपनियों ने सामाजिककरण, कार्य और वाणिज्य के लिए आभासी वास्तविकता स्थानों की खोज की।

2022

- **वेब3 और विकेंद्रीकरण** : वेब3, ब्लॉकचेन, विकेंद्रीकृत वित्त (डीएफआई) और क्रिप्टोकॉर्सेसी के बारे में चर्चा ने वैश्विक ध्यान आकर्षित किया, जिससे अधिक विकेंद्रीकृत इंटरनेट संरचना का वादा किया गया।

2024

- **एआई प्रभुत्व और इंटरनेट वैयक्तिकरण** : एआई ऑनलाइन उपयोगकर्ता अनुभव को आकार देना जारी रखता है। उन्नत सामग्री अनुशंसाओं से लेकर सहज डिजिटल सहायकों तक, वैयक्तिकरण इंटरनेट के भविष्य के लिए एक प्रमुख चालक बना हुआ है।
- **5G प्रौद्योगिकी का निरंतर विस्तार और AI नैतिकता** पर चर्चा आधुनिक इंटरनेट को आकार देती है, तथा यह प्रभावित करती है कि दुनिया भर में कनेक्टिविटी का प्रबंधन और विनियमन कैसे किया जाता है।





# DSG Support Multi Solution

## इंटरनेट की विशेषताएं

इंटरनेट तक पहुंचने की कई विशेषताएं हैं, जैसे:

- **किसी भी जानकारी तक आसानी से पहुँच:** कोई भी उपयोगकर्ता सिर्फ एक क्वेरी के ज़रिए इंटरनेट पर उपलब्ध सभी तरह की जानकारी तक पहुँच सकता है। दूसरे शब्दों में, इसे "विश्व का विश्वकोश" कहा जाता है।
- **आसान संचार और अनुसंधान:** एक संचार नेटवर्क के रूप में, इंटरनेट उपकरणों और नेटवर्क के बीच आसान संचार की स्थापना को सक्षम बनाता है। यहां तक कि किसी भी जानकारी को लिखने या विशेष कीवर्ड के लिए खोजने के लिए, शोध किए गए प्रश्न हमें पर्याप्त जानकारी प्राप्त करने में मदद करते हैं।
- **प्रभावी नेटवर्किंग के अवसर:** इंटरनेट की मदद से, अधिकांश कंपनियाँ और उपयोगकर्ता अन्य नेताओं या व्यवसायों के साथ प्रभावी नेटवर्किंग स्थापित करने के लिए इसका उपयोग कर रहे हैं। इससे अधिक राजस्व वृद्धि और पेशेवर पोर्टफोलियो विकास में मदद मिलती है।

## इंटरनेट के अनुप्रयोग

इंटरनेट के कुछ अनुप्रयोग इस प्रकार हैं:

### 1. गुणवत्तापूर्ण संचार और सूचना तक पहुंच

इंटरनेट ने लोगों के संवाद करने के तरीके में क्रांति ला दी है। ईमेल और इंस्टेंट मैसेजिंग जूम, गूगल मीट, स्काइप आदि के ज़रिए तेज़ी से संवाद करने की सुविधा देते हैं। फ़ेसबुक, ट्विटर और इंस्टाग्राम जैसे सोशल मीडिया प्लेटफ़ॉर्म उपयोगकर्ताओं को अपने परिवार या दोस्तों के साथ अपडेट, फ़ोटो और विचार साझा करने में सक्षम बनाते हैं। कई समाचार वेबसाइट और ब्लॉग सत्यापित स्रोतों के साथ उपयोगकर्ताओं को उचित जानकारी अपडेट करते हैं।

### 2. सामाजिक संपर्क

ऐसे कई ऑनलाइन फोरम और प्लेटफ़ॉर्म हैं जहां हम रेडिट, क्वोरा आदि पर विशेष मुद्दों पर चर्चा और अपने विचार व्यक्त कर सकते हैं। लिंकडइन और पिनटरेस्ट जैसी सोशल नेटवर्किंग साइटें लोगों को पेशेवर और व्यक्तिगत रूप से जुड़ने, रुचियां साझा करने और नए विचारों की खोज करने में मदद करती हैं।

### 3. उत्पादकता और कार्य

इंटरनेट ने स्वचालन और एआई के एकीकरण के माध्यम से अपनी कार्य उत्पादकता में सुधार करने के लिए सभी को बहुत प्रभावित किया है। असाना और जीरा जैसे प्रोजेक्ट प्रबंधन उपकरण टीमों को कार्यों को व्यवस्थित करने और प्रगति को ट्रैक करने में मदद करते हैं।



# DSG Support Multi Solution

## Advantage Of Internet

इंटरनेट शायद अब तक के सबसे बेहतरीन आविष्कारों में से एक है। चूंकि कोई भी व्यक्ति कंप्यूटर का उपयोग कर सकता है, इसलिए इसने भौगोलिक बाधाओं को दूर करके और तुरंत जानकारी साझा करके लोगों को सक्षम बनाया है।

- **संचार मंच:** संचार की गति तेज़ हो जाती है जो वेब के माध्यम से प्राप्त होती है। परिवार और दोस्त आसानी से इंटरनेट का उपयोग कर सकते हैं। स्काइप, ज़ूम, टीम्स आदि जैसे उत्पादों के लिए मंच दुनिया में किसी भी व्यक्ति के साथ वीडियो कॉन्फ्रेंस आयोजित करने की अनुमति देता है, जिसके पास भी पहुंच है। यह कहीं भी मौजूद दो लोगों को बहुत आसानी से जोड़ने में मदद करता है।
- **प्रचुर जानकारी:** कोई भी व्यक्ति लगभग किसी भी कल्पनीय विषय पर जानकारी प्राप्त कर सकता है। कार्यक्रम के माध्यम से अक्सर मिनटों में बहुत सारे संसाधन मिल जाते हैं। इंटरनेट सभी विषयों पर जानकारी प्रदान करने में मदद करता है।
- **असीमित शिक्षा:** उदाहरण के लिए, छात्र अपने होमवर्क के लिए ऑनलाइन आसानी से उपलब्ध सहायता प्राप्त कर सकते हैं। न केवल होमवर्क बल्कि ऐसे कई प्लेटफॉर्म भी हैं जो ऑनलाइन शिक्षा प्रदान करते हैं जो छात्रों को विभिन्न क्षेत्रों में ज्ञान प्राप्त करने में मदद करते हैं। यह शिक्षार्थी के ज्ञान को बढ़ाने में मदद करता है।
- **सबके लिए मनोरंजन:** ज़्यादातर लोग लैपटॉप, स्मार्टफोन और टैबलेट का इस्तेमाल करना पसंद करते हैं। इन गैजेट पर इतना ज़्यादा समय बिताने के पीछे सबसे बड़ा कारण इंटरनेट है। इंटरनेट की आसानी के साथ, मनोरंजन उद्योग भी बढ़ गया है और यह कई तरह से विस्तार करने में मदद करता है जैसे कि ओटीटी प्लेटफॉर्म और कई अन्य प्लेटफॉर्म आदि।
- **ऑनलाइन सेवाएँ और ई-कॉमर्स:** ईमेल, ऑनलाइन बैंकिंग, ऑनलाइन शॉपिंग आदि की कई सेवाएँ उपलब्ध हैं। दुनिया भर में किसी को भी मुफ्त मेल भेजना निश्चित रूप से सुलभ है। ई-कॉमर्स अमेरिका में किसी को भी माउस के कुछ सरल क्लिक के माध्यम से एशिया, अफ्रीका या दुनिया के अन्य क्षेत्रों में चीजों की खरीदारी करने में सक्षम बनाता है।
- **सोशल नेटवर्क:** [सोशल नेटवर्किंग](#) दुनिया भर के लोगों के साथ डेटा साझा करना है। मनोरंजन वेबसाइट होने के अलावा, इसके कई उपयोग हैं। इंटरनेट ने इन सोशल नेटवर्किंग साइट्स को बहुत बड़ा लाभ दिया है क्योंकि यह कुछ ही सेकंड में दूर बैठे लोगों को आपस में जोड़ता है।
- **सीखना:** इंटरनेट अब शिक्षा का एक क्षेत्र बन गया है। होमस्कूलिंग जैसी शिक्षा निश्चित रूप से इंटरनेट का उपयोग करके संचालित की जाती है। शिक्षक अपने शिक्षण वीडियो को वेब पर अपलोड कर सकते हैं और आजकल इंटरनेट की पहुंच के कारण छात्रों को पढ़ाने के लिए कई और प्लेटफॉर्म आ गए हैं।

# DSG Support Multi Solution

- **इंटरनेट बैंकिंग:** इंटरनेट बैंकिंग इंटरनेट के मुख्य लाभों में से एक है क्योंकि यह आजकल लोगों को डिजिटल तरीके से बैंकों का उपयोग करने में मदद करता है जिससे लोगों को बैंकों में जाने की जरूरत नहीं पड़ती, बैंकों में कतारों में खड़े होने की जरूरत नहीं पड़ती, आदि। लोग [इंटरनेट](#) की मदद से अपनी उंगलियों पर बैंकिंग सेवाओं का उपयोग कर सकते हैं ।

## Disadvantage Of Internet

इंटरनेट की कमियों को अब नजरअंदाज नहीं किया जा सकता है क्योंकि कई किशोर इंटरनेट एडिक्शन डिसऑर्डर से प्रभावित हैं, तो कई महिलाएं ऑनलाइन शॉपिंग की आदी हो गई हैं।

- **Internet Addiction Disorder:** [इंटरनेट की लत](#) न केवल स्वास्थ्य के लिए बल्कि मनोवैज्ञानिक स्थिति के लिए भी हानिकारक है। बहुत से लोग इस समस्या से पीड़ित हैं क्योंकि उन्हें इंटरनेट की लत है और वे अपने डिवाइस का इस्तेमाल बिना सोचे-समझे करते हैं कि उन्होंने डिवाइस पर कितना समय बर्बाद किया है।
- **साइबर अपराध:** [साइबर अपराध](#) हैकर प्रोग्राम द्वारा वायरस के माध्यम से होता है जो कंप्यूटर में प्रवेश करता है और मूल्यवान डेटा को नष्ट कर देता है। उपयोगकर्ताओं की व्यक्तिगत जानकारी जैसे नाम, पता, मास्टर कार्ड, बैंक विवरण और अन्य जानकारी अक्सर अपराधियों द्वारा वेब पर उपयोग किए जाने पर एक्सेस की जाती है, जिससे महत्वपूर्ण आर्थिक नुकसान होता है।
- **Social Alienation:** ऑनलाइन बिताया गया समय बिना किसी चेतना के तेज़ी से बीत जाता है। आकर्षित होने के बाद उपयोगकर्ता जाल में फंस जाता है, उपयोगकर्ता एक "जाल" में फंस जाता है, वास्तविक दुनिया में लोगों के साथ कम समय बिताता है। कम बातचीत और आमने-सामने संचार, वास्तव में, सामाजिक क्षमताओं में कमी का कारण बन सकता है।
- **स्पैम:** अनावश्यक ईमेल, विज्ञापन आदि को कभी-कभी स्पैम कहा जाता है क्योंकि उन्हें सिस्टम को बाधित करने और उपयोगकर्ताओं को कई समस्याओं का सामना करने के लिए मजबूर करने की शक्ति की आवश्यकता होती है। इंटरनेट के अत्यधिक उपयोग से बैंक खाते की धोखाधड़ी, इंटरनेट धोखाधड़ी आदि जैसी कई समस्याएं पैदा होती हैं।
- **स्वास्थ्य संबंधी समस्याएं:** गेम खेलने और मॉनिटर पर बहुत ज़्यादा समय बिताने से मोटापा बढ़ता है और जीवनशैली भी खराब होती है। यह डिवाइस का लंबे समय तक इस्तेमाल करने वाले व्यक्ति की आंखों की रोशनी पर भी असर डालता है। इससे स्वास्थ्य संबंधी बहुत गंभीर समस्याएं होती हैं।
- **समय की बर्बादी:** इंटरनेट एक ऐसा तरीका है जो आपके समय की बर्बादी का कारण बन सकता है। आजकल हम सभी के साथ ऐसा होता है, जब हम लगातार सर्फिंग करते हैं और हमें पता ही नहीं चलता कि कितना समय बर्बाद हो गया। इससे स्क्रीन टाइम बढ़ जाता है और इससे मेमोरी पावर कम होती है।

# DSG Support Multi Solution

- **वायरस/मैलवेयर:** इंटरनेट के उपयोग से सिस्टम में वायरस आ सकते हैं और सिस्टम अनुत्पादक हो सकता है, यह इंटरनेट का उपयोग करने का एक बड़ा नुकसान है क्योंकि वायरस के कारण आपका सिस्टम हैंग हो सकता है।
- **ध्यान और धैर्य पर प्रभाव:** इंटरनेट व्यक्ति के ध्यान और धैर्य को प्रभावित कर सकता है, जिसके कारण व्यक्ति जीवन में अपने लक्ष्यों को प्राप्त करने से विचलित हो सकता है और यह इंटरनेट के नुकसानों में से एक है।

## Internetworking Concepts

इंटरनेटवर्किंग दो शब्दों, इंटर और नेटवर्किंग का संयोजन है, जिसका अर्थ है पूरी तरह से अलग-अलग नोड्स या सेगमेंट के बीच संबंध। यह कनेक्शन क्षेत्र इकाई राउटर या गेटवे जैसे इंटरसेसर डिवाइस के माध्यम से स्थापित की जाती है। एसोसिएट डिग्री इंटरनेटवर्क के लिए पहला शब्द कैटनेट था। यह इंटरकनेक्शन अक्सर सार्वजनिक, निजी, वाणिज्यिक, औद्योगिक या सरकारी नेटवर्क के बीच या उनके बीच होता है। इस प्रकार, एसोसिएट डिग्री इंटरनेटवर्क व्यक्तिगत नेटवर्क का एक संग्रह हो सकता है, जो मध्यवर्ती नेटवर्किंग उपकरणों द्वारा जुड़ा होता है, जो एक विशाल नेटवर्क के रूप में कार्य करता है। इंटरनेटवर्किंग व्यापार, उत्पादों और प्रक्रियाओं को संदर्भित करता है जो इंटरनेट कार्यों को बनाने और प्रशासित करने की चुनौती को पूरा करते हैं।

संचार को सक्षम करने के लिए, प्रत्येक व्यक्तिगत नेटवर्क नोड या चरण को एक समान प्रोटोकॉल या संचार तर्क के साथ डिज़ाइन किया गया है, जिसे ट्रांसफर कंट्रोल प्रोटोकॉल (TCP) या इंटरनेट प्रोटोकॉल (IP) कहा जाता है। जब कोई नेटवर्क निरंतर संचार प्रक्रियाओं वाले किसी अन्य नेटवर्क के साथ संचार करता है, तो इसे इंटरनेटवर्किंग कहा जाता है। इंटरनेटवर्किंग को कई लिंक के माध्यम से सूचना के पैकेट को वितरित करने के मामले को हल करने के लिए डिज़ाइन किया गया था।

नेटवर्क का विस्तार करने और इंटरनेटवर्किंग के बीच एक छोटा सा अंतर है। 2 लोकल एरिया नेटवर्क को जोड़ने के लिए स्विच या हब का उपयोग करना LAN का विस्तार है जबकि राउटर के माध्यम से उन्हें जोड़ना इंटरनेटवर्किंग का एक सहयोगी डिग्री उदाहरण है। इंटरनेटवर्किंग को OSI-ISO मॉडल की लेयर तीन (नेटवर्क लेयर) में लागू किया जाता है। इंटरनेटवर्किंग का सबसे उल्लेखनीय उदाहरण इंटरनेट है।

इंटरनेटवर्किंग की मुख्यतः 3 इकाइयाँ हैं:

1. एक्स्ट्रानेट
2. इंट्रानेट
3. इंटरनेट

# DSG Support Multi Solution

इंट्रानेट और एक्स्ट्रानेट का नेट से कनेक्शन हो भी सकता है और नहीं भी। यदि नेट से कनेक्शन है, तो कंप्यूटर नेटवर्क या एक्स्ट्रानेट क्षेत्र इकाई को आमतौर पर नेट से एक्सेस किए जाने से बचाया जाता है, यदि यह अधिकृत नहीं है। नेट को कंप्यूटर नेटवर्क या एक्स्ट्रानेट का हिस्सा नहीं माना जाता है, हालांकि इसे एसोसिएट डिग्री एक्स्ट्रानेट के कुछ हिस्सों तक पहुंच के लिए एक पोर्टल के रूप में कार्य करना चाहिए।

- 1. एक्स्ट्रानेट** - यह इंटरनेटवर्क का एक नेटवर्क है जो एक संगठन या इकाई के दायरे में सीमित है, लेकिन इसके अतिरिक्त इसमें कभी-कभी एक या कई अलग-अलग नेटवर्क के लिए सीमित कनेक्शन होते हैं, हालांकि यह आवश्यक नहीं है। यह इंटरनेटवर्किंग का सबसे निचला स्तर है, जिसे आमतौर पर एक बहुत ही निजी क्षेत्र में लागू किया जाता है। एसोसिएट डिग्री एक्स्ट्रानेट को मैन, WAN या नेटवर्क के अन्य रूप के रूप में भी वर्गीकृत किया जा सकता है, लेकिन यह एक स्थानीय क्षेत्र नेटवर्क को शामिल नहीं कर सकता है यानी इसमें एसोसिएट डिग्री बाहरी नेटवर्क के लिए कम से कम एक संदर्भ होना चाहिए।
- 2. इंट्रानेट** - यह एसोसिएट डिग्री कंप्यूटर नेटवर्क आपस में जुड़े नेटवर्क का एक सेट हो सकता है, जो इंटरनेट प्रोटोकॉल का फायदा उठाता है और वेब ब्राउज़र और FTP टूल जैसे IP-आधारित टूल का उपयोग करता है, जो एक बॉडी एंटीटी के प्रबंधन के अधीन होते हैं। वह बॉडी एंटीटी कंप्यूटर नेटवर्क को बाकी दुनिया से बंद कर देती है और केवल विशिष्ट उपयोगकर्ताओं को अनुमति देती है। आमतौर पर, यह नेटवर्क किसी निगम या अन्य उद्यम का आंतरिक नेटवर्क होता है। एक बड़े कंप्यूटर नेटवर्क में आमतौर पर उपयोगकर्ताओं को ब्राउज़ करने योग्य डेटा प्रदान करने के लिए अपना स्वयं का इंटरनेट सर्वर हो सकता है।
- 3. इंटरनेट** - एक चयनित इंटरनेटवर्किंग, जिसमें सरकारी, शैक्षणिक, सार्वजनिक और व्यक्तिगत नेटवर्क का एक विश्वव्यापी अंतर्संबंध शामिल है, जो मुख्य रूप से अमेरिकी रक्षा विभाग के ARPA द्वारा विकसित एडवांस्ड एनालिसिस कम्स एजेंसी नेटवर्क (ARPANET) पर आधारित है, जो वर्ल्ड वाइड वेब (WWW) का भी घर है और इसे अन्य सभी सामान्य इंटरनेटवर्क से अलग करने के लिए 'इंटरनेट' के रूप में उद्धृत किया गया है। वेब के भीतर प्रतिभागी, या उनके सेवा प्रदाता, असाइनमेंट प्रबंधित करने वाली पता रजिस्ट्री से प्राप्त IP पते का उपयोग करते हैं।

इंटरनेटवर्किंग कुछ प्रमुख समस्याओं के समाधान के रूप में विकसित हुई है: पृथक LAN, संसाधनों का दोहराव और नेटवर्क प्रबंधन की अनुपस्थिति। पृथक LAN ने पूरी तरह से अलग-अलग कार्यालयों या विभागों के बीच संचार की समस्याएँ पैदा कीं। संसाधनों के दोहराव का मतलब था कि हर कार्यस्थल या विभाग को लगातार हार्डवेयर और कोड प्रदान करना पड़ता था, साथ ही एक अलग सहायक कर्मचारी भी।

# DSG Support Multi Solution

नेटवर्क प्रबंधन की इस कमी का मतलब था कि नेटवर्क के प्रबंधन और समस्या निवारण की कोई केंद्रीकृत पद्धति मौजूद नहीं थी।

नेटवर्क के अंतर्संबंध का एक और रूप आमतौर पर नेटवर्किंग मॉडल के लिंक लेयर पर उद्यमों के बीच होता है, यानी TCP/IP लॉजिकल इंटरफेस की मात्रा के नीचे हार्डवेयर-केंद्रित परत पर। इस तरह के अंतर्संबंध को नेटवर्क ब्रिज और नेटवर्क स्विच के माध्यम से पूरा किया जाता है। इसे आमतौर पर गलत तरीके से इंटरनेटवर्किंग कहा जा सकता है, हालाँकि, परिणामी प्रणाली केवल एक बड़ा, एकल सबनेटवर्क है, और इन उपकरणों को पार करने के लिए वेब प्रोटोकॉल के समान किसी इंटरनेटवर्किंग प्रोटोकॉल की आवश्यकता नहीं होती है।

हालाँकि, एक इलेक्ट्रॉनिक नेटवर्क को नेटवर्क को चरणों में विभाजित करके और राउटर के साथ सेगमेंट ट्रैफिक को तार्किक रूप से विभाजित करके एसोसिएट डिग्री इंटरनेटवर्क में पुनर्जन्म भी दिया जाता है। इंटरनेट प्रोटोकॉल का उद्देश्य पूरे नेटवर्क में एक एसोसिएट डिग्री अविश्वसनीय पैकेट सेवा प्रदान करना है। डिज़ाइन नेटवर्क की किसी भी स्थिति को बनाए रखने वाले मध्यवर्ती नेटवर्क घटकों से बचता है। इसके बजाय, यह कार्य प्रत्येक संचार सत्र के अंतिम बिंदुओं को आवंटित किया जाता है। सूचना को सही ढंग से स्थानांतरित करने के लिए, अनुप्रयोगों को एसोसिएट डिग्री लागू ट्रांसपोर्ट लेयर प्रोटोकॉल का उपयोग करना चाहिए, ट्रांसमिशन मैनेजमेंट प्रोटोकॉल (TCP) के समान, जो एक विश्वसनीय स्ट्रीम प्रदान करता है। कुछ अनुप्रयोग कम जटिल, कनेक्शन-रहित ट्रांसपोर्ट प्रोटोकॉल, यूजर डेटाग्राम प्रोटोकॉल (UDP) का उपयोग करते हैं, उन कार्यों के लिए जिन्हें सूचना की विश्वसनीय डिलीवरी की आवश्यकता नहीं होती है या जिन्हें समय की सेवा की आवश्यकता होती है, जैसे वीडियो स्ट्रीमिंग या वॉयस चैट।

## इंटरनेटवर्क एड्रेसिंग -

इंटरनेटवर्क पते डिवाइस को अलग-अलग या समूह के सदस्यों के रूप में स्थापित करते हैं। एड्रेसिंग योजनाएँ प्रोटोकॉल परिवार और इसलिए OSI परत के आधार पर भिन्न होती हैं। तीन प्रकार के इंटरनेटवर्क पते क्षेत्र इकाइयों का आमतौर पर उपयोग किया जाता है: डेटा-लिंक परत पते, मीडिया एक्सेस कंट्रोल (MAC) पते, और नेटवर्क-लेयर पते।

- 1. डेटा लिंक लेयर एड्रेस:** डेटा-लिंक लेयर एड्रेस किसी नेटवर्क डिवाइस के हर भौतिक नेटवर्क एसोसिएशन को स्पष्ट रूप से पहचानता है। डेटा-लिंक एड्रेस आमतौर पर भौतिक या हार्डवेयर एड्रेस के रूप में उद्धृत इकाइयाँ होती हैं। डेटा-लिंक एड्रेस कभी-कभी एक फ्लैट एड्रेस एरिया के बीच मौजूद होते हैं और किसी चयनित डिवाइस के साथ पहले से स्थापित और आमतौर पर तय संबंध रखते हैं। एंड सिस्टम में

# DSG Support Multi Solution

आमतौर पर सिर्फ एक भौतिक नेटवर्क एसोसिएशन होता है, और इसलिए सिर्फ एक डेटा-लिंक एड्रेस होता है। राउटर और विभिन्न इंटरनेटवर्किंग डिवाइस में आमतौर पर कई भौतिक नेटवर्क कनेक्शन होते हैं और इसलिए अंततः कई डेटा-लिंक एड्रेस होते हैं।

- 2. MAC पते:** मीडिया एक्सेस मैनेजमेंट (MAC) पते डेटा-लिंक लेयर पतों के एक सेट को शामिल करते हैं। MAC पते LAN में नेटवर्क इकाइयाँ स्थापित करते हैं जो डेटा-लिंक लेयर के IEEE MAC पतों को लागू करते हैं। MAC पते प्रत्येक स्थानीय क्षेत्र नेटवर्क इंटरफ़ेस के लिए अलग-अलग क्षेत्र इकाइयों को अलग-अलग तरीके से संबोधित करते हैं। MAC पते अड़तालीस बिट लंबे होते हैं और बारह हेक्साडेसिमल अंकों के रूप में व्यक्त किए जाते हैं। प्राथमिक आधा दर्जन हेक्साडेसिमल अंक, जो आमतौर पर IEEE द्वारा प्रशासित होते हैं, निर्माता या व्यापारी को स्थापित करते हैं और इसलिए संगठनात्मक विशिष्ट पहचानकर्ता (OUI) को शामिल करते हैं। अंतिम आधा दर्जन स्थितीय संकेतन अंक इंटरफ़ेस सीरियल किस्म या विशेष व्यापारी द्वारा प्रशासित अन्य मूल्य को शामिल करते हैं। MAC पते आमतौर पर क्षेत्र इकाइयाँ होती हैं जिन्हें बर्न-इन एड्रेस (BIA) के रूप में संदर्भित किया जाता है क्योंकि उन्हें रीड-ओनली मेमोरी (ROM) में बर्न किया जाता है और इंटरफ़ेस कार्ड के आरंभ होने के बाद रैंडम-एक्सेस मेमोरी (RAM) में ट्रेस किया जाता है।
- 3. नेटवर्क-लेयर एड्रेस:** नेटवर्क एड्रेस कभी-कभी एक ग्रेडेबल एड्रेस एरिया और आम तौर पर वर्चुअल या लॉजिकल एड्रेस के रूप में संदर्भित क्षेत्र इकाइयों के बीच मौजूद होते हैं। नेटवर्क एड्रेस और टूल के बीच कनेक्शन तार्किक और अनिश्चित है, यह आमतौर पर या तो भौतिक नेटवर्क विशेषताओं पर या ऐसे समूहों पर निर्भर करता है जिनका कोई भौतिक आधार नहीं होता है। अंतिम सिस्टम को उनके द्वारा समर्थित प्रत्येक नेटवर्क-लेयर प्रोटोकॉल के लिए एक नेटवर्क-लेयर एड्रेस की आवश्यकता होती है। राउटर और अन्य इंटरनेटवर्किंग डिवाइस को समर्थित प्रत्येक नेटवर्क-लेयर प्रोटोकॉल के लिए प्रति भौतिक नेटवर्क एसोसिएशन में एक नेटवर्क-लेयर एड्रेस की आवश्यकता होती है।

## इंटरनेटवर्किंग की चुनौतियाँ -

उपयोगी इंटरनेटवर्क को लागू करना किसी भी तरह से निश्चित नहीं है। कई चुनौतीपूर्ण क्षेत्र हैं, विशेष रूप से निर्भरता, कनेक्टिविटी, नेटवर्क प्रबंधन और अनुकूलनशीलता के क्षेत्रों में, और प्रत्येक क्षेत्र किफायती और प्रभावी इंटरनेटवर्क स्थापित करने में आवश्यक है। उनमें से कुछ हैं:-

- प्रारंभिक चुनौती तब आती है जब हम अलग-अलग तकनीकों के बीच संचार का समर्थन करने के लिए कई प्रणालियों को जोड़ने की कोशिश कर रहे होते हैं। उदाहरण के लिए, पूरी तरह से अलग-अलग साइटें अलग-अलग तरह के मीडिया का उपयोग कर सकती हैं, या वे अलग-अलग गति से काम कर सकती हैं।



# DSG Support Multi Solution

- एक और ज़रूरी विचार है विश्वसनीय सेवा जिसे इंटरनेटवर्क में बनाए रखा जाना चाहिए। व्यक्तिगत उपयोगकर्ता और पूरा संगठन नेटवर्क संसाधनों तक निरंतर, विश्वसनीय पहुंच पर निर्भर करता है।
- नेटवर्क प्रबंधन को इंटरनेटवर्क पर समस्या निवारण क्षमताओं के लिए केंद्रीकृत समर्थन प्रदान करना चाहिए। इंटरनेटवर्क के सुचारू रूप से कार्य करने के लिए कॉन्फ़िगरेशन, सुरक्षा, प्रदर्शन और विभिन्न समस्याओं को पर्याप्त रूप से संबोधित किया जाना चाहिए।
- लचीलापन, अंतिम चिंता का विषय है, जो विभिन्न कारकों के अलावा, नेटवर्क विस्तार और नए अनुप्रयोगों और सेवाओं के लिए महत्वपूर्ण है।

लाभ:

**बढ़ी हुई कनेक्टिविटी:** इंटरनेटवर्किंग विभिन्न नेटवर्क पर स्थित उपकरणों को एक-दूसरे के साथ संचार करने में सक्षम बनाता है, जिससे कनेक्टिविटी बढ़ती है और नए अनुप्रयोगों और सेवाओं को सक्षम किया जाता है।

**संसाधन साझा करना:** इंटरनेटवर्किंग डिवाइस को प्रिंटर, सर्वर और स्टोरेज डिवाइस जैसे नेटवर्क पर संसाधनों को साझा करने की अनुमति देता है। यह कई डिवाइस को संसाधन साझा करने की अनुमति देकर लागत को कम कर सकता है और दक्षता में सुधार कर सकता है।

**उन्नत मापनीयता:** इंटरनेटवर्किंग, नेटवर्क को आवश्यकतानुसार विस्तारित और स्केल करने की अनुमति देता है, ताकि बढ़ती हुई डिवाइसों और उपयोगकर्ताओं की संख्या को समायोजित किया जा सके।

**बेहतर सहयोग:** इंटरनेटवर्किंग टीमों और व्यक्तियों को उनके भौतिक स्थान की परवाह किए बिना अधिक प्रभावी ढंग से सहयोग करने और एक साथ काम करने में सक्षम बनाता है।

**दूरस्थ संसाधनों तक पहुंच:** इंटरनेटवर्किंग उपयोगकर्ताओं को दूरस्थ नेटवर्क पर भौतिक रूप से स्थित संसाधनों और सेवाओं तक पहुंचने की अनुमति देता है, जिससे पहुंच और लचीलेपन में सुधार होता है।

नुकसान:

**सुरक्षा जोखिम:** इंटरनेटवर्किंग सुरक्षा कमज़ोरियाँ पैदा कर सकती है और साइबर हमलों और डेटा उल्लंघनों के जोखिम को बढ़ा सकती है। कई नेटवर्क को एक साथ जोड़ने से हमलावरों के लिए प्रवेश बिंदुओं की संख्या बढ़ जाती है, जिससे पूरे सिस्टम को सुरक्षित करना अधिक कठिन हो जाता है।

**जटिलता:** इंटरनेटवर्किंग जटिल हो सकती है और इसे स्थापित करने और बनाए रखने के लिए विशेष ज्ञान और विशेषज्ञता की आवश्यकता होती है। इससे लागत बढ़ सकती है और अतिरिक्त रखरखाव ओवरहेड बन सकता है।

**प्रदर्शन संबंधी समस्याएं:** इंटरनेटवर्किंग से प्रदर्शन संबंधी समस्याएं हो सकती हैं, खासकर अगर नेटवर्क को ठीक से अनुकूलित और कॉन्फ़िगर नहीं किया गया हो। इसके परिणामस्वरूप धीमी प्रतिक्रिया समय और खराब नेटवर्क प्रदर्शन हो सकता है।



# DSG Support Multi Solution

**संगतता संबंधी समस्याएं:** इंटरनेटवर्किंग से संगतता संबंधी समस्याएं हो सकती हैं, खासकर तब जब अलग-अलग नेटवर्क अलग-अलग प्रोटोकॉल या तकनीक का इस्तेमाल कर रहे हों। इससे अलग-अलग सिस्टम को एकीकृत करना मुश्किल हो सकता है और इसे हल करने के लिए अतिरिक्त संसाधनों की आवश्यकता हो सकती है।

**प्रबंधन ओवरहेड:** इंटरनेटवर्किंग अतिरिक्त प्रबंधन ओवरहेड बना सकता है, खासकर अगर कई नेटवर्क शामिल हों। इससे लागत बढ़ सकती है और प्रभावी ढंग से प्रबंधन करने के लिए अतिरिक्त संसाधनों की आवश्यकता हो सकती है।

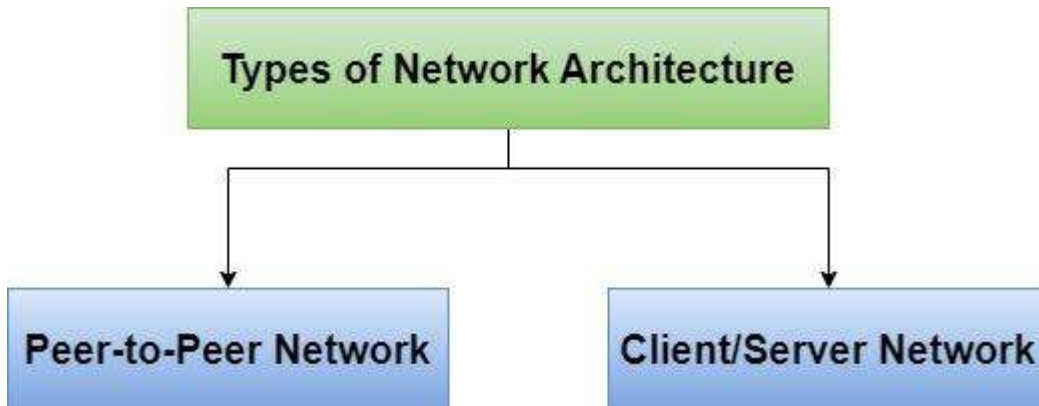
## Internet Architecture

कंप्यूटर नेटवर्क आर्किटेक्ट किसी कंपनी के लिए कंप्यूटर और संचार प्रणाली स्थापित करता है। "नेटवर्क आर्किटेक्चर" का मतलब है नियम, उपकरण और सॉफ्टवेयर जो कंपनी में लोगों को बिना किसी समस्या के जुड़े रहने में मदद करते हैं।

नेटवर्क आर्किटेक्चर के प्रकार

कंप्यूटर नेटवर्क को आर्किटेक्चर के आधार पर दो प्राथमिक प्रकारों में वर्गीकृत किया जा सकता है:

- पीयर-टू-पीयर आर्किटेक्चर
- क्लाइंट/सर्वर आर्किटेक्चर



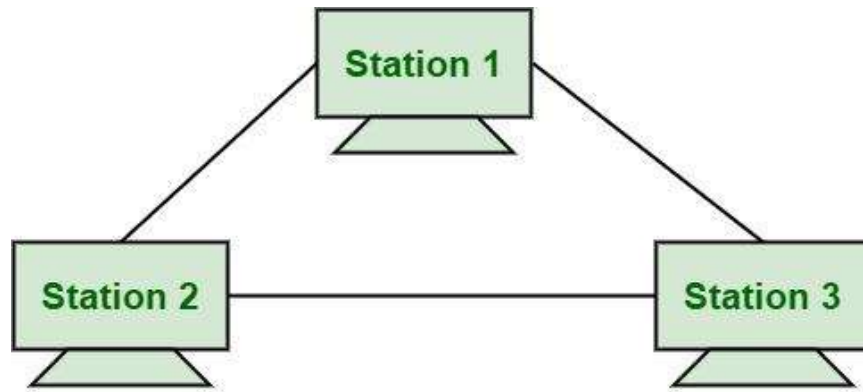
नेटवर्क आर्किटेक्चर के प्रकार

पीयर-टू-पीयर आर्किटेक्चर क्या है?

पी2पी (पीयर-टू-पीयर) नेटवर्क में, "पीयर" आम तौर पर कंप्यूटर सिस्टम का प्रतिनिधित्व करते हैं। ये पीयर इंटरनेट की मदद से एक दूसरे से जुड़े होते हैं। नेटवर्क पर इन सिस्टम के बीच केंद्रीय सर्वर की आवश्यकता के बिना फ़ाइलों को सीधे साझा किया जा सकता है। यह कहा जा सकता है कि पी2पी नेटवर्क पर प्रत्येक कंप्यूटर आमतौर पर क्लाइंट के साथ-साथ फ़ाइल सर्वर भी बन जाता है।

# DSG Support Multi Solution

इस आर्किटेक्चर में, सिस्टम को आम तौर पर विभिन्न कम्प्यूटेशनल नोड्स में विघटित किया जाता है, जिनमें समान और समतुल्य क्षमताएं, योग्यताएं और जिम्मेदारियां होती हैं। इस नेटवर्क में, नेटवर्क पर उपलब्ध प्रत्येक डिवाइस पर कार्य आवंटित किए जाते हैं। यह नेटवर्क छोटे वातावरण के लिए बहुत आवश्यक और महत्वपूर्ण है, आमतौर पर कम से कम 10 कंप्यूटर तक। क्लाइंट और सर्वर के रूप में कोई अलग विभाजन भी नहीं है। इस नेटवर्क में प्रत्येक कंप्यूटर के साथ समान और समान व्यवहार किया जाता है और वे सीधे संदेश भी भेज सकते हैं या प्राप्त कर सकते हैं। यह पी2पी नेटवर्क आम तौर पर व्यवसाय, शिक्षा, सैन्य आदि जैसे विभिन्न क्षेत्रों में उपयोगी होता है।



## Peer-to-Peer Architecture

पीयर-टू-पीयर आर्किटेक्चर

पीयर-टू-पीयर आर्किटेक्चर के लाभ

- पी2पी नेटवर्क कम खर्चीला और सस्ता है। यह किफायती है।
- पी2पी बहुत सरल है और जटिल नहीं है। ऐसा इसलिए है क्योंकि नेटवर्क में जुड़े सभी कंप्यूटर एक दूसरे के साथ कुशल और सुव्यवस्थित तरीके से संचार करते हैं।
- इसे स्थापित करना और प्रबंधित करना बहुत आसान और सरल है क्योंकि इंस्टॉलेशन और सेटअप कम दर्द रहित है और कंप्यूटर खुद को प्रबंधित करता है। यह आधुनिक ऑपरेटिंग सिस्टम में अंतर्निहित समर्थन के कारण है।
- इस प्रकार के नेटवर्क में सुरक्षा एक प्रमुख मुद्दा है।
- यदि कुछ संसाधनों के साथ काम करने वाला कंप्यूटर बंद हो जाए तो संसाधनों को साझा करना बड़ी समस्या बन सकती है।
- इस नेटवर्क पर कंप्यूटरों की संख्या बढ़ने से प्रदर्शन, सुरक्षा और पहुंच भी बड़ी समस्या बन सकती है।

पीयर-टू-पीयर आर्किटेक्चर के नुकसान

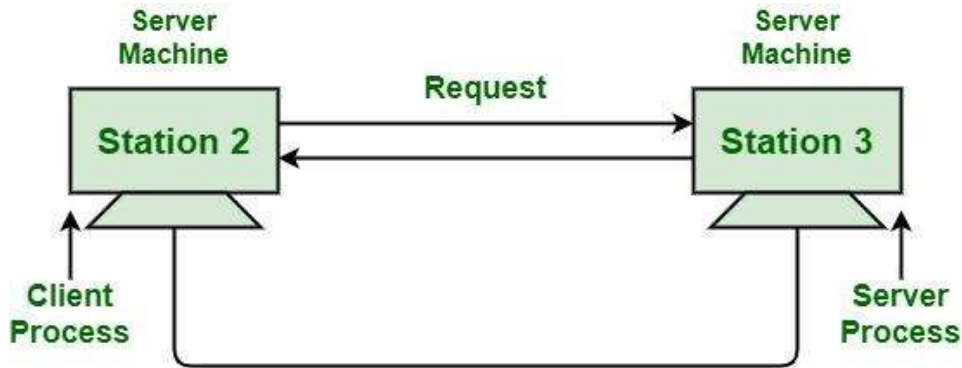
- सुरक्षा नीतियों को लगातार प्रबंधित करना अधिक कठिन है।

# DSG Support Multi Solution

- प्रत्येक सहकर्मी व्यक्तिगत देखभाल और नियंत्रण की मांग करता है।
- जैसे-जैसे नेटवर्क का आकार बढ़ता जाएगा, यह अकुशल होता जाएगा।

क्लाइंट/सर्वर आर्किटेक्चर क्या है?

CSN (क्लाइंट/सर्वर नेटवर्क) एक प्रकार का कंप्यूटर नेटवर्क है जिसमें केंद्रीकृत और शक्तिशाली कंप्यूटरों में से एक (जिसे आमतौर पर सर्वर कहा जाता है) हब होता है जिससे कई कम शक्तिशाली पर्सनल कंप्यूटर या वर्कस्टेशन (जिन्हें आमतौर पर क्लाइंट के रूप में जाना जाता है) जुड़े होते हैं। यह एक प्रकार का सिस्टम है जहाँ क्लाइंट केवल संसाधनों को साझा करने या उपयोग करने के लिए सर्वर से जुड़े होते हैं। इन सर्वरों को आम तौर पर सिस्टम का दिल माना जाता है। इस प्रकार का नेटवर्क P2P नेटवर्क की तुलना में अधिक स्थिर और स्केलेबल होता है। इस आर्किटेक्चर में, सिस्टम को आम तौर पर क्लाइंट और सर्वर प्रोसेसर या प्रक्रियाओं में विघटित किया जाता है।



## Client/Server Architecture

क्लाइंट/सर्वर आर्किटेक्चर

क्लाइंट/सर्वर आर्किटेक्चर के लाभ

- सर्वर द्वारा एक विशेष नेटवर्क ऑपरेटिंग सिस्टम (एनओएस) प्रदान किया जाता है, जो कई उपयोगकर्ताओं को संसाधन उपलब्ध कराने के लिए अनुरोध करता है।
- डेटा अपडेट को सेट अप करना और प्रबंधित करना भी बहुत आसान और सरल है। ऐसा इसलिए है क्योंकि डेटा आमतौर पर सर्वर पर केंद्रीकृत तरीके से संग्रहीत किया जाता है।
- सर्वर आमतौर पर संसाधनों और डेटा सुरक्षा को नियंत्रित करता है।
- यह नेटवर्क संसाधनों को साझा करने की गति को भी बढ़ाता है।
- यदि किसी भी तरह से सर्वर डाउन हो जाता है या क्रैश हो जाता है, तो इससे संपूर्ण प्रभावित होगा।

# DSG Support Multi Solution

- यह P2P की तुलना में बहुत महंगा है। ऐसा इसलिए है क्योंकि इसमें अधिक मेमोरी वाले सर्वर की आवश्यकता होती है और साथ ही कई नेटवर्किंग डिवाइस जैसे हब, राउटर, स्विच आदि की भी आवश्यकता होती है।
- उपलब्ध करायी जा रही एनओएस की लागत बहुत अधिक है।

## क्लाइंट/सर्वर आर्किटेक्चर के नुकसान

- यदि सर्वर विफल हो जाता है, तो क्लाइंट सेवाओं तक पहुंच खो सकते हैं।
- सर्वर स्थापित करने के लिए हार्डवेयर और सॉफ्टवेयर में अधिक निवेश की आवश्यकता होती है।
- सर्वर प्रबंधन के लिए कुशल कर्मियों की आवश्यकता होती है।

## कंप्यूटर नेटवर्क आर्किटेक्ट क्या करता है?

कंप्यूटर नेटवर्क आर्किटेक्ट संगठनों के लिए संचार नेटवर्क डिजाइन करने और बनाने के लिए जिम्मेदार होता है। उनके मुख्य कार्य इस प्रकार हैं:

- **नेटवर्क की डिजाइनिंग, मॉडलिंग, परीक्षण और समस्या निवारण** : इसमें नेटवर्क के लिए योजनाएं बनाना, उनका परीक्षण करना और उत्पन्न होने वाली किसी भी समस्या का समाधान करना शामिल है।
- **मौजूदा नेटवर्क का परीक्षण और निरीक्षण** : आपको मौजूदा नेटवर्क की जांच करनी होगी ताकि यह सुनिश्चित हो सके कि वे सही ढंग से काम कर रहे हैं और उन समस्याओं की पहचान करनी होगी जिन्हें ठीक करने की आवश्यकता है।
- **नेटवर्क (हार्डवेयर और सॉफ्टवेयर) को उन्नत करना** : भौतिक उपकरणों और उन पर चलने वाले सॉफ्टवेयर दोनों में सुधार करके नेटवर्क को अद्यतन रखना।
- **सुरक्षा कमजोरियों का विश्लेषण और समाधान करना** : नेटवर्क में किसी भी कमजोरियों की पहचान करना और उन्हें दूर करना ताकि उन्हें साइबर खतरों से सुरक्षित रखा जा सके।
- **तकनीकी दस्तावेज तैयार करना** : नेटवर्क के निर्माण और रखरखाव के लिए विस्तृत मार्गदर्शिकाएँ और निर्देश तैयार करना।
- **हार्डवेयर घटकों की स्थापना और रखरखाव** : नेटवर्क के भौतिक भागों, जैसे राउटर, केबल और एडाप्टर की स्थापना और देखभाल करना।

## Protocol

इंटरनेट प्रोटोकॉल नियमों का एक सेट है जो कंप्यूटर और अन्य उपकरणों को इंटरनेट पर संचार करने की अनुमति देता है। ये प्रोटोकॉल सुनिश्चित करते हैं कि डेटा को विभिन्न प्रणालियों के बीच सही ढंग से भेजा, प्राप्त और समझा जाए। इंटरनेट प्रोटोकॉल के कई प्रकार हैं, जिनमें से प्रत्येक एक विशिष्ट उद्देश्य

# DSG Support Multi Solution

की पूर्ति करता है, जैसे कि फ़ाइलें स्थानांतरित करना, ईमेल भेजना या डेटा सुरक्षित करना। इंटरनेट को कुशलतापूर्वक और सुरक्षित रूप से काम करने के लिए इन प्रोटोकॉल को समझना महत्वपूर्ण है। इस लेख में हम विभिन्न प्रकार के इंटरनेट प्रोटोकॉल को विस्तार से देखेंगे।

## इंटरनेट प्रोटोकॉल क्या है?

जैसा कि हम चर्चा करते हैं कि **इंटरनेट प्रोटोकॉल (आईपी)** नियमों का एक सेट है जो डिवाइस को इंटरनेट पर एक दूसरे के साथ संवाद करने की अनुमति देता है। यह डेटा भेजने के लिए इस्तेमाल किए जाने वाले एड्रेस सिस्टम की तरह है। इंटरनेट से जुड़े हर डिवाइस का एक अनूठा **आईपी पता** होता है जो डेटा को यह जानने में मदद करता है कि उसे कहाँ जाना है और कहाँ से आ रहा है।

## आईपी एड्रेसिंग क्या है?

**IP पता** एक इंटरनेट प्रोटोकॉल पता दर्शाता है। एक अनूठा पता जो नेटवर्क पर डिवाइस की पहचान करता है। यह लगभग इंटरनेट पर या स्थानीय नेटवर्क के माध्यम से भेजे गए डेटा की संरचना को नियंत्रित करने वाले नियमों के एक सेट की तरह है। एक IP पता इंटरनेट को विभिन्न **राउटर**, कंप्यूटर और वेबसाइटों के बीच अंतर करने में मदद करता है। यह एक विशिष्ट नेटवर्क में एक विशिष्ट मशीन पहचानकर्ता के रूप में कार्य करता है और स्रोत और गंतव्य के बीच दृश्य संचार को बेहतर बनाने में मदद करता है।

## इंटरनेट प्रोटोकॉल का कार्य

इंटरनेट प्रोटोकॉल की चरणबद्ध कार्यप्रणाली:

- **डेटा को पैकेट में विभाजित करना** : जब आप इंटरनेट पर जानकारी भेजते हैं, तो IP इसे छोटे-छोटे हिस्सों में विभाजित करता है जिन्हें **पैकेट** कहा जाता है। प्रत्येक पैकेट में डेटा का एक टुकड़ा और पता होता है कि उसे कहाँ जाना है।
- **एड्रेसिंग** : इंटरनेट से जुड़े हर डिवाइस का अपना **आईपी एड्रेस** होता है। यह एड्रेस यह पहचानने में मदद करता है कि डेटा कहाँ से भेजा जा रहा है और इसे कहाँ डिलीवर किया जाना चाहिए।
- **पैकेटों को रूट करना** : जैसे-जैसे पैकेट इंटरनेट पर यात्रा करते हैं, वे **राउटर** नामक कई उपकरणों से गुजरते हैं। ये राउटर पैकेटों को सही गंतव्य की ओर निर्देशित करने में मदद करते हैं, जैसे कि विभिन्न डाकघरों में मेल को कैसे छांटा जाता है।
- **डेटा को पुनः एकत्रित करना** : जब सभी पैकेट गंतव्य पर पहुंच जाते हैं, तो उन्हें मूल संदेश या फ़ाइल को पुनः बनाने के लिए पुनः एकत्रित कर दिया जाता है।
- **गुम पैकेटों को संभालना** : यदि कुछ पैकेट नहीं पहुंचते हैं, तो सिस्टम अनुरोध कर सकता है कि उन्हें दोबारा भेजा जाए, जिससे यह सुनिश्चित हो सके कि पूरा डेटा प्राप्त हो गया है।

# DSG Support Multi Solution

यह प्रक्रिया इंटरनेट पर डेटा को कुशलतापूर्वक स्थानांतरित करने में मदद करती है, चाहे उसे कितनी भी दूर यात्रा करनी हो या कितने भी नेटवर्क से गुजरना हो।

## इंटरनेट प्रोटोकॉल की आवश्यकता

डेटा का प्रेषक और रिसीवर दुनिया के विभिन्न हिस्सों में स्थित विभिन्न नेटवर्क का हिस्सा हैं, जिनकी डेटा ट्रांसफर दरें अलग-अलग हैं। इसलिए, हमें संचार चैनल में साझा किए जा रहे लिंक के डेटा के प्रवाह नियंत्रण और पहुँच नियंत्रण का प्रबंधन करने के लिए प्रोटोकॉल की आवश्यकता है। मान लीजिए कि एक प्रेषक X है जिसकी डेटा संचरण दर 10 एमबीपीएस है। और, एक रिसीवर Y है जिसकी डेटा प्राप्त करने की दर 5 एमबीपीएस है। चूंकि डेटा प्राप्त करने की दर धीमी है, इसलिए संचरण के दौरान कुछ डेटा खो जाएगा। इससे बचने के लिए, रिसीवर Y को प्रेषक X को गति बेमेल के बारे में सूचित करना होगा ताकि प्रेषक X अपनी संचरण दर को समायोजित कर सके। इसी तरह, एक्सेस कंट्रोल उस नोड को तय करता है जो समय के एक विशेष क्षण में संचार चैनल में साझा किए गए लिंक तक पहुंचेगा।

## इंटरनेट प्रोटोकॉल के प्रकार

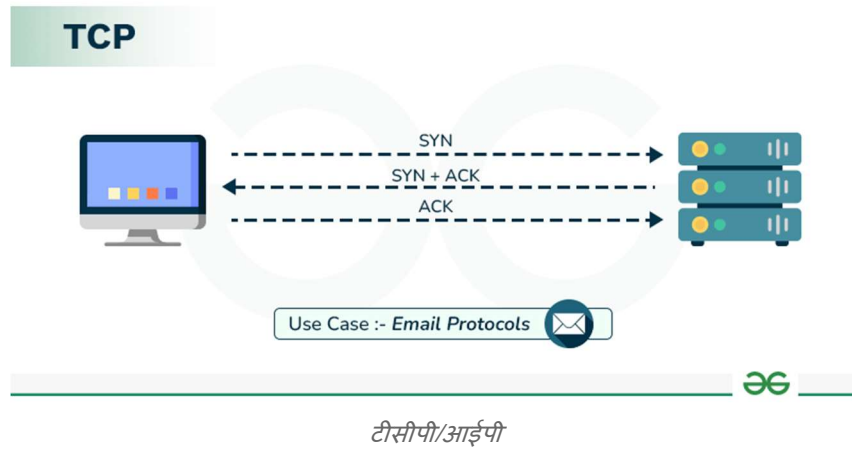
इंटरनेट प्रोटोकॉल विभिन्न प्रकार के होते हैं और इनके अलग-अलग उपयोग होते हैं। इनका उल्लेख नीचे किया गया है:

1. [टीसीपी/आईपी \(ट्रांसमिशन कंट्रोल प्रोटोकॉल/इंटरनेट प्रोटोकॉल\)](#)
2. [एसएमटीपी \(सरल मेल ट्रांसफर प्रोटोकॉल\)](#)
3. [पीपीपी \(पॉइंट-टू-पॉइंट प्रोटोकॉल\)](#)
4. [एफटीपी \(फ़ाइल ट्रांसफर प्रोटोकॉल\)](#)
5. [एसएफटीपी \(सुरक्षित फ़ाइल स्थानांतरण प्रोटोकॉल\)](#)
6. [HTTP \(हाइपर टेक्स्ट ट्रांसफर प्रोटोकॉल\)](#)
7. [HTTPS \(हाइपरटेक्स्ट ट्रांसफर प्रोटोकॉल सिक्योर\)](#)
8. [टेलनेट \(टर्मिनल नेटवर्क\)](#)
9. [POP3 \(पोस्ट ऑफिस प्रोटोकॉल 3\)](#)
10. [आईपीवी 4](#)
11. [आईपीवी6](#)
12. [आईसीएमपी](#)
13. [यूडीपी](#)
14. [आईएमएपी](#)
15. [एसएसएच](#)
16. [धानीमूष](#)

# DSG Support Multi Solution

## 1. टीसीपी/आईपी (ट्रान्समिशन कंट्रोल प्रोटोकॉल/इंटरनेट प्रोटोकॉल)

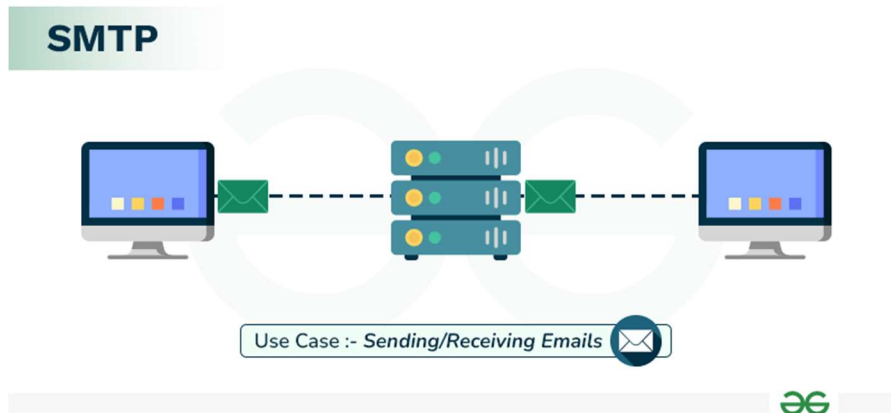
टीसीपी/आईपी में, आईपी प्रोटोकॉल यह सुनिश्चित करता है कि इंटरनेट से जुड़े प्रत्येक कंप्यूटर में एक विशिष्ट सीरियल नंबर हो जिसे आईपी एड्रेस कहा जाता है। टीसीपी निर्दिष्ट करता है कि इंटरनेट पर डेटा का आदान-प्रदान कैसे किया जाता है और इसे आईपी पैकेट में कैसे तोड़ा जाना चाहिए। यह यह भी सुनिश्चित करता है कि पैकेट में संदेश डेटा के स्रोत, संदेश डेटा के गंतव्य, संदेश डेटा को फिर से इकट्ठा करने के क्रम के बारे में जानकारी हो और यह जाँचता है कि संदेश विशिष्ट गंतव्य पर सही तरीके से भेजा गया है या नहीं। टीसीपी को कनेक्शन-उन्मुख प्रोटोकॉल के रूप में भी जाना जाता है।



अधिक जानकारी के लिए कृपया [TCP/IP मॉडल](#) लेख देखें।

## 2. एसएमटीपी (सिंपल मेल ट्रान्सफर प्रोटोकॉल)

SMTP प्रोटोकॉल आउटगोइंग ईमेल भेजने और वितरित करने के लिए महत्वपूर्ण है। यह प्रोटोकॉल मेल के हेडर का उपयोग रिसीवर की ईमेल आईडी प्राप्त करने के लिए करता है और मेल को आउटगोइंग मेल की कतार में दर्ज करता है। और जैसे ही यह मेल को प्राप्तकर्ता ईमेल आईडी पर पहुंचाता है, यह ईमेल को आउटगोइंग सूची से हटा देता है। संदेश या इलेक्ट्रॉनिक मेल टेक्स्ट, वीडियो, छवि आदि पर विचार कर सकता है। यह कुछ संचार सर्वर नियमों को स्थापित करने में मदद करता है।





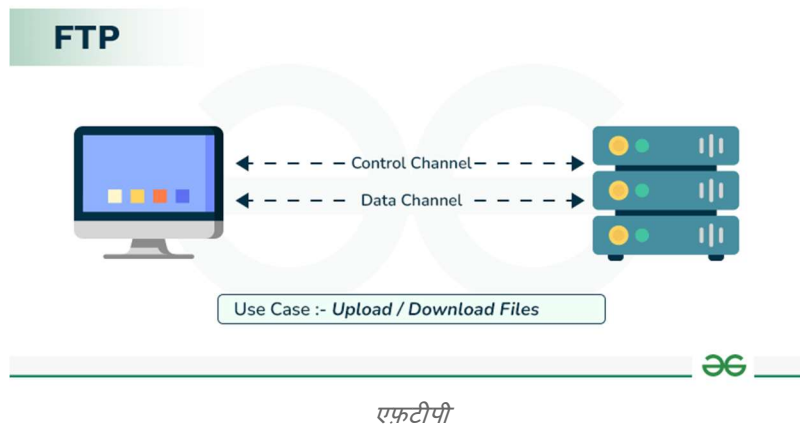
# DSG Support Multi Solution

## 3. पीपीपी (पॉइंट-टू-पॉइंट प्रोटोकॉल)

PPP एक संचार प्रोटोकॉल है जिसका उपयोग दो संचार उपकरणों के बीच सीधा संबंध बनाने के लिए किया जाता है। यह प्रोटोकॉल उन नियमों को परिभाषित करता है जिनका उपयोग करके दो डिवाइस एक दूसरे के साथ प्रमाणीकरण करेंगे और एक दूसरे के साथ सूचना का आदान-प्रदान करेंगे। उदाहरण के लिए, एक उपयोगकर्ता अपने पीसी को इंटरनेट सेवा प्रदाता के सर्वर से जोड़ता है और वह भी PPP का उपयोग करता है। इसी तरह, सीधे संचार के लिए दो राउटर को जोड़ने के लिए यह PPP का उपयोग करता है।

## 4. एफटीपी (फ़ाइल ट्रांसफ़र प्रोटोकॉल)

इस प्रोटोकॉल का उपयोग एक सिस्टम से दूसरे सिस्टम में फ़ाइलों को स्थानांतरित करने के लिए किया जाता है। यह क्लाइंट-सर्वर मॉडल पर काम करता है। जब कोई मशीन किसी दूसरी मशीन से फ़ाइल स्थानांतरण के लिए अनुरोध करती है, तो FTO दोनों के बीच एक कनेक्शन स्थापित करता है और उनके आईडी और पासवर्ड का उपयोग करके एक दूसरे को प्रमाणित करता है। और, मशीनों के बीच वांछित फ़ाइल स्थानांतरण होता है।



## 5. एसएफटीपी (सुरक्षित फ़ाइल स्थानांतरण प्रोटोकॉल)

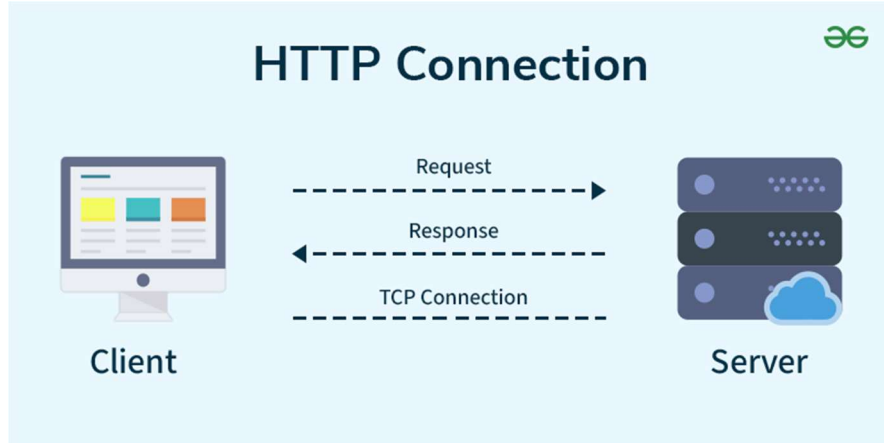
SFTP जिसे SSH FTP के नाम से भी जाना जाता है, सिक्योर शेल (SSH) पर फ़ाइल ट्रांसफ़र प्रोटोकॉल (FTP) को संदर्भित करता है क्योंकि यह ट्रांसमिशन के दौरान कमांड और डेटा दोनों को एन्क्रिप्ट करता है। SFTP SSH के विस्तार के रूप में कार्य करता है और फ़ाइलों और डेटा को एन्क्रिप्ट करता है और फिर उन्हें एक सुरक्षित शेल डेटा स्ट्रीम पर भेजता है। इस प्रोटोकॉल का उपयोग कमांड लाइन से कमांड निष्पादित करते समय अन्य सिस्टम से दूरस्थ रूप से कनेक्ट करने के लिए किया जाता है।

## 6. HTTP (हाइपर टेक्स्ट ट्रांसफर प्रोटोकॉल)

HTTP प्रोटोकॉल का उपयोग इंटरनेट पर हाइपरटेक्स्ट को स्थानांतरित करने के लिए किया जाता है और इसे सूचना हस्तांतरण के लिए www (वर्ल्ड वाइड वेब) द्वारा परिभाषित किया गया है। यह प्रोटोकॉल परिभाषित

# DSG Support Multi Solution

करता है कि सूचना को कैसे प्रारूपित और प्रेषित किया जाना चाहिए। और, यह उन विभिन्न क्रियाओं को भी परिभाषित करता है जो वेब ब्राउज़र को किसी विशेष वेब पेज तक पहुँचने के लिए किए गए कॉल के जवाब में करनी चाहिए। जब भी कोई उपयोगकर्ता अपना वेब ब्राउज़र खोलता है, तो उपयोगकर्ता अप्रत्यक्ष रूप से HTTP का उपयोग करेगा क्योंकि यह वह प्रोटोकॉल है जिसका उपयोग वर्ल्ड वाइड वेब पर टेक्स्ट, चित्र और अन्य मल्टीमीडिया फ़ाइलों को साझा करने के लिए किया जा रहा है।



एचटीटीपी

## 7. HTTPS (हाइपरटेक्स्ट ट्रांसफर प्रोटोकॉल सिक्योर)

HTTPS हाइपरटेक्स्ट ट्रांसफर प्रोटोकॉल (HTTP) का एक विस्तार है। इसका उपयोग एन्क्रिप्शन और प्रमाणीकरण के लिए SSL/TLS प्रोटोकॉल के साथ कंप्यूटर नेटवर्क पर सुरक्षित संचार के लिए किया जाता है। इसलिए, आम तौर पर, एक वेबसाइट में HTTP प्रोटोकॉल होता है, लेकिन अगर वेबसाइट ऐसी है जो क्रेडिट कार्ड विवरण, डेबिट कार्ड विवरण, OTP आदि जैसी कुछ संवेदनशील जानकारी प्राप्त करती है, तो वेबसाइट को अधिक सुरक्षित बनाने के लिए SSL प्रमाणपत्र स्थापित करना आवश्यक है। इसलिए, किसी वेबसाइट पर कोई भी संवेदनशील जानकारी दर्ज करने से पहले, हमें यह जांचना चाहिए कि लिंक HTTPS है या नहीं। यदि यह HTTPS नहीं है, तो यह संवेदनशील जानकारी दर्ज करने के लिए पर्याप्त सुरक्षित नहीं हो सकता है।

## 8. टेलनेट (टर्मिनल नेटवर्क)

TELNET एक मानक TCP/IP प्रोटोकॉल है जिसका उपयोग ISO द्वारा दी गई वर्चुअल टर्मिनल सेवा के लिए किया जाता है। यह एक स्थानीय मशीन को दूसरे से कनेक्ट करने में सक्षम बनाता है। जिस कंप्यूटर को जोड़ा जा रहा है उसे रिमोट कंप्यूटर कहा जाता है और जो कनेक्ट हो रहा है उसे स्थानीय कंप्यूटर कहा जाता है। TELNET ऑपरेशन हमें दूरस्थ कंप्यूटर पर किए जा रहे किसी भी कार्य को स्थानीय कंप्यूटर में प्रदर्शित करने देता है। यह क्लाइंट/सर्वर सिद्धांत पर काम करता है। स्थानीय कंप्यूटर टेलनेट क्लाइंट प्रोग्राम का उपयोग करता है जबकि दूरस्थ कंप्यूटर टेलनेट सर्वर प्रोग्राम का उपयोग करता है।

# DSG Support Multi Solution

## 9. POP3 (पोस्ट ऑफिस प्रोटोकॉल 3)

POP3 का मतलब है पोस्ट ऑफिस प्रोटोकॉल वर्जन 3. इसमें दो मैसेज एक्सेस एजेंट (MAAs) हैं, जिनमें से एक क्लाइंट MAA (मैसेज एक्सेस एजेंट) है और दूसरा सर्वर MAA (मैसेज एक्सेस एजेंट) है जो मेलबॉक्स से संदेशों तक पहुँचने के लिए है। यह प्रोटोकॉल हमें रिसीवर मेल सर्वर पर मेलबॉक्स से रिसीवर के कंप्यूटर पर ईमेल प्राप्त करने और प्रबंधित करने में मदद करता है। यह रिसीवर और रिसीवर मेल सर्वर के बीच निहित है। इसे वन-वे क्लाइंट-सर्वर प्रोटोकॉल भी कहा जा सकता है। POP3 दो पोर्ट यानी पोर्ट 110 और पोर्ट 995 पर काम करता है।

## 10. आईपीवी4

इंटरनेट प्रोटोकॉल का चौथा और शुरू में व्यापक रूप से इस्तेमाल किया जाने वाला संस्करण IPv4 (इंटरनेट प्रोटोकॉल संस्करण 4) कहलाता है। यह इंटरनेट प्रोटोकॉल का सबसे लोकप्रिय संस्करण है और पूरे नेटवर्क में डेटा पैकेट वितरित करने का काम करता है। IPv4 के लिए अधिकतम अद्वितीय पते 4,294,967,296 (2<sup>32</sup>) हैं, जो 32-बिट पतों के उपयोग के कारण संभव हैं। नेटवर्क पता और होस्ट पता प्रत्येक पते के दो घटक हैं। होस्ट पता नेटवर्क के भीतर एक विशेष डिवाइस की पहचान करता है, जबकि नेटवर्क पता उस नेटवर्क की पहचान करता है जिससे होस्ट संबंधित है। "डॉटेड डेसीमल" नोटेशन में, जो IPv4 पतों के लिए मानक है, पते के प्रत्येक ऑक्टेट (8 बिट) को उसके दशमलव मान द्वारा दर्शाया जाता है और एक डॉट (जैसे 192.168.1.1) द्वारा अलग किया जाता है।

## 11. आईपीवी6

इंटरनेट प्रोटोकॉल का सबसे हालिया संस्करण, IPv6, IPv4 प्रोटोकॉल की कमियों को दूर करने के लिए बनाया गया था। IPv4 के 32-बिट पतों के साथ अधिकतम 4.3 बिलियन अद्वितीय पते संभव हैं। इसके विपरीत, IPv6 128-बिट पतों का उपयोग करता है, जो काफी अधिक संख्या में अद्वितीय पतों को सक्षम बनाता है। यह महत्वपूर्ण है क्योंकि IPv4 पते समाप्त हो रहे थे और ऐसे उपकरणों की संख्या बढ़ रही है जिन्हें इंटरनेट एक्सेस की आवश्यकता होती है। इसके अतिरिक्त, IPv6 एकीकृत प्रमाणीकरण और एन्क्रिप्शन जैसी उन्नत सुरक्षा सुविधाएँ प्रदान करता है और साथ ही मोबाइल उपकरणों के लिए बेहतर समर्थन भी प्रदान करता है। IPv6 समर्थन वेबसाइटों और इंटरनेट सेवा प्रदाताओं के बीच फैल गया है, और यह अनुमान लगाया जा रहा है कि यह धीरे-धीरे IPv4 को मुख्य इंटरनेट प्रोटोकॉल के रूप में विस्थापित कर देगा।

अधिक जानकारी के लिए कृपया [IPv4 और IPv6 के बीच अंतर](#) लेख देखें।

## 12. आईसीएमपी

ICMP (इंटरनेट कंट्रोल मैसेज प्रोटोकॉल) एक नेटवर्क प्रोटोकॉल है जिसका उपयोग त्रुटि संदेश और नेटवर्क स्थितियों के बारे में परिचालन जानकारी भेजने के लिए किया जाता है। यह इंटरनेट प्रोटोकॉल (IP) सूट का

# DSG Support Multi Solution

एक अभिन्न अंग है और इसका उपयोग नेटवर्क कनेक्टिविटी के साथ समस्याओं का निदान और समस्या निवारण करने में मदद करने के लिए किया जाता है। ICMP संदेश आमतौर पर नेटवर्क डिवाइस, जैसे राउटर द्वारा डेटाग्राम को अग्रेषित करने में आने वाली त्रुटियों या असाधारण स्थितियों के जवाब में उत्पन्न होते हैं। ICMP संदेशों के कुछ उदाहरणों में शामिल हैं:

- इको अनुरोध और इको उत्तर (पिंग)
- गंतव्य पहुंच - योग्य नहीं है
- समय बीत गया
- पुनर्निर्देशन

ICMP का उपयोग नेटवर्क प्रबंधन उपकरणों द्वारा होस्ट की पहुंच क्षमता का परीक्षण करने और पैकेट के स्रोत से गंतव्य तक और वापस यात्रा करने के लिए राउंड-ट्रिप समय को मापने के लिए भी किया जा सकता है। यह ध्यान दिया जाना चाहिए कि ICMP एक सुरक्षित प्रोटोकॉल नहीं है, इसका उपयोग [DDoS](#) प्रवर्धन जैसे कुछ प्रकार के नेटवर्क हमलों में किया जा सकता है।

## 13. यूडीपी

UDP (यूजर डेटाग्राम प्रोटोकॉल) एक कनेक्शन रहित, अविश्वसनीय ट्रांसपोर्ट लेयर प्रोटोकॉल है। TCP के विपरीत, यह डेटा संचारित करने से पहले डिवाइस के बीच एक विश्वसनीय कनेक्शन स्थापित नहीं करता है, और यह गारंटी नहीं देता है कि डेटा पैकेट उसी क्रम में प्राप्त किए जाएंगे जिस क्रम में उन्हें भेजा गया था या वे प्राप्त किए जाएंगे ही। इसके बजाय, UDP बिना किसी त्रुटि जाँच या प्रवाह नियंत्रण के बस डेटा के पैकेट को गंतव्य पर भेजता है। UDP का उपयोग आम तौर पर स्ट्रीमिंग वीडियो और ऑडियो, ऑनलाइन गेमिंग और [VoIP \(वॉयस ओवर इंटरनेट प्रोटोकॉल\)](#) जैसे वास्तविक समय के अनुप्रयोगों के लिए किया जाता है, जहाँ थोड़ी मात्रा में डेटा खोना स्वीकार्य है और कम विलंबता महत्वपूर्ण है। UDP TCP से तेज़ है क्योंकि इसमें ओवरहेड कम है। इसे कनेक्शन स्थापित करने की आवश्यकता नहीं है, इसलिए यह तुरंत डेटा पैकेट भेज सकता है। इसे और अधिक भेजने से पहले इस बात की पुष्टि के लिए प्रतीक्षा करने की भी आवश्यकता नहीं है कि डेटा प्राप्त हो गया है, इसलिए यह उच्च दर पर डेटा संचारित कर सकता है।

## 14. आईएमएपी

IMAP (इंटरनेट मैसेज एक्सेस प्रोटोकॉल) एक प्रोटोकॉल है जिसका उपयोग मेल सर्वर से ईमेल प्राप्त करने के लिए किया जाता है। यह उपयोगकर्ताओं को स्थानीय डिवाइस पर डाउनलोड करने के बजाय सर्वर पर अपने ईमेल तक पहुँचने और उन्हें प्रबंधित करने की अनुमति देता है। इसका मतलब है कि उपयोगकर्ता कई डिवाइस से अपने ईमेल तक पहुँच सकता है और ईमेल सभी डिवाइस में सिंक हो जाएंगे। IMAP [POP3 \(पोस्ट ऑफिस प्रोटोकॉल संस्करण 3\)](#) की तुलना में अधिक लचीला है क्योंकि यह उपयोगकर्ताओं को सर्वर

# DSG Support Multi Solution

पर अपने ईमेल तक पहुँचने और उन्हें व्यवस्थित करने की अनुमति देता है, और कई उपयोगकर्ताओं को एक ही मेलबॉक्स तक पहुँचने की अनुमति भी देता है।

## 15. एसएसएच

SSH (सिक्योर शेल) एक प्रोटोकॉल है जिसका उपयोग सुरक्षित रिमोट लॉगिन और अन्य सुरक्षित नेटवर्क सेवाओं के लिए किया जाता है। यह सर्वर, नेटवर्क डिवाइस और अन्य कंप्यूटर सिस्टम को दूरस्थ रूप से एक्सेस करने और प्रबंधित करने का एक सुरक्षित और एन्क्रिप्टेड तरीका प्रदान करता है। SSH उपयोगकर्ता को प्रमाणित करने और संचारित किए जा रहे डेटा को एन्क्रिप्ट करने के लिए सार्वजनिक-कुंजी क्रिप्टोग्राफी का उपयोग करता है, जो इसे टेलनेट जैसे पारंपरिक रिमोट लॉगिन प्रोटोकॉल की तुलना में बहुत अधिक सुरक्षित बनाता है। SSH SCP (सिक्योर कॉपी) और [SFTP \(सिक्योर फाइल ट्रांसफर प्रोटोकॉल\)](#) प्रोटोकॉल का उपयोग करके सुरक्षित फ़ाइल ट्रांसफर की भी अनुमति देता है। इसका व्यापक रूप से यूनिक्स-आधारित ऑपरेटिंग सिस्टम में उपयोग किया जाता है और यह विंडोज के लिए भी उपलब्ध है। इसका उपयोग आमतौर पर सिस्टम प्रशासक, डेवलपर्स और अन्य तकनीकी उपयोगकर्ताओं द्वारा सर्वर और अन्य नेटवर्क डिवाइस को दूरस्थ रूप से एक्सेस करने और प्रबंधित करने के लिए किया जाता है।

## 16. गोफर

गोफर एक प्रकार का फ़ाइल पुनर्प्राप्ति प्रोटोकॉल है जो फ़ाइलों के आसान प्रबंधन, पुनर्प्राप्ति और खोज के लिए कुछ विवरण के साथ डाउनलोड करने योग्य फ़ाइलें प्रदान करता है। सभी फ़ाइलों को एक दूरस्थ कंप्यूटर पर स्तरीकृत तरीके से व्यवस्थित किया जाता है। यह एक पुराना प्रोटोकॉल है और आजकल इसका ज़्यादा इस्तेमाल नहीं होता है।

## Switch

राउटर एक नेटवर्किंग डिवाइस है जो नेटवर्क लेयर यानी ISO-OSI मॉडल की तीसरी लेयर पर काम करता है और यह मल्टीपोर्ट डिवाइस है। यह नेटवर्क के बीच डेटा प्रवाह प्रदान करने के लिए नेटवर्क के बीच एक सरल कनेक्शन स्थापित करता है। राउटर पैकेट के रूप में डेटा ट्रांसफर करता है जिसका उपयोग LAN के साथ-साथ MAN में भी किया जाता है।

यह नेटवर्क लेयर 3 पर काम करता है और इसका उपयोग LAN, MAN और WAN में किया जाता है। यह IP एड्रेस को स्टोर करता है और खुद ही एड्रेस को बनाए रखता है।

## राउटर का कार्य

- कई नेटवर्क वाले उपकरण, जिनमें पी.सी., टैबलेट, प्रिंटर और अन्य वस्तुएं शामिल हैं, को घर या कार्यस्थल में राउटर का उपयोग करके इंटरनेट से जोड़ा जा सकता है और नेटवर्क बनाया जा सकता है।

# DSG Support Multi Solution

- इन उपकरणों और इंटरनेट के बीच संचार को सुविधाजनक बनाने के लिए, राउटर पहले मॉडेम को अन्य उपकरणों से जोड़ता है।
- निर्दिष्ट आईपी पते वाले डेटा पैकेटों को राउटरों द्वारा नेटवर्कों के पार या नेटवर्कों के भीतर रूट और प्रेषित किया जाता है।
- यह इंटरनेट से जुड़े प्रत्येक उपकरण को एक स्थानीय आईपी पता प्रदान करके ऐसा करता है; इससे उचित गंतव्य की गारंटी मिलती है, तथा नेटवर्क में डेटा के खो जाने से बचाव होता है।
- एक बार इष्टतम और सबसे तेज़ पथ निर्धारित हो जाने पर, डेटा पैकेट उस पथ से नेटवर्क से जुड़े उपकरणों तक भेजे जाते हैं।

## राउटर के प्रकार

### 1. वायरलेस राउटर

- चूंकि नेटवर्किंग उपकरणों से जुड़ने के लिए इन्हें तारों या केबलों की आवश्यकता नहीं होती, इसलिए वायरलेस राउटर घरों और कंपनियों में सबसे व्यापक रूप से उपयोग किए जाने वाले राउटर हैं।
- आईडी और पासवर्ड के साथ, केवल अधिकृत उपयोगकर्ता ही नेटवर्क तक पहुंच सकते हैं, जिससे सुरक्षित कनेक्शन सुनिश्चित होता है।
- निर्दिष्ट सीमा के भीतर कोई भी n उपयोगकर्ता वायरलेस राउटर का उपयोग करके इंटरनेट का उपयोग कर सकता है।

### 2. वायर्ड राउटर

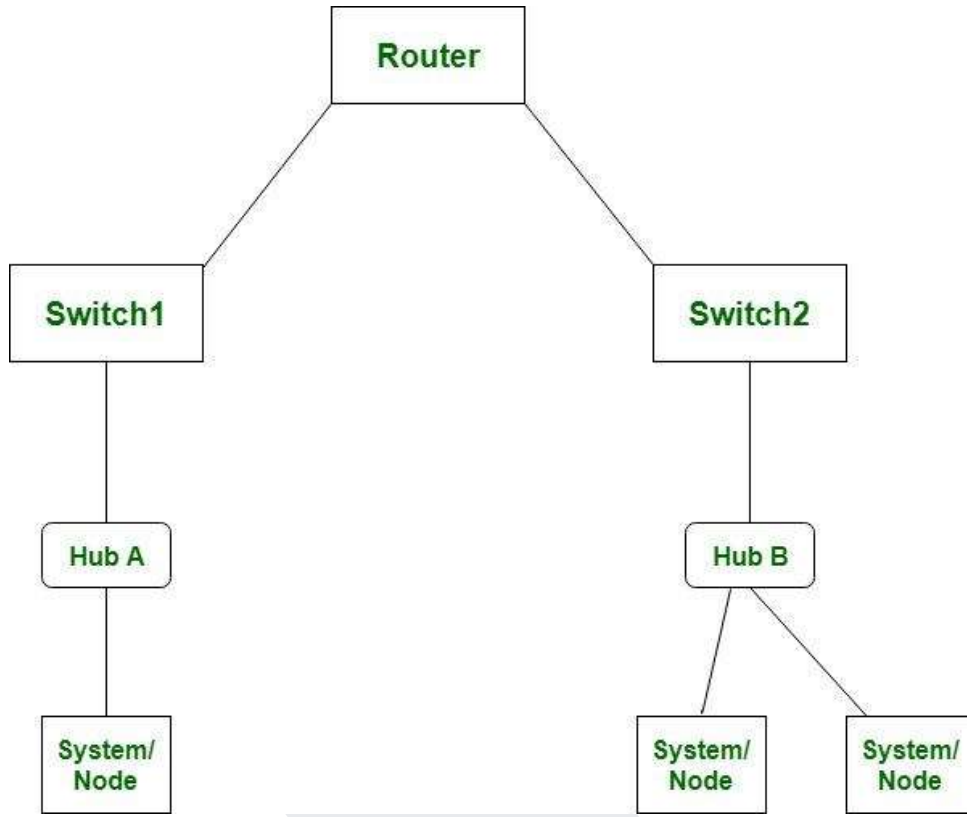
- जैसा कि नाम से ही स्पष्ट है, नेटवर्क डिवाइसों से कनेक्ट करने के लिए इसे एक तार या केबल की आवश्यकता होती है।
- इन राउटरों का उपयोग ज्यादातर छोटी कंपनियों या स्कूलों में पीसी को ईथरनेट केबल से जोड़ने के लिए किया जाता है।
- इसमें वाई-फाई एक्सेस प्वाइंट भी है तथा वीओआईपी (वॉयस-ओवर-इंटरनेट प्रोटोकॉल) तकनीक के माध्यम से मोबाइल फोन को इससे जोड़ा जा सकता है।

## राउटर के लाभ

- अधिकांश नेटवर्किंग डिवाइस किसी भी समय तारों की उलझन की चिंता किए बिना वायरलेस राउटर से कनेक्ट हो सकते हैं।
- यह ईथरनेट, WLAN और वाई-फाई सहित कई नेटवर्क आर्किटेक्चर का उपयोग करके कनेक्शन स्थापित कर सकता है।
- यह पासवर्ड-संरक्षित, अत्यंत सुरक्षित नेटवर्क पहुंच प्रदान करता है।
- यह नेटवर्क ट्रैफिक को कम करने के लिए टकराव फंक्शन का उपयोग करता है।

# DSG Support Multi Solution

- इंटेलिजेंस और रूटिंग टेबल के उपयोग से, यह डेटा पैकेट्स को सबसे कुशल पथ के माध्यम से सही स्थान पर पहुंचाता है।



## Router

यह एक पॉइंट-टू-पॉइंट संचार उपकरण है। मूल रूप से, यह एक प्रकार का पुल है जो बेहतर कनेक्शन प्रदान करता है। यह एक प्रकार का उपकरण है जो उस समय की आवश्यकताओं के अनुसार कनेक्शन सेट करता है और रोकता है। यह फ्लडिंग, फिल्टरिंग और फ्रेम ट्रांसमिशन जैसी कई विशेषताओं के साथ आता है।

## स्विच का कार्य

- जहां तक हमारी जानकारी है, प्रत्येक नेटवर्किंग डिवाइस का एक विशिष्ट मीडिया एक्सेस कंट्रोल (MAC) पता होता है।
- एक IP पैकेट एक डिवाइस या कंप्यूटर से दूसरे डिवाइस या कंप्यूटर पर एक स्विच के माध्यम से भेजा जाता है, जो IP पैकेट को एक फ्रेम और उसके स्रोत और गंतव्य MAC पते के साथ समाहित कर देता है।
- लक्ष्य डिवाइस पर पहुंचने पर फ्रेम को हटा दिया जाता है, जो तब आईपी पैकेट प्राप्त करता है और केवल उस डिवाइस से कनेक्ट कर सकता है जो दर्ज किए गए गंतव्य मैक पते से मेल खाता है।



# DSG Support Multi Solution

## स्विच के प्रकार

### 1. प्रबंधित स्विच

क्योंकि हम अपने नेटवर्क को पूरी तरह से पूरा करने के लिए प्रबंधित स्विच को आसानी से डिज़ाइन कर सकते हैं, वे अतिरिक्त क्षमताएँ प्रदान करते हैं और अप्रबंधित स्विच की तुलना में अधिक लचीले होते हैं। वे अधिक सुरक्षित भी हैं। परिणामस्वरूप, हम अधिक नियंत्रण करने, अपने नेटवर्क की बेहतर सुरक्षा करने और नेटवर्क उपयोगकर्ताओं को प्रदान की जाने वाली सेवा की गुणवत्ता बढ़ाने में सक्षम होंगे।

### 2. अप्रबंधित स्विच

अनमैनेज्ड स्विच के लिए बेसिक कनेक्शन प्राथमिक उपयोग है। ये आम तौर पर छोटे नेटवर्क या ऐसे स्थानों पर पाए जाते हैं जहाँ मामूली संख्या में अतिरिक्त पोर्ट की आवश्यकता होती है, जैसे कि कॉन्फ्रेंस रूम, लैब या निवास। अनमैनेज्ड स्विच के काम करने के लिए प्लग इन करना ही पर्याप्त है; किसी कॉन्फिगरेशन की आवश्यकता नहीं है।

## स्विच के लाभ

- यह नेटवर्क की उपलब्ध बैंडविड्थ को बढ़ाता है।
- इसका डिवाइस या वर्कस्टेशन से सीधा संबंध हो सकता है।
- नेटवर्क के प्रदर्शन में सुधार करता है।
- स्विच नेटवर्क में फ्रेम टकराव को कम करते हैं क्योंकि वे प्रत्येक नेटवर्क के टकराव डोमेन को विकसित करने में मदद करते हैं।
- यह विशिष्ट होस्टों, जैसे कि पीसी, पर दबाव को कम करने में सहायता करता है

राउटर का मुख्य उद्देश्य विभिन्न नेटवर्क को एक साथ जोड़ना है और यह नेटवर्क लेयर में काम करता है। जबकि स्विच का मुख्य उद्देश्य विभिन्न उपकरणों को एक साथ जोड़ना है और यह डेटा लिंक लेयर में काम करता है।

आइये राउटर और स्विच के बीच अंतर देखें:

Router	Switch
राउटर का मुख्य उद्देश्य विभिन्न नेटवर्कों को एक साथ जोड़ना है।	जबकि स्विच का मुख्य उद्देश्य विभिन्न उपकरणों को एक साथ जोड़ना है।
यह <u>नेटवर्क परत</u> में काम करता है .	जबकि यह <u>डेटा लिंक परत</u> में काम करता है .

# DSG Support Multi Solution

Router	Switch
राउटर का उपयोग <a href="#">LAN</a> के साथ-साथ <a href="#">MAN</a> द्वारा भी किया जाता है ।	जबकि स्विच का उपयोग केवल LAN द्वारा किया जाता है।
राउटर के माध्यम से डेटा पैकेट के रूप में भेजा जाता है।	जबकि स्विच के माध्यम से डेटा फ्रेम के रूप में भेजा जाता है।
राउटर में टकराव कम होता है।	जबकि पूर्ण द्वाैध स्विच में कोई टकराव नहीं होता है।
राउटर <a href="#">NAT</a> के साथ संगत है .	हालांकि यह NAT के साथ संगत नहीं है।
<a href="#">राउटर</a> स्विच की तुलना में अपेक्षाकृत अधिक महंगा उपकरण है।	<a href="#">स्विच हब</a> से महंगा उपकरण है , लेकिन राउटर से सस्ता है।
वायरलेस के लिए अधिकतम गति 1-10 एमबीपीएस है और वायर्ड कनेक्शन के लिए अधिकतम गति 100 एमबीपीएस है।	अधिकतम गति 10Mbps से 100Mbps है।
राउटर को कनेक्ट होने के लिए कम से कम दो नेटवर्क की आवश्यकता होती है।	स्विच को कनेक्ट करने के लिए कम से कम एकल नेटवर्क की आवश्यकता होती है।
रूटिंग के प्रकार हैं: <a href="#">अनुकूली</a> और <a href="#">गैर-अनुकूली रूटिंग</a> ।	स्विचिंग के प्रकार हैं: <a href="#">सर्किट</a> , <a href="#">पैकेट</a> और <a href="#">संदेश</a> स्विचिंग।

## Internet Address

IP पता या इंटरनेट प्रोटोकॉल पता, संख्याओं की एक अनूठी स्ट्रिंग है जो कंप्यूटर नेटवर्क से जुड़े प्रत्येक डिवाइस को सौंपी जाती है जो संचार के लिए इंटरनेट प्रोटोकॉल का उपयोग करता है। यह एक पहचानकर्ता

# DSG Support Multi Solution

के रूप में कार्य करता है जो डिवाइस को नेटवर्क पर डेटा भेजने और प्राप्त करने की अनुमति देता है, यह सुनिश्चित करता है कि यह डेटा सही गंतव्य तक पहुँचता है।

इंटरनेट पर मौजूद हर डिवाइस को एक घर की तरह कल्पना करें। इन घरों में से किसी एक में रहने वाले दोस्त को पत्र भेजने के लिए आपको उनके घर का पता चाहिए। डिजिटल दुनिया में, इस घर के पते को हम **IP (इंटरनेट प्रोटोकॉल) पता** कहते हैं। यह संख्याओं की एक अनूठी स्ट्रिंग है जिसे अवधि (IPv4) या कोलन (IPv6) द्वारा अलग किया जाता है जो इंटरनेट या स्थानीय नेटवर्क से जुड़े प्रत्येक डिवाइस की पहचान करता है।

## आईपी एड्रेस के प्रकार

आईपी पतों को उनकी संरचना, उद्देश्य और जिस प्रकार के नेटवर्क में वे उपयोग किए जाते हैं, उसके आधार पर कई तरीकों से वर्गीकृत किया जा सकता है। यहां आईपी पतों के विभिन्न वर्गीकरणों का विवरण दिया गया है:

### 1. एड्रेसिंग स्कीम के आधार पर (IPv4 बनाम IPv6)

#### आईपीवी4:

यह IP पते का सबसे आम रूप है। इसमें डॉट्स द्वारा अलग किए गए संख्याओं के चार सेट होते हैं। उदाहरण के लिए, 192.158.1.38। संख्याओं का प्रत्येक सेट 0 से 255 तक हो सकता है। यह प्रारूप 4 बिलियन से अधिक अद्वितीय पतों का समर्थन कर सकता है। यहाँ बताया गया है कि संरचना कैसे विभाजित की गई है:

- **चार अष्टक** : प्रत्येक अष्टक आठ बिट या एक बाइट का प्रतिनिधित्व करता है, और 0 से 255 तक मान ले सकता है। यह सीमा आठ बिट के संभावित संयोजनों ( $2^8 = 256$  संयोजन) से प्राप्त होती है।
- **IPv4 पते का उदाहरण** : 192.168.1.1
  - 192 पहला अष्टक है
  - 168 दूसरा अष्टक है
  - 1 तीसरा अष्टक है
  - 1 चौथा अष्टक है

आईपी एड्रेस का प्रत्येक भाग नेटवर्क कॉन्फिगरेशन के विभिन्न पहलुओं को इंगित कर सकता है, नेटवर्क से लेकर उस नेटवर्क के भीतर विशिष्ट डिवाइस तक। ज्यादातर मामलों में, एड्रेस का नेटवर्क हिस्सा पहले एक से तीन ऑक्टेट्स द्वारा दर्शाया जाता है, जबकि शेष भाग होस्ट (डिवाइस) की पहचान करता है।

# DSG Support Multi Solution

आईपीवी6:

IPv6 पते IPv4 पतों की कमी से निपटने के लिए बनाए गए थे। वे 32 के बजाय 128 बिट्स का उपयोग करते हैं, जिससे संभावित पतों की संख्या बहुत अधिक हो जाती है। इन पतों को चार हेक्साडेसिमल अंकों के आठ समूहों के रूप में व्यक्त किया जाता है, प्रत्येक समूह 16 बिट्स का प्रतिनिधित्व करता है। समूहों को कोलन द्वारा अलग किया जाता है।

• **IPv6 पते का उदाहरण** : 2001:0db8:85a3:0000:0000:8a2e:0370:7334

- प्रत्येक समूह (जैसे **2001** , **0db8** , **85a3** , आदि) पते के 16-बिट ब्लॉक का प्रतिनिधित्व करता है।

विस्तृत जानकारी के लिए, इस लेख का संदर्भ लें - [IPv4 बनाम IPv6](#)

## 2. उपयोग के आधार पर (सार्वजनिक बनाम निजी)

सार्वजनिक आईपी पते

इंटरनेट तक सीधे पहुंचने वाले हर डिवाइस को एक सार्वजनिक आईपी पता दिया जाता है। यह पता पूरे इंटरनेट पर अद्वितीय है। यहाँ सार्वजनिक आईपी पतों की मुख्य विशेषताएँ और उपयोग दिए गए हैं:

- **विशिष्टता:** प्रत्येक सार्वजनिक आईपी पता वैश्विक रूप से अद्वितीय होता है। इंटरनेट पर किसी भी दो डिवाइस का एक ही समय में एक ही सार्वजनिक आईपी पता नहीं हो सकता।
- **पहुंच:** सार्वजनिक आईपी पते वाले डिवाइसों तक इंटरनेट पर कहीं से भी सीधे पहुंचा जा सकता है, बशर्ते कि कोई फ़ायरवॉल या सुरक्षा सेटिंग पहुंच को अवरुद्ध न करे।
- **ISP द्वारा असाइन किया गया:** सार्वजनिक IP पते इंटरनेट सेवा प्रदाताओं (ISP) द्वारा असाइन किए जाते हैं। जब आप किसी ISP के माध्यम से इंटरनेट से कनेक्ट होते हैं, तो आपके डिवाइस या राउटर को एक सार्वजनिक IP पता प्राप्त होता है।
- **प्रकार:** सार्वजनिक आईपी पते स्थैतिक (किसी डिवाइस को स्थायी रूप से निर्दिष्ट) या गतिशील (अस्थायी रूप से निर्दिष्ट और समय के साथ बदल सकते हैं) हो सकते हैं।

**उदाहरण उपयोग:** सार्वजनिक आईपी पते आमतौर पर वेबसाइट होस्ट करने वाले सर्वर, ईमेल सर्वर या किसी भी डिवाइस के लिए उपयोग किए जाते हैं, जिसे इंटरनेट से एक्सेस करने की आवश्यकता होती है। उदाहरण के लिए, यदि आप घर पर अपने सर्वर पर कोई वेबसाइट होस्ट करते हैं, तो आपके ISP को आपके सर्वर को एक सार्वजनिक आईपी पता प्रदान करना होगा ताकि दुनिया भर के उपयोगकर्ता आपकी साइट तक पहुंच सकें।

# DSG Support Multi Solution

## निजी आईपी पते

निजी IP पते निजी नेटवर्क (जैसे होम नेटवर्क, ऑफिस नेटवर्क, आदि) के भीतर उपयोग किए जाते हैं और इंटरनेट पर रूट करने योग्य नहीं होते हैं। इसका मतलब यह है कि निजी IP पते वाले डिवाइस नेटवर्क एड्रेस ट्रांसलेशन (NAT) करने वाले राउटर जैसे ट्रांसलेटिंग मैकेनिज्म के बिना इंटरनेट पर डिवाइस के साथ सीधे संवाद नहीं कर सकते हैं। मुख्य विशेषताओं में शामिल हैं:

- **वैश्विक रूप से अद्वितीय नहीं:** निजी आईपी पते को केवल अपने नेटवर्क के भीतर अद्वितीय होना आवश्यक है। विभिन्न निजी नेटवर्क बिना किसी संघर्ष के आईपी पतों की एक ही श्रेणी का उपयोग कर सकते हैं।
- **स्थानीय संचार:** इन पतों का उपयोग एक ही नेटवर्क के भीतर उपकरणों के बीच संचार के लिए किया जाता है। इनका उपयोग इंटरनेट पर उपकरणों के साथ सीधे संचार करने के लिए नहीं किया जा सकता है।
- **परिभाषित श्रेणियाँ:** इंटरनेट असाइन्ड नंबरर्स अथॉरिटी (IANA) ने निजी उपयोग के लिए विशिष्ट IP पता श्रेणियाँ आरक्षित की हैं:
  - **आईपीवी4:** 10.0.0.0 से 10.255.255.255, 172.16.0.0 से 172.31.255.255, 192.168.0.0 से 192.168.255.255
  - **IPv6:** FD या FC से शुरू होने वाले पते

**उदाहरण उपयोग:** एक सामान्य होम नेटवर्क में, राउटर आरक्षित रेंज से प्रत्येक डिवाइस (जैसे स्मार्टफोन, लैपटॉप, स्मार्ट टीवी) को निजी आईपी पते प्रदान करता है। ये डिवाइस एक दूसरे के साथ और राउटर के साथ संचार करने के लिए अपने निजी आईपी का उपयोग करते हैं। राउटर इन डिवाइस को अपने सार्वजनिक आईपी पते का उपयोग करके इंटरनेट तक पहुंचने की अनुमति देने के लिए NAT का उपयोग करता है।

## 3. असाइनमेंट विधि के आधार पर (स्थिर बनाम गतिशील)

### स्थैतिक आईपी पते:

- इन्हें स्थायी रूप से किसी डिवाइस को सौंपा जाता है, जो आमतौर पर उन सर्वरों या डिवाइसों के लिए महत्वपूर्ण होता है जिन्हें एक स्थिर पते की आवश्यकता होती है।
- नेटवर्क सेवाओं के लिए विश्वसनीय, जिनके लिए नियमित पहुंच की आवश्यकता होती है जैसे वेबसाइट, दूरस्थ प्रबंधन।

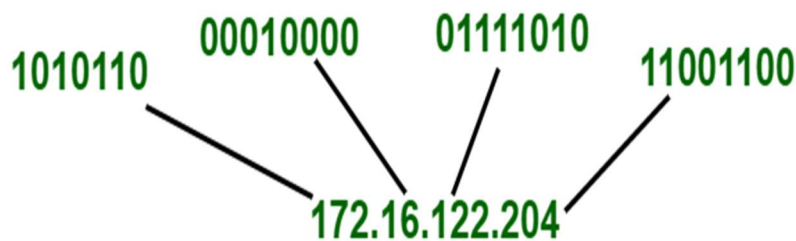
# DSG Support Multi Solution

गतिशील आईपी पते:

- डायनेमिक होस्ट कॉन्फिगरेशन प्रोटोकॉल (DHCP) द्वारा उपलब्ध पतों के पूल से अस्थायी रूप से असाइन किया गया।
- प्रदाताओं के लिए लागत प्रभावी और कुशल, तथा उपभोक्ता उपकरणों के लिए उपयुक्त, जिन्हें स्थायी पते की आवश्यकता नहीं होती।

## Domain Name

इंटरनेट पर प्रत्येक कंप्यूटर का एक पता होता है जो प्रकृति में अद्वितीय होता है। यह संख्याओं की एक स्ट्रिंग है और इसे आईपी एड्रेस कहा जाता है। एक दूसरे के साथ संवाद करने के लिए, कंप्यूटर अपने [आईपी पते](#) के माध्यम से दूसरे कंप्यूटर की पहचान करते हैं। इसे या तो बिंदीदार दशमलव संकेतन या बाइनरी दशमलव संकेतन में दर्शाया जाता है। उदाहरण: पता 172.16.122.204 जब बिंदीदार-दशमलव संकेतन में इन जैसे दर्शाया जाता है और इसे बाइनरी संकेतन में परिवर्तित किया जा सकता है। रूपांतरण के बाद, यह 10101100 00010000 01111010 11001100 हो जाता है। लेकिन इंसानों के लिए इस आईपी एड्रेस को याद रखना मुश्किल है। [इसलिए](#) इंटरनेट पर आसानी से लोकेशन ढूँढने के लिए DNS का आविष्कार किया गया। DNS का मतलब डोमेन नेम सर्वर है। यह कंप्यूटर नेटवर्क पर एक अद्वितीय आईपी पते के लिए एक संकेतक की तरह है। एक सादृश्य के रूप में **डोमेन नाम को पता** और [DNS को](#) इंटरनेट की पता पुस्तिका के रूप में माना जा सकता है।



उदाहरण-1:

आइए डोमेन नाम के लिए एक उदाहरण पर विचार करें;

www.google.com, www.yahoo.com

# DSG Support Multi Solution

इसमें “yahoo.com” को डोमेन नाम कहा जाता है।

“www.” ब्राउज़र को उस डोमेन के लिए वर्ल्ड वाइड वेब इंटरफेस देखने के लिए कहता है।

उपरोक्त उदाहरण से यह स्पष्ट है कि डोमेन नाम को याद रखना आईपी पते की तुलना में आसान है।

## उदाहरण-2:

मान लीजिए कि www.yahoo.com का IP पता 69.147.76.15 है। IP पता 69.147.76.15 की तुलना में www.yahoo.com को याद रखना आसान है।

इस प्रकार, हम इस तरह कह सकते हैं; डोमेन नाम एक आईपी पते से जुड़े अक्षरों की स्ट्रिंग को संदर्भित करता है और DNS एक तंत्र है जिसका उपयोग आईपी पते को डोमेन नाम में बदलने के लिए किया जाता है।

## डोमेन नाम के प्रकार:

DNS ने सभी डोमेन नामों को एक पदानुक्रमित संरचना में व्यवस्थित किया है। इस पदानुक्रम के शीर्ष पर विभिन्न शीर्ष-स्तरीय डोमेन आते हैं, उसके बाद दूसरे और तीसरे स्तर के डोमेन और उप-डोमेन आते हैं।

इन सभी प्रकार के डोमेन नाम इस प्रकार सूचीबद्ध हैं -

### शीर्ष स्तरीय डोमेन (TLD):

शीर्ष स्तरीय डोमेन इंटरनेट की DNS संरचना में सबसे ऊंचे स्तर पर होते हैं। इसे कभी-कभी एक्सटेंशन भी कहा जाता है। इसे आगे देश कोड TLD और जेनेरिक TLD में वर्गीकृत किया जाता है, जिसका देश इस प्रकार वर्णित है -

#### • देश कोड शीर्ष स्तरीय डोमेन (ccTLDs):

इसमें दो-अक्षर वाले डोमेन होते हैं जिनमें हर देश के लिए एक प्रविष्टि शामिल होती है। उदाहरण -

भारत के लिए .in, ऑस्ट्रेलिया के लिए .au, संयुक्त राष्ट्र के लिए .us, जापान के लिए .jp आदि।

स्थानीय दर्शकों को लक्षित करने के लिए इसका उपयोग कंपनियों और संगठनों द्वारा किया जाता है।

केवल देश के निवासियों को ही उनके निर्दिष्ट ccTLD में प्रवेश करने की अनुमति है, लेकिन अब कुछ

देशों ने अपने देश के बाहर के उपयोगकर्ताओं को उनके संबंधित ccTLD को पंजीकृत करने की अनुमति दी है।

#### • जेनेरिक टॉप लेवल डोमेन (gTLDs):

ये सभी उपयोगकर्ताओं के लिए पंजीकरण के लिए खुले हैं, चाहे उनकी नागरिकता, निवास या आयु कुछ

भी हो। कुछ gTLDs हैं: वाणिज्यिक साइटों के लिए .com, नेटवर्क कंपनियों के लिए .net, व्यवसाय के

लिए .biz, संगठनों के लिए .org, शिक्षा के लिए .edu।



# DSG Support Multi Solution

ऐसे कई अन्य स्तर हैं जो TLD से नीचे हैं -

## दूसरा स्तर:

यह DNS पदानुक्रम में TLD के ठीक नीचे है। इसे लेबल भी कहा जाता है। उदाहरण: .co.in में, .co ccTLD में .in के अंतर्गत दूसरा-स्तरीय डोमेन है।

## तीसरा स्तर:

यह दूसरे स्तर से ठीक नीचे है। उदाहरण: yahoo.co.in में, .yahoo दूसरे स्तर के डोमेन .co के अंतर्गत तीसरा स्तर डोमेन है जो .in ccTLD के अंतर्गत है।

## उप-डोमेन:

यह DNS पदानुक्रम में उच्च डोमेन नाम का हिस्सा है। उदाहरण: yahoo.com में .com डोमेन का उप-डोमेन शामिल है, और login.yahoo.com में .yahoo.com डोमेन का उप-डोमेन शामिल है।

## डोमेन नाम के लाभ:

- उपयोगकर्ता को आईपी पता याद रखने की आवश्यकता नहीं है।
- अधिक विश्वसनीय एवं सुरक्षित।

## डोमेन नाम के नुकसान:

- आईपी एड्रेस कई कारणों से बदलता है, इसके कारण कंप्यूटर का आईपी एड्रेस बदल जाता है लेकिन डीएनएस ने पिछले आईपी को कैश कर लिया होगा जिससे हमें गलत जानकारी मिल जाएगी।

## World Wide Web

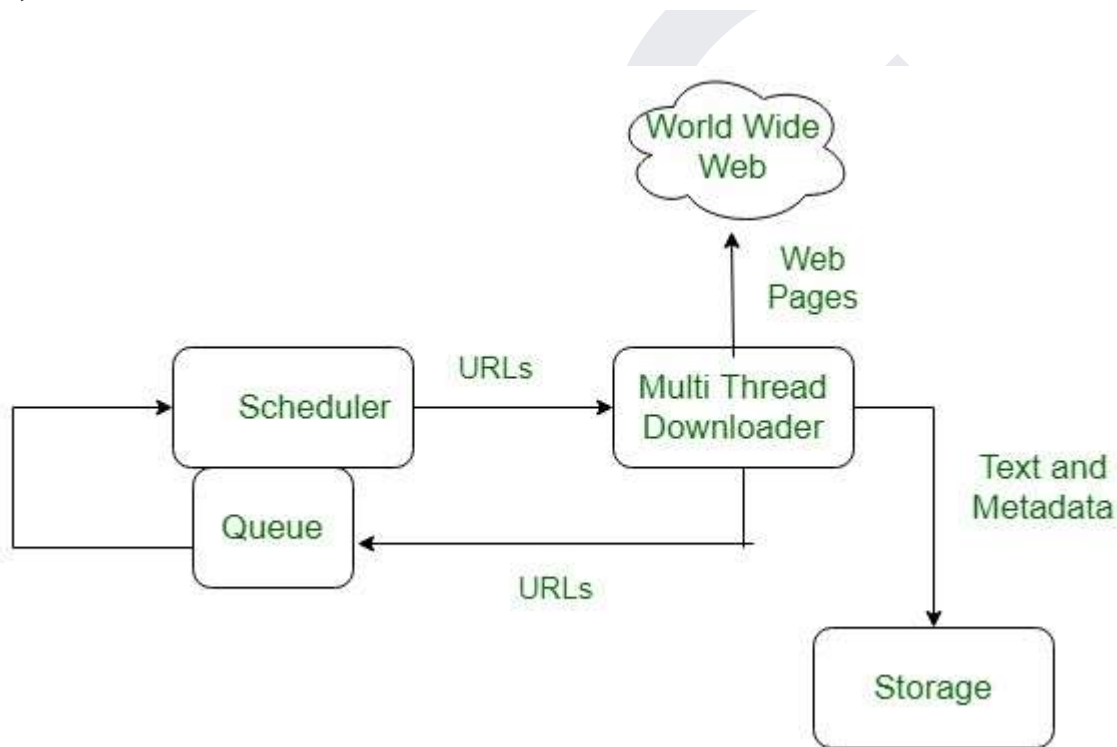
WWW का मतलब वर्ल्ड वाइड वेब है और इसे आम तौर पर वेब के नाम से जाना जाता है। WWW की शुरुआत CERN ने 1989 में की थी। WWW को दुनिया भर की अलग-अलग वेबसाइटों के संग्रह के रूप में परिभाषित किया जाता है, जिसमें स्थानीय सर्वर (या कंप्यूटर) के माध्यम से साझा की गई अलग-अलग जानकारी होती है।

वेब पेज हाइपरलिंक का उपयोग करके एक साथ जुड़े होते हैं जो HTML-स्वरूपित होते हैं और जिन्हें हाइपरटेक्स्ट भी कहा जाता है, ये इंटरनेट की मूलभूत इकाइयाँ हैं और इन्हें [हाइपरटेक्स्ट ट्रांसफर प्रोटोकॉल \(HTTP\)](#) के माध्यम से एक्सेस किया जाता है। ऐसे डिजिटल कनेक्शन या लिंक, उपयोगकर्ताओं को जानकारी के प्रासंगिक टुकड़ों को जोड़कर आसानी से वांछित जानकारी तक पहुँचने की अनुमति देते हैं। हाइपरटेक्स्ट का लाभ यह है कि यह आपको टेक्स्ट से कोई शब्द या वाक्यांश चुनने और अन्य साइटों पर क्लिक करने की अनुमति देता है, जिनके पास इसके बारे में अधिक जानकारी है। यह डेटा इंटरनेट पर टेक्स्ट, चित्र, ऑडियो या वीडियो प्रारूपों में प्रस्तुत किया जा सकता है।

# DSG Support Multi Solution

## WWW का इतिहास

यह 1989 में टिम बर्नर ली द्वारा बनाया गया एक प्रोजेक्ट है, जिसका उद्देश्य शोधकर्ताओं को CERN में प्रभावी ढंग से एक साथ काम करने में मदद करना है। यह एक संगठन है, जिसका नाम [वर्ल्ड वाइड वेब कंसोर्टियम \(W3C\)](#) है, जिसे वेब के आगे के विकास के लिए विकसित किया गया था। इस संगठन का निर्देशन टिम बर्नर ली द्वारा किया जाता है, जिन्हें वेब का जनक भी कहा जाता है। CERN, जहाँ टिम बर्नर्स ने काम किया, 100 से अधिक देशों के 1700 से अधिक शोधकर्ताओं का एक समुदाय है। ये शोधकर्ता CERN में थोड़ा समय बिताते हैं और बाकी समय वे अपने देश में अपने कॉलेजों और राष्ट्रीय शोध सुविधाओं में काम करते हैं, इसलिए ठोस संचार की आवश्यकता थी ताकि वे डेटा का आदान-प्रदान कर सकें।



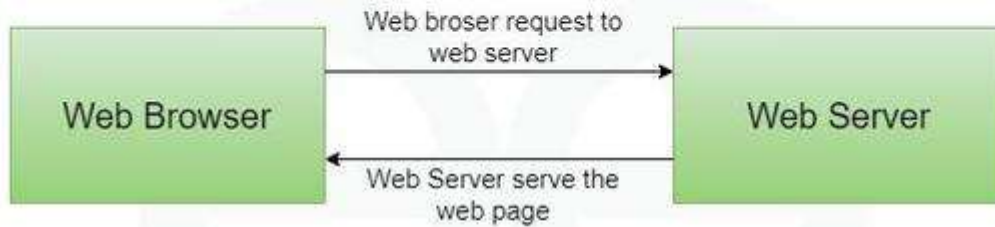
## WWW की कार्यप्रणाली

वेब ब्राउज़र का उपयोग वेब पेजों तक पहुँचने के लिए किया जाता है। वेब ब्राउज़र को ऐसे प्रोग्राम के रूप में परिभाषित किया जा सकता है जो इंटरनेट पर टेक्स्ट, डेटा, चित्र, एनीमेशन और वीडियो प्रदर्शित करते हैं। वर्ल्ड वाइड वेब पर हाइपरलिंक किए गए संसाधनों को वेब ब्राउज़र द्वारा प्रदान किए गए सॉफ्टवेयर इंटरफ़ेस का उपयोग करके एक्सेस किया जा सकता है। शुरुआत में, वेब ब्राउज़र का उपयोग केवल वेब सर्फ़िंग के लिए किया जाता था, लेकिन अब वे अधिक सार्वभौमिक हो गए हैं।

नीचे दिया गया आरेख दर्शाता है कि वेब किस तरह से काम करता है, ठीक वैसे ही जैसे इंटरनेट का [क्लाइंट-सर्वर आर्किटेक्चर](#) होता है। जब उपयोगकर्ता वेब पेज या अन्य जानकारी का अनुरोध करते हैं, तो आपके सिस्टम का वेब ब्राउज़र सर्वर से जानकारी के लिए अनुरोध करता है और फिर वेब सर्वर अनुरोधित सेवाएँ

# DSG Support Multi Solution

वेब ब्राउज़र को वापस प्रदान करता है और अंत में अनुरोधित सेवा का उपयोग उस उपयोगकर्ता द्वारा किया जाता है जिसने अनुरोध किया था।



World Wide Web



वर्ल्ड वाइड वेब

वेब ब्राउज़र का इस्तेमाल कई कामों के लिए किया जा सकता है, जिसमें सर्च करना, मेल भेजना, फाइल ट्रांसफर करना और बहुत कुछ शामिल है। आमतौर पर इस्तेमाल किए जाने वाले कुछ ब्राउज़र हैं इंटरनेट एक्सप्लोरर, ओपेरा मिनी और गूगल क्रोम।

## WWW की विशेषताएं

- WWW खुला स्रोत है।
- यह विभिन्न वेबसाइटों पर फैली एक वितरित प्रणाली है।
- यह एक हाइपरटेक्स्ट सूचना प्रणाली है।
- यह क्रॉस-प्लेटफॉर्म है।
- कई सेवाओं के लिए एकल इंटरफ़ेस प्रदान करने के लिए वेब ब्राउज़र का उपयोग करता है।
- गतिशील, इंटरैक्टिव और विकासशील।

## HTTP

**HTTP** का मतलब **हाइपरटेक्स्ट ट्रांसफर प्रोटोकॉल** है। यह इंटरनेट पर जानकारी साझा करने के लिए वेब ब्राउज़र और सर्वर द्वारा संचार का मुख्य तरीका है। टिम बर्नर ने इसका आविष्कार किया। हाइपरटेक्स्ट एक प्रकार का टेक्स्ट है जिसे विशेष रूप से **हाइपरटेक्स्ट मार्कअप लैंग्वेज (HTML)** नामक कुछ मानक कोडिंग

# DSG Support Multi Solution

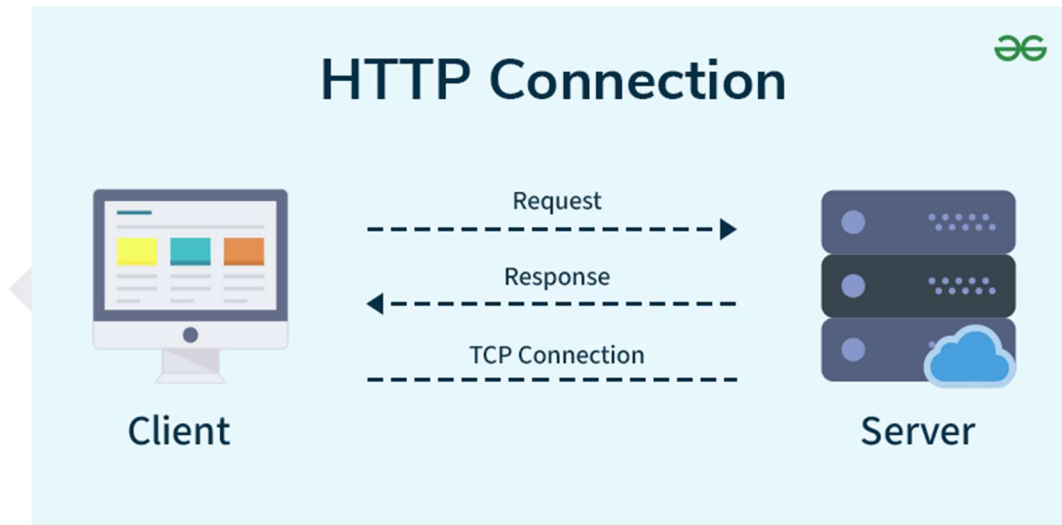
भाषा की मदद से कोड किया जाता है। **HTTP/2** HTTP का नया संस्करण है। HTTP/3 HTTP का नवीनतम संस्करण है, जिसे 2022 में प्रकाशित किया गया है।

जब आप किसी वेबसाइट पर जाते हैं, तो HTTP आपके ब्राउज़र को आपके द्वारा देखे जाने वाले वेब पेजों को प्रदर्शित करने के लिए आवश्यक डेटा का अनुरोध करने और प्राप्त करने में मदद करता है। यह इंटरनेट के काम करने का एक मूलभूत हिस्सा है, जिससे हमारे लिए वेबसाइटों को ब्राउज़ करना और उनसे बातचीत करना संभव हो जाता है।

## HTTP का कार्य

सबसे पहले, जब भी हम कोई वेबसाइट खोलना चाहते हैं तो हम सबसे पहले एक वेब ब्राउज़र खोलते हैं उसके बाद हम उस वेबसाइट का URL टाइप करते हैं (जैसे, [www.facebook.com](http://www.facebook.com))। यह URL अब **डोमेन नेम सर्वर (DNS)** को भेजा जाता है। फिर DNS सबसे पहले अपने डेटाबेस में इस URL के लिए रिकॉर्ड की जाँच करता है, और फिर DNS इस URL से संबंधित वेब ब्राउज़र को IP पता लौटाता है। अब ब्राउज़र वास्तविक सर्वर को अनुरोध भेजने में सक्षम है।

सर्वर द्वारा क्लाइंट को डेटा भेजने के बाद, कनेक्शन बंद हो जाएगा। अगर हम सर्वर से कुछ और चाहते हैं तो हमें क्लाइंट और सर्वर के बीच कनेक्शन को फिर से स्थापित करना होगा।



## HTTP की विशेषताएँ

HTTP एक IP आधारित संचार प्रोटोकॉल है जिसका उपयोग सर्वर से क्लाइंट तक या इसके विपरीत डेटा पहुंचाने के लिए किया जाता है।

- सर्वर क्लाइंट द्वारा उठाए गए अनुरोध को संसाधित करता है, और सर्वर और क्लाइंट एक दूसरे को केवल वर्तमान बोली और प्रतिक्रिया अवधि के दौरान ही जानते हैं।
- किसी भी प्रकार की सामग्री का आदान-प्रदान किया जा सकता है, बशर्ते सर्वर और क्लाइंट इसके अनुकूल हों।

# DSG Support Multi Solution

- एक बार डेटा का आदान-प्रदान हो जाने के बाद, सर्वर और क्लाइंट आपस में जुड़े नहीं रहते।
- यह क्लाइंट और सर्वर आवश्यकताओं पर आधारित अनुरोध और प्रतिक्रिया प्रोटोकॉल है।
- यह एक कनेक्शन-रहित प्रोटोकॉल है क्योंकि कनेक्शन बंद होने के बाद, सर्वर को क्लाइंट के बारे में कुछ भी याद नहीं रहता है और क्लाइंट को सर्वर के बारे में कुछ भी याद नहीं रहता है।
- यह एक स्टेटलेस प्रोटोकॉल है क्योंकि क्लाइंट और सर्वर दोनों एक दूसरे से कुछ भी अपेक्षा नहीं करते हैं लेकिन फिर भी वे संवाद करने में सक्षम होते हैं।

## HTTP के लाभ

- एक साथ कम कनेक्शन होने के कारण मेमोरी उपयोग और CPU उपयोग कम होता है।
- चूंकि टीसीपी कनेक्शन कम हैं, इसलिए नेटवर्क भीड़भाड़ कम है।
- चूंकि हैंडशेकिंग प्रारंभिक कनेक्शन चरण में की जाती है, इसलिए विलंबता कम हो जाती है, क्योंकि बाद के अनुरोधों के लिए हैंडशेकिंग की कोई आवश्यकता नहीं होती।
- कनेक्शन बंद किये बिना भी त्रुटि की सूचना दी जा सकती है।
- HTTP अनुरोधों या प्रतिक्रियाओं की HTTP पाइपलाइन-लाइनिंग की अनुमति देता है।

## HTTP के नुकसान

- HTTP को संचार स्थापित करने और डेटा स्थानांतरित करने के लिए उच्च शक्ति की आवश्यकता होती है।
- HTTP कम सुरक्षित है क्योंकि यह HTTPS जैसी किसी एन्क्रिप्शन विधि का उपयोग नहीं करता है और नियमित HTTP अनुरोधों और प्रतिक्रियाओं को एन्क्रिप्ट करने के लिए TLS का उपयोग करता है।
- HTTP सेलुलर फोन के लिए अनुकूलित नहीं है और यह बहुत अधिक अस्पष्ट है।
- HTTP वास्तविक डेटा विनिमय की सुविधा प्रदान नहीं करता है क्योंकि यह कम सुरक्षित है।
- क्लाइंट तब तक कनेक्शन बंद नहीं करता जब तक उसे सर्वर से पूरा डेटा प्राप्त नहीं हो जाता; इसलिए, सर्वर को डेटा पूरा होने तक प्रतीक्षा करनी पड़ती है और इस दौरान वह अन्य क्लाइंट के लिए उपलब्ध नहीं हो सकता।

## Web Server

वेब सर्वर एक ऐसा सिस्टम है जो वेब पेजों को स्टोर करता है और उपयोगकर्ताओं तक पहुंचाता है। यह ब्राउज़र से अनुरोधों को संभालने के लिए HTTP/HTTPS प्रोटोकॉल का उपयोग करता है और HTML फ़ाइलें, चित्र और वीडियो जैसी चीज़ें वापस भेजता है।

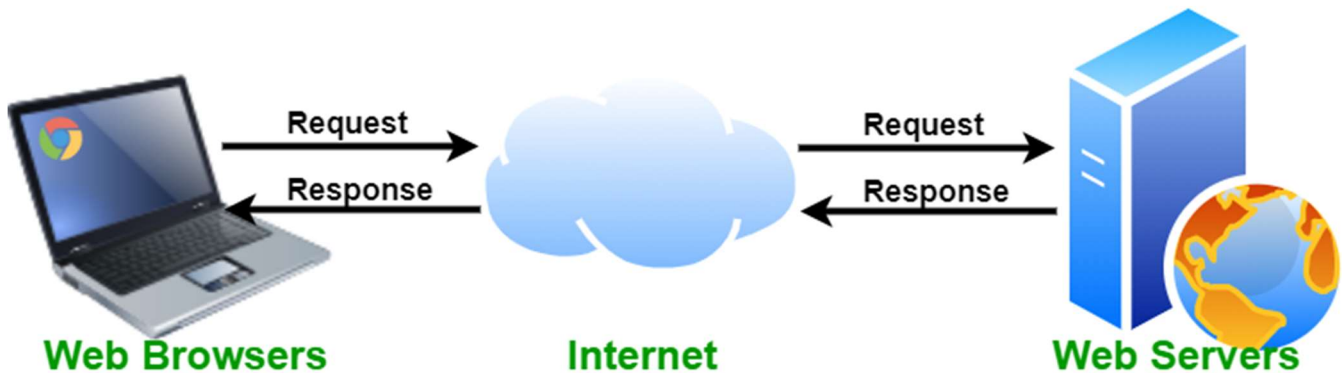
- **वेब ब्राउज़र** यह वह जगह है जहाँ आप URL टाइप करते हैं (उदाहरण के लिए, [www.example.com](http://www.example.com))। वेब ब्राउज़र उस पेज के लिए अनुरोध भेजता है जिसे आप एक्सेस करना चाहते हैं।

# DSG Support Multi Solution

- **इंटरनेट के माध्यम से भेजा गया अनुरोध:** यह अनुरोध इंटरनेट के माध्यम से वेब सर्वर तक जाता है तथा वेबपृष्ठ या छवि जैसी विशिष्ट सामग्री के लिए पूछता है।
- **वेब सर्वर :** वेब सर्वर एक ऐसा कंप्यूटर है जो वेब पेजों को संग्रहीत और वितरित करता है। यह ब्राउज़र से अनुरोध प्राप्त करता है और उसे संसाधित करता है। फिर सर्वर अनुरोधित जानकारी (जैसे HTML, चित्र, आदि) तैयार करता है।
- **प्रतिक्रिया वापस भेजी गई:** वेब सर्वर प्रतिक्रिया (अनुरोधित वेबपेज या सामग्री) को इंटरनेट के माध्यम से वापस भेजता है।
- **वेब ब्राउज़र पृष्ठ प्रदर्शित करता है :** ब्राउज़र वेब सर्वर से प्रतिक्रिया प्राप्त करता है और उपयोगकर्ता को वेब पेज प्रदर्शित करता है।

वेब सर्वर कैसे काम करता है?

- **क्लाइंट अनुरोध:** वेब ब्राउज़र (<https://www.example.com>) में उपयोगकर्ता एक URL दर्ज करता है।
- **DNS रिज़ॉल्यूशन:** अनुरोधित डोमेन का IP पता प्राप्त करने के लिए, ब्राउज़र डोमेन नाम सिस्टम (DNS) सर्वर से संपर्क करता है।
- **वेब सर्वर से जुड़ना:** प्राप्त आईपी पते का उपयोग करके ब्राउज़र वेब सर्वर से कनेक्शन स्थापित करता है।
- **अनुरोध प्रसंस्करण:** वेब सर्वर अनुरोध प्राप्त करता है और उसे संसाधित करता है।
- **प्रतिक्रिया प्रदान करना:** अनुरोधित फ़ाइलें (HTML, CSS, JavaScript, छवियाँ) वेब सर्वर द्वारा क्लाइंट के ब्राउज़र पर वापस भेजी जाती हैं।
- **वेब पेज प्रस्तुत करना:** प्राप्त डेटा के आधार पर ब्राउज़र उपयोगकर्ता को वेब पेज प्रदर्शित करता है।



DSG

वेब सर्वर कैसे काम करता है?

# DSG Support Multi Solution

## वेब सर्वर के प्रकार

वेब सर्वर को उनकी कार्यक्षमता, उपयोग और कार्यान्वयन के आधार पर वर्गीकृत किया जा सकता है। नीचे कुछ सबसे आम प्रकार दिए गए हैं

वेब सर्वर

### 1. अपाचे वेब सर्वर

[अपाचे वेब सर्वर](#) अपाचे लाइसेंस 2.0 के तहत उपलब्ध सबसे व्यापक रूप से इस्तेमाल किए जाने वाले और सबसे पुराने ओपन-सोर्स वेब सर्वरों में से एक है, जिसका मतलब है कि इसका स्वतंत्र रूप से उपयोग किया जा सकता है और इसे संशोधित किया जा सकता है। इसे अपाचे सॉफ्टवेयर फाउंडेशन द्वारा विकसित किया गया था, और पहली बार 1995 में जारी किया गया था। यह अत्यधिक लचीला और अनुकूलन योग्य है। यह [C](#) प्रोग्रामिंग भाषा में लिखा गया है।

- एकाधिक ऑपरेटिंग सिस्टम (विंडोज़, लिनक्स, मैकओएस) का समर्थन करता है।
- उन्नत रूटिंग की अनुमति देता है।
- निर्देशिका-स्तरीय कॉन्फिगरेशन प्रदान करता है।

### 2. एनजीइंक्स वेब सर्वर

[Nginx](#) (जिसे “इंजन-एक्स” के रूप में उच्चारित किया जाता है) एक उच्च-प्रदर्शन वेब सर्वर है जो अपनी गति, मापनीयता और समवर्ती कनेक्शनों के कुशल संचालन के लिए जाना जाता है। इसे इगोर सिसोव द्वारा विकसित किया गया था, और शुरू में 2004 में जारी किया गया था और अपने उच्च प्रदर्शन और मापनीयता के कारण जल्दी ही लोकप्रिय हो गया। यह C भाषा में लिखा गया है।

- बड़ी मात्रा में यातायात को संभालने के लिए डिज़ाइन किया गया।
- रिवर्स प्रॉक्सी और लोड बैलेंसर के रूप में कार्य करता है।
- स्थैतिक सामग्री परोसने के लिए कुशल।

### 3. माइक्रोसॉफ्ट आईआईएस (इंटरनेट सूचना सेवा)

[IIS](#) माइक्रोसॉफ्ट द्वारा विकसित एक वेब सर्वर है, जिसे विंडोज सर्वर वातावरण के साथ काम करने के लिए डिज़ाइन किया गया है। इसे माइक्रोसॉफ्ट द्वारा विकसित किया गया था, और पहली बार 1995 में विंडोज-आधारित सिस्टम के लिए विशेष रूप से डिज़ाइन किए गए वेब सर्वर के रूप में जारी किया गया था। यह [C++](#) में लिखा गया है।

- ASP.NET, PHP, और अन्य वेब प्रौद्योगिकियों का समर्थन करता है।
- अंतर्निहित सुरक्षा सुविधाएँ प्रदान करता है।



# DSG Support Multi Solution

- माइक्रोसॉफ्ट उत्पादों के साथ अच्छी तरह से एकीकृत करता है।

## 4. लाइटस्पीड वेब सर्वर

लाइटस्पीड एक उच्च-प्रदर्शन वेब सर्वर है जो अपनी गति और सुरक्षा सुविधाओं के लिए जाना जाता है। लाइटस्पीड वेब सर्वर, लाइटस्पीड टेक्नोलॉजीज द्वारा विकसित किया गया है, और पहली बार 2003 में अपाचे के उच्च-प्रदर्शन विकल्प के रूप में पेश किया गया था। यह C में लिखा गया है।

- कुछ परिदृश्यों में अपाचे की तुलना में तेज़ प्रसंस्करण।
- अंतर्निहित DDoS सुरक्षा.
- उच्च दक्षता के साथ PHP अनुप्रयोगों का समर्थन करता है।

## 5. अपाचे टॉमकैट वेब सर्वर

अपाचे टॉमकैट एक वेब सर्वर है जिसका उपयोग मुख्य रूप से जावा-आधारित अनुप्रयोगों को चलाने के लिए किया जाता है। अपाचे टॉमकैट वेब सर्वर को अपाचे सॉफ्टवेयर फाउंडेशन द्वारा विकसित किया गया था और इसे शुरू में 1998 में रिलीज़ किया गया था। यह जावा में लिखा गया है, जो इसे स्प्रिंग बूट जैसे जावा-आधारित फ्रेमवर्क के साथ अत्यधिक संगत बनाता है ।

- जावा सर्वलेट्स और JSP (जावा सर्वर पेजेस) का समर्थन करता है।
- जावा ईई अनुप्रयोगों के लिए एक मजबूत वातावरण प्रदान करता है।
- अपाचे वेब सर्वर के साथ अच्छी तरह से काम करता है।

## 6. नोडजेएस वेब सर्वर

NodeJS खुद ही HTTP अनुरोधों को सीधे संभालकर एक वेब सर्वर के रूप में कार्य कर सकता है, बिना किसी अतिरिक्त सॉफ्टवेयर की आवश्यकता के। NodeJS, रयान डाहल द्वारा विकसित किया गया है, और 2009 में एक पारंपरिक वेब सर्वर के बजाय एक जावास्क्रिप्ट रनटाइम वातावरण के रूप में पेश किया गया था । यह जावास्क्रिप्ट और C++ में लिखा गया है।

- घटना-संचालित, गैर-अवरुद्ध वास्तुकला।
- वास्तविक समय अनुप्रयोगों के लिए अत्यधिक कुशल।
- क्लाइंट-साइड और सर्वर-साइड विकास दोनों के लिए जावास्क्रिप्ट का उपयोग करता है।

## 7. लाइटटीपीडी

लाइटटीपीडी को जन केनेशके द्वारा विकसित किया गया था, और 2003 में एक हल्के और तेज़ वेब सर्वर के रूप में जारी किया गया था। यह C में लिखा गया है और BSD लाइसेंस के तहत एक ओपन-सोर्स प्रोजेक्ट के रूप में उपलब्ध है। लाइटटीपीडी एक क्रॉस-प्लेटफॉर्म वेब सर्वर है, जो विंडोज, लिनक्स और मैकओएस पर चलता है।

# DSG Support Multi Solution

- यह कम मेमोरी उपयोग और उच्च गति प्रदर्शन के लिए अनुकूलित है, जिससे यह सीमित संसाधनों वाले सर्वरों के लिए आदर्श है।
- एसिंक्रोनस अनुरोध प्रबंधन का उपयोग करता है, जो एकाधिक कनेक्शनों को संभालने के लिए दक्षता और मापनीयता में सुधार करता है।
- HTTPS, FastCGI और URL पुनर्लेखन का समर्थन करता है, जो इसे वेब होस्टिंग के लिए एक सुरक्षित और कुशल विकल्प बनाता है।

## 8. ओपनलाइटस्पीड

ओपनलाइटस्पीड को लाइटस्पीड टेक्नोलॉजीज द्वारा विकसित किया गया था, और 2013 में लाइटस्पीड वेब सर्वर के ओपन-सोर्स संस्करण के रूप में लॉन्च किया गया था। यह C में लिखा गया है और GPLv3 लाइसेंस के तहत उपलब्ध है। यह क्रॉस-प्लेटफॉर्म है, जो विंडोज, लिनक्स और मैकओएस को सपोर्ट करता है। ओपनलाइटस्पीड बिल्ट-इन कैशिंग, HTTP/3 सपोर्ट और हाई-परफॉरमेंस ऑप्टिमाइज़ेशन प्रदान करता है, जो इसे लाइटस्पीड के मुफ्त विकल्प की तलाश करने वाले डेवलपर्स के लिए एक बेहतरीन विकल्प बनाता है।

- अंतर्निहित कैशिंग और इवेंट-संचालित आर्किटेक्चर के साथ तेज़ प्रसंस्करण गति प्रदान करता है, जिससे वेबसाइट लोडिंग समय में सुधार होता है।
- क्लाइंट और सर्वर के बीच तेज़ और अधिक सुरक्षित संचार के लिए आधुनिक वेब प्रोटोकॉल का समर्थन करता है।
- सर्वर सेटिंग्स को कॉन्फ़िगर करने के लिए उपयोग में आसान इंटरफ़ेस प्रदान करता है, जिससे प्रबंधन सरल हो जाता है।

## 9. जिगसाँ सर्वर

जिगसाँ सर्वर को वर्ल्ड वाइड वेब कंसोर्टियम (W3C) द्वारा विकसित किया गया था, और इसे पहली बार 1996 में एक शोध-आधारित वेब सर्वर के रूप में जारी किया गया था। जिगसाँ जावा में लिखा गया है और ओपन-सोर्स है। इसे मुख्यधारा की वेब होस्टिंग के लिए इस्तेमाल किए जाने के बजाय नए वेब मानकों के परीक्षण और विकास के लिए डिज़ाइन किया गया है।

- जिगसाँ क्रॉस-प्लेटफॉर्म है, जो विंडोज़, लिनक्स और मैकओएस पर चलता है।
- यह उपयोगकर्ताओं को इसकी कार्यक्षमता को आसानी से विस्तारित और संशोधित करने की अनुमति देता है, जिससे यह अनुसंधान और विकास के लिए लचीला बन जाता है।
- HTTP/1.1 को पूर्णतः समर्थन करता है तथा इसे नई वेब प्रौद्योगिकियों के साथ प्रयोग करने के लिए डिज़ाइन किया गया है।

# DSG Support Multi Solution

## 10. सन जावा सिस्टम वेब सर्वर

सन जावा सिस्टम वेब सर्वर को सन माइक्रोसिस्टम्स द्वारा विकसित किया गया था, और 1996 में जावा अनुप्रयोगों के लिए अनुकूलित वेब सर्वर के रूप में जारी किया गया था। इसे C और C++ में लिखा गया था। Oracle द्वारा सन माइक्रोसिस्टम्स का अधिग्रहण करने के बाद, इस वेब सर्वर का विकास बंद कर दिया गया।

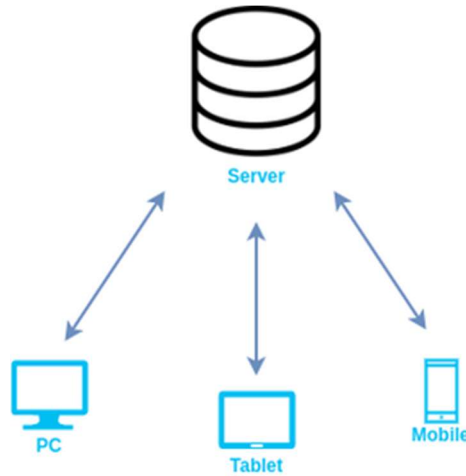
- विशेष रूप से जावा-आधारित उद्यम अनुप्रयोगों की मेजबानी के लिए डिज़ाइन किया गया, जो निर्बाध एकीकरण सुनिश्चित करता है।
- उच्च यातायात को कुशलतापूर्वक संभालता है, जिससे यह बड़े पैमाने के अनुप्रयोगों के लिए उपयुक्त हो जाता है।
- लचीले परिणियोजन के लिए विंडोज, लिनक्स और सोलारिस सहित एकाधिक ऑपरेटिंग सिस्टम का समर्थन करता है।

## Web Browser

- वेब ब्राउज़र [www](http://www) ([वर्ल्ड वाइड वेब](http://www)) का पता लगाने के लिए एक एप्लीकेशन सॉफ्टवेयर है। यह सर्वर और क्लाइंट के बीच एक इंटरफ़ेस प्रदान करता है और यह वेब दस्तावेज़ों और सेवाओं के लिए सर्वर से अनुरोध करता है। यह HTML को रेंडर करने के लिए एक कंपाइलर के रूप में काम करता है जिसका उपयोग वेबपेज को डिज़ाइन करने के लिए किया जाता है। जब भी हम इंटरनेट पर कुछ भी खोजते हैं, तो ब्राउज़र HTML में लिखा हुआ एक वेब पेज लोड करता है, जिसमें टेक्स्ट, लिंक, इमेज और स्टाइल शीट और जावास्क्रिप्ट फ़ंक्शन जैसे अन्य आइटम शामिल होते हैं। Google Chrome, Microsoft Edge, Mozilla Firefox और Safari वेब ब्राउज़र के उदाहरण हैं।
- वेब ब्राउज़र का इतिहास
- पहला वेब ब्राउज़र वर्ल्ड वाइड वेब का आविष्कार 1990 में टिम बर्नर्स-ली ने किया था। बाद में, यह नेक्सस बन गया। 1993 में, मार्क एंड्रीसेन और उनकी टीम ने एक नए ब्राउज़र मोज़ेक का आविष्कार किया। यह डिवाइस स्क्रीन पर एक बार में टेक्स्ट और इमेज प्रदर्शित करने वाला पहला ब्राउज़र था। उन्होंने 1994 में एक और ब्राउज़र नेटस्केप का भी आविष्कार किया। अगले साल माइक्रोसॉफ्ट ने एक वेब ब्राउज़र इंटरनेट एक्सप्लोरर लॉन्च किया जो पहले से ही विंडोज ऑपरेटिंग सिस्टम में इंस्टॉल था। इसके बाद मोज़िला फ़ायरफ़ॉक्स, गूगल क्रोम, सफ़ारी, ओपेरा आदि जैसे विभिन्न विशेषताओं वाले कई ब्राउज़रों का आविष्कार किया गया।

# DSG Support Multi Solution

- वेब ब्राउज़र कैसे काम करता है?
- वेब ब्राउज़र हमें इंटरनेट पर कहीं भी जानकारी खोजने में मदद करता है। यह क्लाइंट कंप्यूटर पर इंस्टॉल होता है और वेब सर्वर से जानकारी का अनुरोध करता है। इस तरह के वर्किंग मॉडल को क्लाइंट-सर्वर मॉडल कहा जाता है।



• क्लाइंट-सर्वर मॉडल

- ब्राउज़र HTTP प्रोटोकॉल के माध्यम से जानकारी प्राप्त करता है। जिसमें डेटा का संचरण परिभाषित किया जाता है। जब ब्राउज़र सर्वर से डेटा प्राप्त करता है, तो इसे HTML में उपयोगकर्ता-पठनीय रूप में प्रस्तुत किया जाता है और, डिवाइस स्क्रीन पर जानकारी प्रदर्शित होती है।
- वेबसाइट कुकीज़
- जब हम इंटरनेट पर किसी वेबसाइट पर जाते हैं तो हमारा वेब ब्राउज़र हमारी जानकारी को छोटी फ़ाइलों में संग्रहीत करता है जिन्हें कुकीज़ कहा जाता है। कुकीज़ हमारे ब्राउज़िंग इतिहास के बारे में स्टेटफुल जानकारी को याद रखने के लिए डिज़ाइन की गई हैं। कुछ और कुकीज़ हमारे बारे में याद रखने के लिए उपयोग की जाती हैं जैसे हमारी रुचियाँ, हमारे ब्राउज़िंग पैटर्न, आदि। वेबसाइटें कुकीज़ का उपयोग करके हमारी रुचियों के आधार पर हमें विज्ञापन दिखाती हैं।
- कुछ लोकप्रिय वेब ब्राउज़र
- यहां 7 लोकप्रिय वेब ब्राउज़रों की सूची दी गई है:
- 1. **गूगल क्रोम:**
- गूगल द्वारा विकसित क्रोम विश्व में सबसे अधिक उपयोग किये जाने वाले वेब ब्राउज़रों में से एक है, जो अपनी गति और सरलता के लिए जाना जाता है।

# DSG Support Multi Solution

## • 2. मोज़िला फ़ायरफ़ॉक्स:

• मोज़िला फाउंडेशन द्वारा विकसित फ़ायरफ़ॉक्स एक ओपन-सोर्स ब्राउज़र है जो अपनी गोपनीयता सुविधाओं और अनुकूलन विकल्पों के लिए जाना जाता है।

## • 3. एप्पल सफारी:

• एप्पल द्वारा विकसित सफारी मैक और आईओएस डिवाइसों पर डिफ़ॉल्ट ब्राउज़र है और यह अपनी गति और अन्य एप्पल उत्पादों के साथ एकीकरण के लिए जाना जाता है।

## • 4. माइक्रोसॉफ्ट एज:

• माइक्रोसॉफ्ट द्वारा विकसित, एज विंडोज 10 पर डिफ़ॉल्ट ब्राउज़र है और अन्य माइक्रोसॉफ्ट उत्पादों और सेवाओं के साथ एकीकरण के लिए जाना जाता है।

## • 5. टोर ब्राउज़र:

• टोर प्रोजेक्ट द्वारा विकसित, टोर ब्राउज़र एक वेब ब्राउज़र है जो गुमनाम वेब ब्राउज़िंग के लिए डिज़ाइन किया गया है और मोज़िला फ़ायरफ़ॉक्स पर आधारित है।

## • 6. ओपेरा:

• ओपेरा सॉफ्टवेयर द्वारा विकसित, ओपेरा एक वेब ब्राउज़र है जो अपनी गति और अंतर्निहित वीपीएन सुविधा के लिए जाना जाता है।

## • 7. Brave:

• ब्रेव सॉफ्टवेयर द्वारा विकसित, ब्रेव एक वेब ब्राउज़र है जो गोपनीयता और सुरक्षा पर केंद्रित है और डिफ़ॉल्ट रूप से तृतीय-पक्ष विज्ञापनों और ट्रैकर्स को ब्लॉक करता है।

• ये कुछ सबसे लोकप्रिय वेब ब्राउज़र हैं, इसके अलावा विवाल्डी, वाटरफ़ॉक्स आदि जैसे अन्य ब्राउज़र भी उपलब्ध हैं। वेब ब्राउज़र का चुनाव उपयोगकर्ता की पसंद और आवश्यकताओं पर निर्भर करता है।

## Comman Gateway Interface

कॉमन गेटवे इंटरफ़ेस (CGI) एक मानक है जो वेब सर्वर और बाहरी डेटाबेस या सूचना स्रोतों के बीच संचार की सुविधा प्रदान करता है। यह मिडलवेयर के रूप में कार्य करता है, जिससे वेब सर्वर उन अनुप्रयोगों के साथ बातचीत कर सकते हैं जो डेटा को संसाधित करते हैं और प्रतिक्रियाएँ वापस भेजते हैं। CGI मानक को वर्ल्ड वाइड वेब कंसोर्टियम (W3C) द्वारा परिभाषित किया गया था और यह निर्दिष्ट करता है कि कोई प्रोग्राम हाइपर टेक्स्ट ट्रांसफ़र प्रोटोकॉल ( <https://www.geeksforgeeks.org/understanding-http-using-browsers/> ) सर्वर के साथ कैसे इंटरैक्ट करता है।

# DSG Support Multi Solution

CGI लाइब्रेरी एक पायथन लाइब्रेरी थी जो CGI स्क्रिप्ट (वेब सर्वर पर चलने वाली स्क्रिप्ट) लिखने के लिए एक सरल इंटरफ़ेस प्रदान करती थी।

इस लाइब्रेरी को अधिक आधुनिक और सुविधा संपन्न विकल्पों, जैसे कि वेब फ्रेमवर्क Django के पक्ष में हटा दिया गया है।

## सीजीआई की विशेषताएं:

- यह एक बहुत अच्छी तरह से परिभाषित और समर्थित मानक है।
- CGI स्क्रिप्ट आम तौर पर पर्ल, सी या शेल स्क्रिप्ट जैसी भाषाओं में लिखी जाती हैं। उदाहरण के लिए, एक साधारण पर्ल स्क्रिप्ट का उपयोग वेब पेज से सबमिट किए गए फॉर्म डेटा को प्रोसेस करने और एक गतिशील प्रतिक्रिया उत्पन्न करने के लिए किया जा सकता है।
- CGI अनुप्रयोगों को HTML के साथ इंटरफ़ेस करने की अनुमति देता है, जिससे वेब पेजों के लिए गतिशील सामग्री निर्माण संभव हो जाता है। उदाहरण के लिए, CGI स्क्रिप्ट का उपयोग डेटाबेस से डेटा प्राप्त करने और वेब पेज पर प्रदर्शित करने के लिए गतिशील HTML सामग्री उत्पन्न करने के लिए किया जा सकता है।
- काउंटर बनाने के लिए CGI सबसे अच्छी विधि है क्योंकि यह वर्तमान में सबसे तेज़ है
- CGI मानक आम तौर पर आज के ब्राउज़रों के साथ सबसे अधिक संगत है

## सीजीआई के लाभ:

- त्वरित कार्यान्वयन: CGI सरल वेब-आधारित कार्यों, जैसे कि फॉर्म प्रोसेसिंग या डेटा हेरफेर को लागू करने के लिए एक त्वरित और सरल विधि हो सकती है। उदाहरण के लिए, एक CGI स्क्रिप्ट को वेब फॉर्म से उपयोगकर्ता इनपुट को संसाधित करने और प्रतिक्रिया उत्पन्न करने के लिए जल्दी से विकसित और तैनात किया जा सकता है।
- मौजूदा कोड का उपयोग करना आसान है: CGI स्क्रिप्ट का एक बड़ा मौजूदा कोड बेस मौजूद है, खास तौर पर Perl में, जिसे आसानी से इस्तेमाल किया जा सकता है या विशिष्ट आवश्यकताओं के लिए अनुकूलित किया जा सकता है। उदाहरण के लिए, फॉर्म प्रोसेसिंग, डेटा वैलिडेशन और फ़ाइल अपलोड जैसे कार्यों के लिए कई CGI स्क्रिप्ट उपलब्ध हैं।
- सुपरिभाषित और समर्थित मानक: CGI वेब विकास में एक व्यापक रूप से स्वीकृत मानक है, जो विभिन्न प्रणालियों और प्लेटफॉर्मों में संगतता सुनिश्चित करता है।
- सरल कार्य करने के लिए CGI-आधारित काउंटर और CGI कोड प्रचुर मात्रा में उपलब्ध हैं।

## सीजीआई के नुकसान:

- पेज लोड में ओवरहेड: CGI में प्रत्येक पेज लोड में स्क्रिप्ट को मेमोरी में लोड करने के कारण ओवरहेड होता है, जो प्रदर्शन को प्रभावित कर सकता है, खासकर उच्च-ट्रैफिक वाली वेबसाइटों में। उदाहरण के

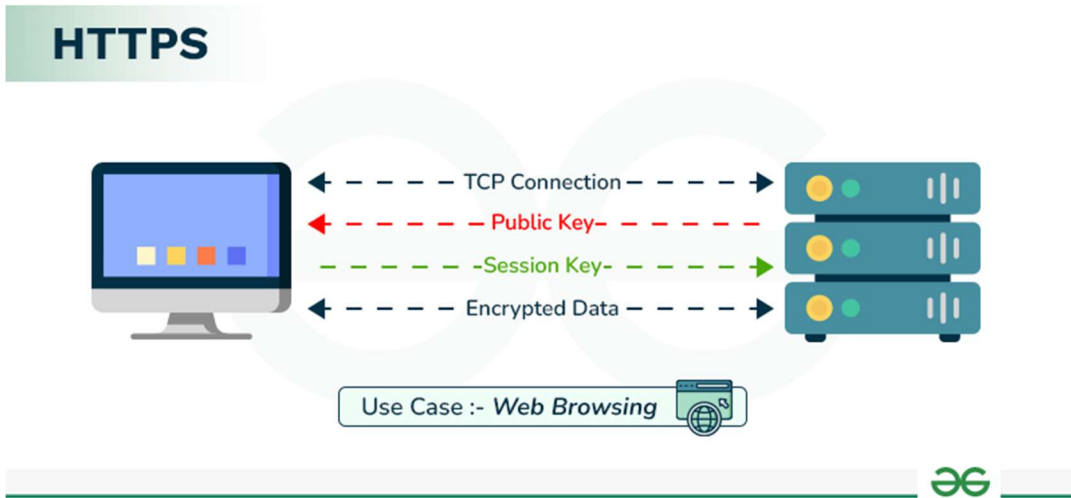
# DSG Support Multi Solution

लिए, यदि किसी वेबसाइट को बड़ी संख्या में अनुरोध प्राप्त होते हैं, तो प्रत्येक अनुरोध के लिए CGI स्क्रिप्ट लोड करने का ओवरहेड धीमी प्रतिक्रिया समय का कारण बन सकता है।

- सीमित कैशिंग क्षमताएँ: CGI स्क्रिप्ट पेज लोड के बीच मेमोरी में डेटा को आसानी से कैश करने की अनुमति नहीं देती हैं, जो प्रदर्शन और मापनीयता को प्रभावित कर सकती हैं। उदाहरण के लिए, यदि किसी वेबसाइट को बार-बार डेटाबेस क्वेरी या अन्य संसाधन-गहन कार्यों की आवश्यकता होती है, तो सीमित कैशिंग क्षमताओं के कारण CGI सबसे अच्छा विकल्प नहीं हो सकता है।
- सुरक्षा: चूँकि CGI स्क्रिप्ट सर्वर पर निष्पादित की जाती हैं, इसलिए सुरक्षा कमज़ोरियों का जोखिम हमेशा बना रहता है। उदाहरण के लिए, यदि स्क्रिप्ट उपयोगकर्ता इनपुट को ठीक से मान्य नहीं करती है, तो यह SQL इंजेक्शन हमलों के प्रति कमज़ोर हो सकती है।
- स्केलेबिलिटी: CGI बड़ी मात्रा में ट्रैफ़िक या अनुरोधों को संभालने के लिए उपयुक्त नहीं है। जैसे-जैसे अनुरोधों की संख्या बढ़ती है, सर्वर का प्रदर्शन प्रभावित हो सकता है, जिससे प्रतिक्रिया समय धीमा हो सकता है या क्रैश भी हो सकता है।

## HTTPS

हाइपरटेक्स्ट ट्रांसफर प्रोटोकॉल सिक्योर एक प्रोटोकॉल है जिसका उपयोग उपयोगकर्ता ब्राउज़र और वेबसाइट के बीच संचार करने के लिए किया जाता है। यह डेटा के हस्तांतरण में भी मदद करता है। यह HTTP का सुरक्षित संस्करण है। डेटा ट्रांसफर को अधिक सुरक्षित बनाने के लिए इसे एन्क्रिप्ट किया जाता है। पासवर्ड, संपर्क जानकारी आदि जैसी संवेदनशील जानकारी संचारित करते समय सुरक्षा सुनिश्चित करने के लिए एन्क्रिप्शन की आवश्यकता होती है।





# DSG Support Multi Solution

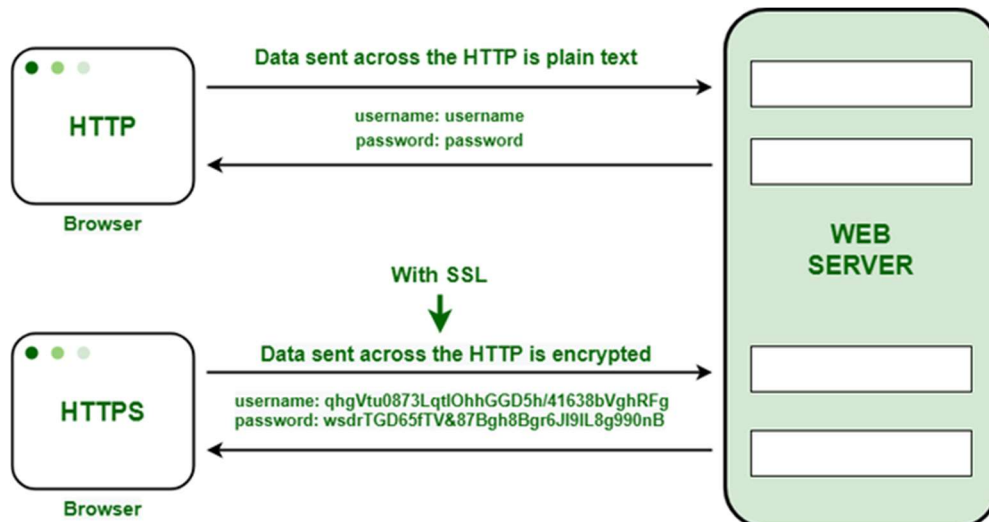
एचटीटीपी	HTTPS के
HTTP का तात्पर्य हाइपरटेक्स्ट ट्रांसफर प्रोटोकॉल है।	HTTPS का तात्पर्य हाइपरटेक्स्ट ट्रांसफर प्रोटोकॉल सिक्योर है।
यूआरएल "http://" से शुरू होता है।	यूआरएल "https://" से शुरू होता है।
HTTP <a href="#">एप्लीकेशन लेयर</a> पर काम करता है .	HTTPS <a href="#">ट्रांसपोर्ट लेयर</a> पर काम करता है ।
HTTP की गति HTTPS से अधिक तेज है।	HTTPS की गति HTTP से धीमी है।

[इन दोनों के बीच अधिक अंतर के लिए, http:// और https:// के बीच अंतर](#) लेख देखें ।

## HTTPS कैसे काम करता है?

HTTPS ब्राउज़र और वेब सर्वर के बीच संचार स्थापित करता है। यह संचार स्थापित करने के लिए **सिक्योर सॉकेट लेयर (SSL)** और **ट्रांसपोर्ट लेयर सिक्योरिटी (TLS)** प्रोटोकॉल का उपयोग करता है। SSL का नया संस्करण **TLS (ट्रांसपोर्ट लेयर सिक्योरिटी)** है ।

HTTPS पारंपरिक HTTP प्रोटोकॉल का उपयोग करता है और इसके ऊपर SSL/TLS की एक परत जोड़ता है। [HTTP और HTTPS का वर्कफ़्लो](#) एक जैसा रहता है, ब्राउज़र और सर्वर अभी भी HTTP प्रोटोकॉल का उपयोग करके एक दूसरे से संवाद करते हैं। हालाँकि, यह एक सुरक्षित SSL कनेक्शन पर किया जाता है। SSL कनेक्शन डेटा सुरक्षा सुनिश्चित करने के लिए एक्सचेंज किए जा रहे डेटा के एन्क्रिप्शन और डिक्लिप्शन के लिए जिम्मेदार है।



# DSG Support Multi Solution

## सुरक्षित सॉकेट लेयर (SSL)

एसएसएल की मुख्य जिम्मेदारी यह सुनिश्चित करना है कि संचार प्रणालियों के बीच डेटा स्थानांतरण सुरक्षित और विश्वसनीय हो। यह मानक सुरक्षा तकनीक है जिसका उपयोग अनुरोधों के प्रसारण के दौरान डेटा के एन्क्रिप्शन और डिक्लिप्शन के लिए किया जाता है।

जैसा कि पहले चर्चा की गई है, HTTP मूल रूप से वही पुराना HTTP है लेकिन SSL के साथ। संचार उपकरणों के बीच एक सुरक्षित संचार लिंक स्थापित करने के लिए, SSL एक डिजिटल प्रमाणपत्र का उपयोग करता है जिसे **SSL प्रमाणपत्र** कहा जाता है।

SSL परत की दो प्रमुख भूमिकाएँ हैं

- यह सुनिश्चित करना कि ब्राउज़र आवश्यक सर्वर से सीधे संचार करता है।
- यह सुनिश्चित करना कि केवल संचार प्रणालियों को ही आदान-प्रदान किये जाने वाले संदेशों तक पहुंच हो।

## HTTPS में एन्क्रिप्शन

HTTP ब्राउज़र और वेब सर्वर के बीच हाइपरटेक्स्ट फॉर्मेट में डेटा ट्रांसफ़र करता है, जबकि HTTPS एन्क्रिप्टेड फॉर्मेट में डेटा ट्रांसफ़र करता है। नतीजतन, HTTPS वेबसाइटों को उनकी जानकारी को इस तरह से प्रसारित होने से बचाता है कि नेटवर्क पर कोई भी व्यक्ति आसानी से देख सके। ब्राउज़र और वेब सर्वर के बीच ट्रांज़िट के दौरान, HTTPS डेटा को हैकर्स द्वारा एक्सेस और बदले जाने से बचाता है। भले ही ट्रांसमिशन को रोक दिया गया हो, हैकर्स इसका उपयोग नहीं कर पाएंगे क्योंकि संदेश एन्क्रिप्टेड है।

यह संचार लिंक को सुरक्षित करने के लिए असममित सार्वजनिक कुंजी अवसंरचना का उपयोग करता है। एन्क्रिप्शन के लिए दो अलग-अलग प्रकार की कुंजियों का उपयोग किया जाता है -

- **निजी कुंजी** : इसका उपयोग सार्वजनिक कुंजी द्वारा एन्क्रिप्ट किए गए डेटा के डिक्लिप्शन के लिए किया जाता है। यह सर्वर-साइड पर रहता है और वेबसाइट के मालिक द्वारा नियंत्रित किया जाता है। यह प्रकृति में निजी है।
- **सार्वजनिक कुंजी** : यह प्रकृति में सार्वजनिक है और सर्वर से संचार करने वाले सभी उपयोगकर्ताओं के लिए सुलभ है। निजी कुंजी का उपयोग सार्वजनिक कुंजी द्वारा एन्क्रिप्ट किए गए डेटा के डिक्लिप्शन के लिए किया जाता है।

## HTTPS का लाभ

- **सुरक्षित संचार**: HTTPS संचरण के दौरान एन्क्रिप्शन प्रदान करके संचार प्रणाली के बीच एक सुरक्षित संचार लिंक स्थापित करता है।

# DSG Support Multi Solution

- **डेटा अखंडता:** डेटा को एन्क्रिप्ट करके, HTTPS डेटा अखंडता सुनिश्चित करता है। इसका मतलब यह है कि अगर किसी भी बिंदु पर डेटा से समझौता किया जाता है, तो हैकर्स एक्सचेंज किए जा रहे डेटा को पढ़ने या संशोधित करने में सक्षम नहीं होंगे।
- **गोपनीयता और सुरक्षा:** HTTPS हमलावरों को निष्क्रिय रूप से आदान-प्रदान किए जा रहे डेटा तक पहुंचने से रोकता है, जिससे उपयोगकर्ताओं की गोपनीयता और सुरक्षा की रक्षा होती है।
- **तेज़ प्रदर्शन:** TTPS डेटा को एन्क्रिप्ट करता है और उसका आकार कम करता है। HTTPS के मामले में छोटा आकार तेज़ डेटा ट्रांसमिशन के लिए ज़िम्मेदार है।

## Principle Of Web Design

डिज़ाइन के सिद्धांत बुनियादी नियम हैं जो डिज़ाइन को अच्छा दिखने और अच्छी तरह से काम करने में मदद करते हैं। इन नियमों में संतुलन, विपरीतता, जोर, गति, पैटर्न, लय और एकता शामिल हैं। इन सिद्धांतों का उपयोग करके, डिजाइनर अधिक आकर्षक और प्रभावी डिज़ाइन बना सकते हैं। इस लेख में, हम इनमें से प्रत्येक सिद्धांत की व्याख्या करेंगे और बताएंगे कि वे किसी डिज़ाइन को सफल बनाने में कैसे मदद करते हैं। चाहे आप अभी शुरुआत कर रहे हों या अपने कौशल में सुधार करना चाहते हों, ये सिद्धांत बेहतरीन डिज़ाइन बनाने के लिए ज़रूरी हैं।

### डिजाइन के सिद्धांत क्या हैं?

डिज़ाइन के सिद्धांत बुनियादी नियम हैं जो डिज़ाइन को अच्छा दिखने और अच्छी तरह से काम करने में मदद करते हैं। वे मार्गदर्शन करते हैं कि डिज़ाइन में तत्वों को कैसे व्यवस्थित किया जाए ताकि इसे देखने में आकर्षक और प्रभावी बनाया जा सके। डिज़ाइन के मुख्य सिद्धांत इस प्रकार हैं:

1. **संतुलन** : स्थिरता की भावना पैदा करने के लिए तत्वों को समान रूप से वितरित करना।
2. **कंट्रास्ट (विपरीतता)** : तत्वों को उजागर करने के लिए रंग, आकार या आकृति में अंतर का उपयोग करना।
3. **जोर** : किसी डिज़ाइन के सबसे महत्वपूर्ण भागों पर प्रकाश डालना।
4. **गति (मूवमेंट)** : दर्शकों की आंखों को डिजाइन के माध्यम से एक विशिष्ट तरीके से निर्देशित करना।
5. **पैटर्न** : क्रम और स्थिरता की भावना पैदा करने के लिए तत्वों को दोहराना।
6. **लय** : नियमित अंतराल पर तत्वों को दोहराकर गति की भावना पैदा करना।
7. **एकता** : यह सुनिश्चित करना कि डिज़ाइन के सभी भाग एक साथ मिलकर काम करें।

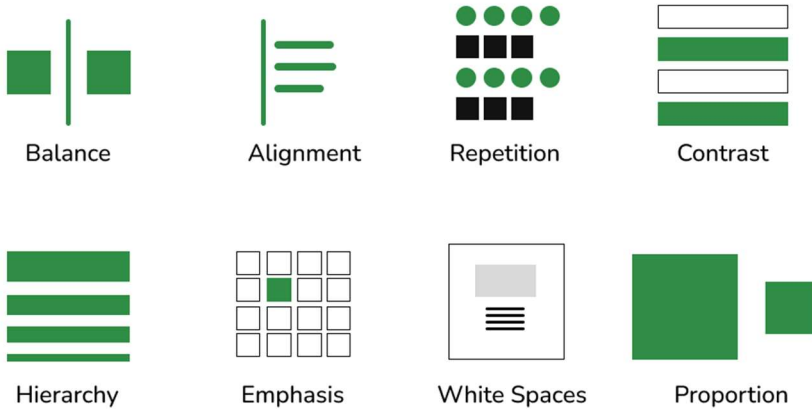
इन सिद्धांतों का उपयोग करके, डिजाइनर अधिक प्रभावी और आकर्षक डिजाइन बना सकते हैं।

# DSG Support Multi Solution

## डिजाइन के सिद्धांत

हमने पहले चर्चा की थी कि डिजाइन के विभिन्न सिद्धांत हैं और यह डिजाइनर से डिजाइनर तक भिन्न होता है, लेकिन यहां डिजाइन सिद्धांतों की एक सूची दी गई है जो सबसे महत्वपूर्ण और सबसे अधिक उपयोग किए जाते हैं।

## Principles of Design



डिजाइन के सिद्धांत

## संतुलन

संतुलन का मतलब बस दृश्य सामंजस्य है। अच्छा संतुलन आपकी वेबसाइट पर दृश्य सामंजस्य बनाने की कोशिश करने जैसा है। उदाहरण के लिए, वेबसाइट के एक तरफ़ सारी जानकारी होने से बचना जब दूसरी तरफ़ लगभग कुछ भी न हो। संतुलन को तीन रूपों में मापा जा सकता है: सममित रूप से, विषम रूप से और रेडियल रूप से।

## संरेखण

संरेखण वह तरीका है जिसमें वेबसाइट पर तत्वों को रखा जाता है। संरेखण का सबसे आम उपयोग टेक्स्ट संरेखण के साथ होता है, जिसमें लिखे गए टेक्स्ट को बाएं, केंद्र, दाएं या जस्टिफाई संरेखित किया जाता है। लेकिन संरेखण केवल टेक्स्ट संरेखण से कहीं अधिक है, इसका उपयोग आपके डिज़ाइन में रुचि जोड़ने के लिए भी किया जाता है। डिज़ाइनर अपने डिज़ाइन में रचनात्मकता जोड़ने के लिए संरेखण का उपयोग करते हैं। अच्छा संरेखण उचित पठनीयता सुनिश्चित करता है जो बदले में आपके संदेश के अच्छे संचार को सुनिश्चित करता है।

# DSG Support Multi Solution

## दोहराव

दोहराव मूल रूप से यूजर इंटरफेस को देखते हुए यूजर के दिमाग में एक पैटर्न बनाना है। दोहराव सभी तत्वों के बीच सामंजस्य बनाने की कोशिश में डिज़ाइन की स्थिरता और एकता के बारे में है। दोहराव शायद स्वाभाविक रूप से सहज ज्ञान युक्त डिज़ाइन सिद्धांत में सबसे अधिक इस्तेमाल किया जाता है। यह बेहद असंभव है कि किसी वेबसाइट में मामूली मात्रा में भी दोहराव न हो। दोहराव का सबसे स्पष्ट उदाहरण टाइपोग्राफी है, इसके बारे में सोचें, हर पैराग्राफ को एक ही फॉन्ट आकार में डिज़ाइन किया गया है, हर H1 शीर्षक को हर दूसरे H1 शीर्षक की तरह दिखना चाहिए और हर H2 शीर्षक को हर दूसरे H2 शीर्षक की तरह दिखना चाहिए और इसी तरह, यह एक सामंजस्य बनाता है। दोहराव ब्रांड, शैली, स्थिरता, एक बेहतर उपयोगकर्ता अनुभव और निश्चित रूप से दोहराव बनाने में भी मदद करता है। दोहराव एक वेबसाइट की टाइपोग्राफी, बटनों के सेट, स्टाइलिंग, स्पेसिंग और सबसे महत्वपूर्ण रूप से हेडर और फूटर में एक समान होना चाहिए।

## अंतर

कंट्रास्ट एक डिज़ाइन सिद्धांत है, जहाँ हम रंगों, फॉन्ट्स, टाइपोग्राफी, दोहराव, संरेखण या किसी अन्य चीज़ की मदद से एक या एक से अधिक तत्वों/दृश्यों को दूसरों से अलग करने की कोशिश करते हैं। विचार यह है कि उपयोगकर्ताओं के लिए एक विशेष तत्व या दृश्य को अलग से प्रस्तुत किया जाए। डिज़ाइन के संदर्भ में - कंट्रास्ट का विज़ुअल डिज़ाइन सिद्धांत दो अलग-अलग तत्वों के एक दूसरे से जुड़ाव को संदर्भित करता है। कंट्रास्ट तब होता है जब आप उपयोगकर्ता इंटरफ़ेस में दो ऑब्जेक्ट्स के बीच अंतर देखते हैं और यह अंतर इस बात पर जोर देता है कि वे अलग हैं। इस अंतर का मतलब यह हो सकता है कि वे अलग-अलग श्रेणियों से संबंधित हैं या उनके अलग-अलग कार्य हैं या वे अलग-अलग व्यवहार करते हैं आदि। कंट्रास्ट के बारे में अधिक जानने के लिए आप यह लेख देख सकते हैं - [UI/UX डिजाइन में कलर कंट्रास्ट क्यों मायने रखता है?](#)

## पदानुक्रम

ऐसे डिज़ाइन बनाते समय जो डेटा को क्रमिक क्रम में रखते हैं, डिज़ाइन के सिद्धांतों को शामिल करना विषयों को एक पदानुक्रमित रूप में तदनुसार विभाजित करने के लिए महत्वपूर्ण है। डिज़ाइन के सिद्धांतों का यह अनुप्रयोग उपयोगकर्ता के लिए विषयों के माध्यम से नेविगेट करना आसान बनाता है, एक दृश्य पथ बनाता है जो मूल से शुरू करके और माध्यमिक महत्वपूर्ण पथों की ओर ले जाकर डिज़ाइन के सिद्धांतों का पालन करता है। इस स्पष्टता और नेविगेशनल आसानी को प्राप्त करने के लिए डिज़ाइन के सिद्धांतों को क्रम को परिभाषित करने के लिए विभिन्न फॉन्ट आकारों के उपयोग के माध्यम से लागू किया जा सकता है, जैसे कि सबसे ऊपर वाले को H1 टैग देना, उसके बाद H2, H3, और इसी तरह, डिज़ाइन के सिद्धांतों के अनुसार।

# DSG Support Multi Solution

## ज़ोर

विज़ुअल बनाते समय, आपके काम की विज़ुअल अपील और महत्व को बेहतर बनाने के लिए डिज़ाइन के सिद्धांतों को लागू करना महत्वपूर्ण है। न्यूनतम टेक्स्ट, और इसलिए अधिकतम ग्राफिक्स और इमेज। पोस्ट पर ज़ोर देने के लिए बॉर्डर का उपयोग करें, और उनमें इमेज जोड़ने के लिए फ्रेम का उपयोग करें। यह डिज़ाइन को अच्छी तरह से संरचित और संतुलित बनाता है। डिज़ाइन या कुछ इमेज के चारों ओर बॉर्डर का उपयोग ग्राफिक डिज़ाइन में सर्वोपरि है क्योंकि वे दर्शकों की विज़ुअल रुचि को सिर्फ़ साधारण टेक्स्ट की तुलना में ज़्यादा हद तक आकर्षित करते हैं।

## सफेद स्थान

डिज़ाइन को अव्यवस्थित और गन्दा न दिखाने के लिए, डिज़ाइन के सिद्धांतों के अनुसार, छवि के सौंदर्य और डिज़ाइन को बनाए रखने के लिए कुछ खाली जगह छोड़ना एक अच्छा अभ्यास है। यह दृश्य पदानुक्रम को बढ़ावा देता है और उपयोगकर्ता के लिए अन्य जानकारी के माध्यम से नेविगेट करना आसान बनाता है। खाली जगह को शामिल करने से डिज़ाइन को व्यवस्थित करने में मदद मिलती है और डिज़ाइन अपने आप में अलग दिखता है।

## अनुपात

अनुपात किसी डिज़ाइन में दो तत्वों के बीच आकार और पैमाने को परिभाषित करता है। डिज़ाइन में अनुपात होने से डिज़ाइन में तत्वों को एक अनूठी पहचान देने में मदद मिलती है। तत्वों को एक दूसरे के अनुपात में रखने से उनके रिश्ते को बेहतर ढंग से परिभाषित करने में मदद मिलती है। उनके बीच का संबंध डिज़ाइन को आकर्षक और सौंदर्यपूर्ण बनाता है। यह समान तत्वों को एक साथ रखने में मदद करता है। यह अलग-अलग पहचान वाले डिज़ाइन में कई तत्वों के बीच दृश्य संतुलन बनाए रखने में भी मदद करता है।

## Design Low BandWidth

कम बैंडविड्थ वाले क्षेत्र ऐसे क्षेत्र हैं जहाँ इंटरनेट कनेक्शन की गति और विश्वसनीयता उच्च कॉल या सामान्य ब्राउज़िंग गतिविधियों का समर्थन करने के लिए पर्याप्त नहीं है। ऐसी कठिनाइयों का कारण अपर्याप्त सुविधाओं की अनुपस्थिति, भौगोलिक स्थान या ब्रॉडबैंड सेवाओं की उच्च लागत हो सकती है। अक्सर, ऐसे व्यक्ति पुराने और कम कुशल उपकरणों, महंगे और सीमित डेटा कनेक्शन और धीमे मोबाइल कनेक्शन का उपयोग करते हैं, जिससे इंटरनेट तक पहुँचने की समस्याएँ और भी बढ़ती जाती हैं।

# DSG Support Multi Solution

उपलब्ध जानकारी से पता चलता है कि दुनिया भर में इंटरनेट कनेक्शन की गति में बहुत अंतर है। यदि 2024 में, उत्तरी अमेरिका और पश्चिमी यूरोप, जिनकी औसत डाउनलोड गति 100 एमबीपीएस से अधिक है, विकसित क्षेत्रों का हिस्सा हैं, तो कई विकासशील क्षेत्रों में 10 एमबीपीएस से कम की गति का अनुभव होता है। उदाहरण के लिए, उप-सहारा अफ्रीका और ग्रामीण एशिया के कुछ हिस्सों में 3 एमबीपीएस जितनी कम औसत गति आम है। यह कनेक्टिविटी अंतर ही है जिसकी वजह से स्थिति विकसित होने पर सबसे कमज़ोर समुदायों के लिए नए वेब समाधान डिज़ाइन करना महत्वपूर्ण है।

कम बैंडविड्थ वाले क्षेत्रों में उपयोगकर्ताओं के पास साइट के विशिष्ट उपयोग पैटर्न होते हैं। यह स्थापित है कि कोई भी वेबसाइट जो खुलने में पाँच सेकंड से अधिक समय लेती है, उपयोगकर्ताओं द्वारा छोड़ दिए जाने की संभावना है। लोग ऐसी साइटें पसंद करते हैं जो जल्दी लोड होती हैं और जिनमें केवल आवश्यक सुविधाएँ होती हैं, न कि दिखावट। साथ ही, चूँकि इंटरनेट ज़्यादातर लोगों के लिए महंगा है, इसलिए बहुत ज़्यादा डेटा की ज़रूरत वाली सोशल वेबसाइटें शायद ही कभी फिर से देखी जाती हैं।

कम बैंडविड्थ वाले क्षेत्रों के लिए डिजाइनिंग के मूल सिद्धांत

## Prioritizing Essential Features

कम बैंडविड्थ वाली डिज़ाइन रणनीति की नींव को सादगी के सिद्धांत द्वारा संक्षेपित किया जा सकता है। प्रस्तावित विचार यह है कि फ़ीचर्ड कार्यक्षमताएँ वेबसाइटों को उपयोगी बनाने के लिए पर्याप्त हैं और उपयोगकर्ताओं को सूचना अधिभार में नहीं भेजती हैं। निर्धारित करें कि वेबसाइट का कौन सा पहलू आपके उपयोगकर्ताओं के लिए सबसे महत्वपूर्ण है - चाहे वह पहुँच की गति हो, [नेविगेशन की आसानी हो](#), या विशेष उपकरण हों - और किसी भी अनावश्यक तामझाम को हटा दें।

## Optimizing Content Delivery

सामग्री का कुशल वितरण यह गारंटी देता है कि जानकारी जितनी जल्दी हो सके उपयोगकर्ता तक पहुँच जाए। कैशिंग, फ़ाइल संपीड़न और वर्तमान बैंडविड्थ के लिए अनुकूलित डेटा जैसे दृष्टिकोण भी अनुप्रयोगों के प्रदर्शन



# DSG Support Multi Solution

को बेहतर बना सकते हैं। इन उपायों से लोडिंग गति को बढ़ाने और डेटा ट्रैफ़िक को कम करने का अतिरिक्त लाभ होता है।

## Leveraging Minimalist Design Practices

एक बात जो ध्यान देने योग्य है वह यह है कि न्यूनतम डिजाइन का मतलब सुंदरता को अनदेखा करना नहीं है। वह अव्यवस्थित इंटरफ़ेस, भ्रामक टाइपोग्राफी और जटिल लेआउट का विरोध करता है। एक ही पेज में कई लिंक को संयोजित करने और अनावश्यक घटकों को हटाने से पेज का द्रव्यमान कम हो जाता है, जिससे यह साबित होता है कि कम कनेक्शन वाली वेब साइटों के लिए न्यूनतमवाद संभव है।

## Optimizing Media for Slow Connections

### 1. Image Compression Techniques

वेबपेज पर इस्तेमाल किए जाने वाले तत्वों के मामले में आमतौर पर चित्र सबसे बड़े आकार के होते हैं। छवियों के आकार को कम करने से लोड समय पर अपेक्षाकृत बड़ा प्रभाव पड़ सकता है, बिना किसी गुणवत्ता हानि के। इस समस्या के लिए अलग-अलग समाधान हैं: **TinyPNG** और **JPEGmini** जैसी सेवाओं का उपयोग करना या **WebP** प्रारूप का उपयोग करना। साथ ही, छवि प्रतिक्रियाशीलता उपयोगकर्ताओं के लिए क्लाइंट के डिवाइस के आकार और रिज़ॉल्यूशन के आधार पर छवि डाउनलोड करना संभव बनाती है।

### 2. Alternatives to Video Content

हालाँकि, वीडियो कंटेंट जितना दिलचस्प है, उसके लिए बहुत सारे संसाधनों की आवश्यकता होती है। जहाँ संभव हो, वीडियो को बेहतर-अनुकूलित **GIF**, **इन्फोग्राफ़िक्स** या सादे टेक्स्ट विवरण से बदलें। जब वीडियो महत्वपूर्ण होते हैं, तो अनुकूली स्ट्रीमिंग तकनीकों को न्यूनतम बफरिंग के साथ अंतिम उपयोगकर्ता की कनेक्शन गति के आधार पर वीडियो गुणवत्ता प्रदान करनी चाहिए।

### 3. Leveraging Lightweight Fonts and Graphics

जबकि कस्टम फ्रॉन्ट विकल्प और विस्तृत ग्राफ़िक्स किसी वेब एप्लिकेशन के लोड समय को बदल सकते हैं। सिस्टम से फ्रॉन्ट का उपयोग करना और **SVG** जैसे वेक्टर पर आधारित ग्राफ़िक्स का उपयोग करना भी फ़ाइल

# DSG Support Multi Solution

आकार को बढ़ाए बिना साइट को आकर्षक बनाए रखने के लाभ में योगदान दे सकता है। इसके अलावा, ओवरहेड को बचाने के लिए, फ्रॉन्ट वजन और शैलियों की न्यूनतम संख्या से चिपके रहें, एक से बेहतर।

## Grid System Design

ग्रिड सिस्टम एक ढांचा है जिसका उपयोग ग्राफिक डिज़ाइन में सामग्री को संरचित और सुसंगत लेआउट में व्यवस्थित करने के लिए किया जाता है। इसमें क्षैतिज और ऊर्ध्वाधर रेखाओं की एक श्रृंखला होती है जो पृष्ठ को खंडों में विभाजित करती है, जिससे डिज़ाइनरों को पाठ, चित्र और ग्राफिक्स जैसे तत्वों को इस तरह से व्यवस्थित करने में मदद मिलती है जिससे पठनीयता और दृश्य अपील बढ़ जाती है।

ग्रिड सिस्टम वेब डिज़ाइन, प्रिंट लेआउट और टाइपोग्राफी सहित विभिन्न क्षेत्रों में आवश्यक हैं, जो संतुलित और सामंजस्यपूर्ण रचनाएँ बनाने के लिए एक आधार प्रदान करते हैं। वास्तव में, अध्ययनों से पता चलता है कि संरचित लेआउट वाले डिज़ाइन उपयोगकर्ता की समझ को 50% तक बेहतर बना सकते हैं, जिससे प्रभावी संचार के लिए ग्रिड महत्वपूर्ण हो जाते हैं।

### ग्रिड सिस्टम: परिभाषा

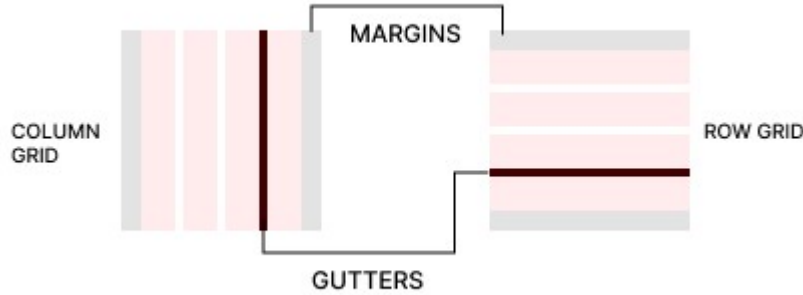
ग्रिड सिस्टम एक ऐसा ढांचा है जो पंक्तियों और स्तंभों की एक श्रृंखला का उपयोग करके वेब पेज या डिज़ाइन लेआउट पर सामग्री की व्यवस्था करने की अनुमति देता है। यह जानकारी को सुसंगत और संगठित तरीके से संरचित करने का एक तरीका है, जिससे सामग्री को पढ़ना, समझना और उससे बातचीत करना आसान हो जाता है। ग्रिड सिस्टम का व्यापक रूप से [ग्राफिक डिज़ाइन](#), [वेब डेवलपमेंट](#) और [प्रिंट मीडिया](#) में नेत्रहीन आकर्षक और कार्यात्मक लेआउट बनाने के लिए उपयोग किया जाता है।

### ग्रिड सिस्टम की डिजाइन विशेषताएं

- **कॉलम:** ऊर्ध्वाधर कंटेनर जो छवियों या पाठ जैसे कुछ तत्व रखते हैं। ग्रिड में अधिकांश क्षेत्र पर कब्जा करता है। स्क्रीन के आधार पर कॉलम की संख्या भिन्न होती है।  
उदाहरण के लिए, मोबाइल स्क्रीन में ग्रिड में 4 कॉलम होते हैं और लैपटॉप/डेस्कटॉप स्क्रीन में ग्रिड में 12 कॉलम होते हैं। नीचे कॉलम कैसे दिखते हैं इसका एक उदाहरण दिया गया है।
- **पंक्तियाँ:** क्षैतिज कंटेनर जो छवियों या पाठ जैसे कुछ तत्व रखते हैं। यह एक कॉलम के समान है, लेकिन जब तक आवश्यक न हो, इसका ज्यादातर उपयोग नहीं किया जाता है।
- **गटर:** 2 कॉलम या पंक्तियों के बीच की जगह को "गली" भी कहा जाता है। गटर के मान निर्धारित होते हैं लेकिन स्क्रीन के आधार पर वे भी अलग-अलग होते हैं। लैपटॉप या डेस्कटॉप जैसी बड़ी स्क्रीन में चौड़े गटर होते हैं जबकि मोबाइल फोन जैसी छोटी स्क्रीन में पतले गटर होते हैं। कॉलम और गटर का संयोजन स्क्रीन की क्षैतिज चौड़ाई को घेरता है।

# DSG Support Multi Solution

- **मार्जिन:** स्क्रीन के बाईं और दाईं ओर की जगह जहाँ कोई स्क्रीन तत्व मौजूद नहीं है। मार्जिन स्क्रीन को घना बनाने में मदद करते हैं लेकिन अव्यवस्थित नहीं। मार्जिन का आकार गटर के समान हो भी सकता है और नहीं भी, आम तौर पर यह गटर से ज़्यादा होता है। यह स्क्रीन तत्वों के लिए सीमा को चिह्नित करता है। लेआउट को ठीक करने में मदद करता है।



*ग्रिड प्रणाली में प्रयुक्त शब्दावलियाँ*

ये कुछ बुनियादी शब्दावली थीं जिन्हें एक डिजाइनर को ग्रिड सिस्टम का उपयोग करने से पहले जानना चाहिए। एक बात ध्यान में रखनी चाहिए कि ग्रिड सिस्टम का उपयोग करते समय "तत्वों को एक ही मार्किंग पर समाप्त होने की आवश्यकता नहीं है, लेकिन सभी तत्वों को एक ही मार्किंग पर शुरू होना चाहिए"।

व्यावसायिक डिज़ाइन बनाने के लिए ग्रिड के प्रकार

चूंकि ग्रिड अलग-अलग आकार के डिवाइस के लिए एक समान लेआउट बनाने के लिए लचीला है। यह कई तरह के ग्रिड सिस्टम का उपयोग करने के लिए अलग-अलग तरीकों से भी उपलब्ध है। अब आइए UI डिज़ाइन में मौजूद विभिन्न प्रकार के ग्रिड सिस्टम पर नज़र डालें और जानें कि उनका उपयोग कैसे करें:

## 1. सममित ग्रिड:

यूआई डिज़ाइन में इस प्रकार की ग्रिड प्रणाली समान रूप से फैली हुई है, जिसका अर्थ है कि **सभी तत्व केंद्र के साथ संरेखित हैं**। इस तरह की ग्रिड प्रणाली में पंक्तियों और स्तंभों की समान संख्या होती है। ये देखने में बहुत ही आकर्षक लगते हैं और तत्व स्क्रीन पर समान रूप से फैले होते हैं। डिज़ाइन को एक औपचारिक और साफ-सुथरा रूप दें।

## कॉलम ग्रिड:

सबसे ज़्यादा इस्तेमाल किया जाने वाला ग्रिड प्रकार, जहाँ एक फ़्रेम को गटर से अलग किए गए कई कॉलम में विभाजित किया जाता है। यह आंखों को भाता है और एक संतुलित दृश्य पदानुक्रम बनाता है। इन कॉलम का उपयोग करके टेक्स्ट, इमेज और बटन जैसे स्क्रीन तत्वों को संरेखित किया जाता है।

# DSG Support Multi Solution

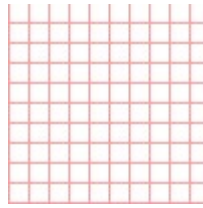
कॉलम ग्रिड का इस्तेमाल ज्यादातर पत्रिकाओं और समाचार पत्रों में किया जाता है। मार्जिन तय होते हैं और स्वतंत्र रूप से आकार के होते हैं।



कॉलम ग्रिड

पिक्सेल ग्रिड:

यह एक प्रकार का सममित ग्रिड है, जहाँ लेआउट को स्तंभों और पंक्तियों के साथ तय किया जाता है जो इसे ग्रिड पिंजरे की तरह बनाते हैं। ऐसा लगता है जैसे किसी ने फिगमा में सुपर ज़ूम किया हो या हमारे गणित के प्राथमिक नोटबुक की तरह। यह ग्रिड प्रकार पिक्सेल-परफेक्ट डिज़ाइन बनाने में मदद करता है, प्रत्येक बॉक्स एक पिक्सेल को दर्शाता है। डिज़ाइनर कभी-कभी डिजिटल स्क्रीन बनाने वाले लाखों पिक्सेल में पिक्सेल-दर-पिक्सेल समायोजन करने के लिए करीब आते हैं।



पिक्सेल ग्रिड

आधार रेखा ग्रिड:

यह एक प्रकार का सममित ग्रिड भी है, जहाँ लेआउट तय होता है जिसमें केवल क्षैतिज रेखाएँ ही मार्गदर्शक के रूप में होती हैं, एक रूल्ड नोटबुक एक बेसलाइन ग्रिड का सीधा चित्रण है। यह सुनिश्चित करने के लिए कि प्रत्येक कॉलम में पाठ की पंक्तियाँ एक प्रसार में लगातार संरेखित हों, बेसलाइन ग्रिड का उपयोग अक्सर कॉलम ग्रिड के साथ किया जाता है।

**Hello**

This is baseline

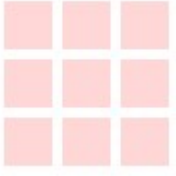
Grid

मॉड्यूलर ग्रिड:

यह एक और प्रकार का सममित ग्रिड है जो आंखों को बहुत ही अच्छा लगता है। मॉड्यूलर ग्रिड को यह नाम "मॉड्यूल" के कारण मिला है जो स्तंभों और पंक्तियों के प्रतिच्छेदन द्वारा बनाए जाते हैं जो बहुत

# DSG Support Multi Solution

अच्छी तरह से व्यवस्थित होते हैं और गटर का उपयोग करके रखे जाते हैं। मार्जिन सुसंगत होते हैं जो एक बॉक्सी लुक देते हैं, इस प्रकार का उपयोग आम तौर पर व्यावसायिक पत्रिकाओं द्वारा किया जाता है।



## पाण्डुलिपि ग्रिड:

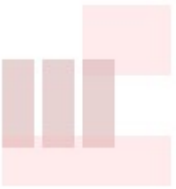
यह एक प्रकार का सममित ग्रिड है, जिसे "एकल स्तंभ" या "एक-स्तंभ" ग्रिड के रूप में भी जाना जाता है। जैसा कि नाम से पता चलता है, पाण्डुलिपि ग्रिड में केवल एक कॉलम होता है जो लेआउट को निर्देशित करता है। इस प्रकार के ग्रिड का उपयोग अक्सर पुस्तकों, विशेष रूप से लोक या पारंपरिक पुस्तकों के लिए किया जाता है। इसकी उपस्थिति के कारण, यह सभी तरफ से निश्चित मार्जिन के साथ छोटी लाइनों का उपयोग करके पढ़ना आसान बनाता है।



## 2. असममित ग्रिड:

यूआई डिज़ाइन में इस प्रकार की ग्रिड प्रणाली सममित ग्रिड के विपरीत होती है क्योंकि इसमें ग्रिड को केंद्र में संरेखित करने का कोई प्रतिबंध नहीं होता है। इसे टूटा हुआ ग्लास ग्रिड भी कहा जाता है, क्योंकि टुकड़ों को असमान रूप से रखा जाता है जो डिज़ाइन को एक शानदार, आधुनिक और अमूर्त रूप देता है।

**पदानुक्रमित ग्रिड:** यह एक प्रकार का असममित ग्रिड है, जहाँ लेआउट तय नहीं होता है। एक पदानुक्रमित ग्रिड पूरी तरह से मुक्त रूप है, जो दो या अधिक ग्रिड से बना होता है जो ओवरले होते हैं, या अपने भीतर अन्य ग्रिड घटक होते हैं। बॉक्सी वाइब से बचने के लिए, कई समकालीन वेबसाइट इंटरफ़ेस पदानुक्रमित ग्रिड लेआउट का उपयोग करते हैं। चूंकि यह असममित है, इसलिए यह अधिक आधुनिक और अमूर्त है, जिसका उपयोग करने पर वेबसाइटों पर एक चंचल मूड सेट होता है।



# DSG Support Multi Solution

ये कुछ लोकप्रिय प्रकार के ग्रिड हैं जिनका उपयोग विभिन्न आवश्यकताओं और जरूरतों के आधार पर किया जाता है। इनमें से प्रत्येक के बारे में जानना अच्छा है ताकि एक डिजाइनर के रूप में कोई भी सबसे अच्छा मार्गदर्शक लेआउट ग्रिड सिस्टम का उपयोग और लाभ उठा सके।

ग्रिड सिस्टम के इष्टतम उपयोग को जानने के लिए, ग्रिड का उपयोग कैसे करें, यह जानना बहुत महत्वपूर्ण है। एक शुरुआती के रूप में ग्रिड का उपयोग करना बहुत भारी और भ्रमित करने वाला हो सकता है, लेकिन चिंता न करें, हम आपकी मदद करने के लिए यहाँ हैं। अब आइए देखें कि एक संगठित और सुसंगत लेआउट बनाने के लिए ग्रिड का उपयोग कैसे करें।

## डिज़ाइन में ग्रिड सिस्टम का उपयोग कैसे करें

डिज़ाइन में ग्रिड का उपयोग करने से आपके प्रोजेक्ट का संगठन और दृश्य अपील काफी हद तक बढ़ सकता है। ग्रिड का प्रभावी ढंग से उपयोग करने के तरीके के बारे में चरण-दर-चरण मार्गदर्शिका यहाँ दी गई है:

### चरण 1: ग्रिड सिस्टम का चयन करें

अपने प्रोजेक्ट के लिए उपयुक्त ग्रिड सिस्टम चुनें। सामान्य प्रकारों में टेक्स्ट लेआउट के लिए कॉलम ग्रिड, जटिल ग्राफिक इंटरफ़ेस के लिए मॉड्यूलर ग्रिड और टेक्स्ट को संरेखित करने के लिए बेसलाइन ग्रिड शामिल हैं।

### चरण 2: ग्रिड पैरामीटर परिभाषित करें

अपने ग्रिड के पैरामीटर स्थापित करें, जैसे कि कॉलम की संख्या, गटर की चौड़ाई (कॉलम के बीच की जगह), और कंटेंट एरिया के चारों ओर मार्जिन। इन ग्रिड को सटीक रूप से सेट करने के लिए **Adobe Photoshop**, **Illustrator**, **Sketch** या **Figma** जैसे डिज़ाइन सॉफ़्टवेयर टूल का उपयोग करें।

### चरण 3: तत्वों को ग्रिड पर रखें

अपने डिज़ाइन तत्वों - टेक्स्ट, इमेज, आइकन और अन्य - को ग्रिड कॉलम में संरेखित करें। यह आपके डिज़ाइन में स्थिरता और संतुलन बनाए रखने में मदद करता है, जिससे नेविगेट करना आसान हो जाता है और यह दिखने में आकर्षक लगता है।

### चरण 4: निरंतरता महत्वपूर्ण है

एक सुसंगत रूप और अनुभव सुनिश्चित करने के लिए अपने प्रोजेक्ट के सभी पृष्ठों या अनुभागों पर एक ही ग्रिड लेआउट लागू करें। ग्रिड का लगातार उपयोग एक लय स्थापित करने में मदद करता है जो दर्शक की नज़र को निर्देशित करता है और सामग्री को अधिक सुपाच्य बनाता है।

# DSG Support Multi Solution

चरण 5: जब आवश्यक हो तो समायोजित करें

हालांकि ग्रिड से चिपके रहना महत्वपूर्ण है, लेकिन जब कुछ तत्वों को उजागर करने या दृश्य रुचि जोड़ने की आवश्यकता हो तो उससे अलग होने में संकोच न करें। ग्रिड को तोड़ने से आपके डिज़ाइन के विशिष्ट भागों पर ध्यान आकर्षित हो सकता है और गतिशीलता बढ़ सकती है।

चरण 6: परीक्षण और पुनरावृत्ति

अपने डिज़ाइन को अलग-अलग डिवाइस (अगर डिजिटल है) या अलग-अलग फॉर्मेट (अगर प्रिंट है) पर रिव्यू करें, ताकि पता चल सके कि ग्रिड लेआउट को कितनी अच्छी तरह से पूरा करता है। उपयोगकर्ता अनुभव को बेहतर बनाने के लिए फीडबैक और व्यावहारिक परीक्षण के आधार पर समायोजन करें।

ग्रिड लेआउट वेबसाइट उदाहरण

जब आप वेब डिज़ाइन की खोज कर रहे हों, तो ग्रिड लेआउट कैसे काम करते हैं, यह समझना वास्तव में आपकी रचनात्मकता और दक्षता का विस्तार कर सकता है। यहाँ वेबसाइट प्रकारों के कुछ क्लासिक उदाहरण दिए गए हैं जो अक्सर ग्रिड लेआउट का प्रभावी ढंग से उपयोग करते हैं:

- 1. पत्रिका वेबसाइटें :** ये साइटें आमतौर पर **लेखों**, **शीर्षकों** और **छवियों** को व्यवस्थित करने के लिए ग्रिड लेआउट का उपयोग करती हैं, जो नेविगेट करने में आसान और देखने में आकर्षक होती हैं। **उदाहरण : वोग या वायर्ड के ऑनलाइन संस्करणों का लेआउट ।**
- 2. ई-कॉमर्स साइट्स :** उत्पादों को साफ-सुथरा दिखाने के लिए ग्रिड लेआउट बहुत ज़रूरी हैं। प्रत्येक उत्पाद को आम तौर पर ग्रिड में अपना बॉक्स मिलता है, जिससे उपयोगकर्ताओं के लिए ब्राउज़ करना आसान हो जाता है। **उदाहरण : Amazon और Etsy** अपने उत्पादों को ग्रिड लेआउट का उपयोग करके प्रदर्शित करते हैं।
- 3. पोर्टफोलियो वेबसाइट :** कलाकार और डिज़ाइनर अपने काम को **व्यवस्थित**, **साफ-सुथरे तरीके** से पेश करने के लिए ग्रिड लेआउट का इस्तेमाल करते हैं, जिससे विज़िटर बिना किसी अव्यवस्था के हर काम की सराहना कर पाते हैं। **उदाहरण : बेहांस या किसी पेशेवर फ़ोटोग्राफ़र की पोर्टफोलियो साइट।**
- 4. समाचार वेबसाइटें :** पत्रिकाओं की तरह ही, समाचार वेबसाइटें भी विविध प्रकार की सामग्री - **पाठ**, **चित्र** और **वीडियो** - को संभालने के लिए ग्रिड का उपयोग करती हैं, जिससे यह सुनिश्चित होता है कि मुखपृष्ठ एक नज़र में कई विषयों का स्नैपशॉट प्रदान करता है। **उदाहरण : CNN और BBC** श्रेणी के अनुसार समाचार कहानियों को व्यवस्थित करने के लिए ग्रिड लेआउट का उपयोग करते हैं।
- 5. ब्लॉग :** ग्रिड लेआउट ब्लॉग को पोस्ट की एक साधारण सूची से एक आकर्षक दृश्य संग्रह में बदल सकते हैं, जिससे पाठकों के लिए यह चुनना आसान हो जाता है कि उन्हें कौन सी पोस्ट पढ़नी है। **उदाहरण :**



# DSG Support Multi Solution

मीडियम पढ़ने के समय और लोकप्रियता के आधार पर कहानियों को दिखाने के लिए ग्रिड लेआउट का उपयोग करता है।

6. **शैक्षिक साइटें** : ये साइटें पाठ्यक्रमों, संसाधनों और सूचनाओं को प्रभावी ढंग से व्यवस्थित करने के लिए ग्रिड का उपयोग करती हैं, जिससे छात्रों के लिए अपनी ज़रूरत की चीज़ें ढूँढना आसान हो जाता है। **उदाहरण** : **खान अकादमी** विषयों और पाठ्यक्रमों को व्यवस्थित करने के लिए ग्रिड लेआउट का उपयोग करती है।
7. **सोशल मीडिया प्लेटफॉर्म** : **पिनटरेस्ट** और **इंस्टाग्राम** जैसे प्लेटफॉर्म **ग्रिड लेआउट** के प्रमुख उदाहरण हैं , जहां प्रत्येक पोस्ट या छवि ग्रिड सेल में फिट हो जाती है, जिससे व्यापक सामग्री को व्यवस्थित तरीके से ब्राउज़ किया जा सकता है।

## वेब विकास में ग्रिड सिस्टम

वेब डेवलपमेंट में, रिस्पॉन्सिव लेआउट बनाने के लिए ग्रिड सिस्टम बहुत ज़रूरी हैं। **बूटस्ट्रैप** और **CSS ग्रिड** जैसे फ्रेमवर्क पहले से परिभाषित ग्रिड सिस्टम प्रदान करते हैं जो डेवलपर्स को तरल और अनुकूली डिज़ाइन बनाने में मदद करते हैं।

## सीएसएस ग्रिड

- **सीएसएस ग्रिड** सीएसएस में एक शक्तिशाली लेआउट प्रणाली है जो डेवलपर्स को पंक्तियों और स्तंभों के साथ दो-आयामी लेआउट बनाने की अनुमति देती है।
- यह फ्लोट्स जैसी पारंपरिक लेआउट विधियों की तुलना में तत्वों की स्थिति और संरेखण पर अधिक नियंत्रण प्रदान करता है।

## बूटस्ट्रैप ग्रिड

- **बूटस्ट्रैप** एक लोकप्रिय फ्रंट-एंड फ्रेमवर्क है जिसमें **12-कॉलम ग्रिड सिस्टम** शामिल है ।
- यह विभिन्न स्क्रीन आकारों के अनुकूल उत्तरदायी लेआउट बनाने के लिए **मीडिया क्वेरीज़** और **फ्लेक्सबॉक्स** के संयोजन का उपयोग करता है ।

## Intreccion & UX Design

इंटरैक्शन डिज़ाइन (IxD) और यूजर एक्सपीरियंस डिज़ाइन (UX डिज़ाइन) शब्दों का अक्सर साथ-साथ उल्लेख किया जाता है, जिससे दुविधा पैदा होती है क्योंकि लोग एक के बजाय दूसरे को चुनते हैं और दोनों के बीच अंतर देखने में विफल रहते हैं। वे समान हैं लेकिन फिर भी डिज़ाइन कार्य के प्रक्रिया और कार्य दोनों पक्षों को कवर करने के लिए अलग-अलग काम करते हैं। इंटरैक्टिव डिज़ाइन (IxD) और यूजर एक्सपीरियंस डिज़ाइन (UX डिज़ाइन) डिजिटल डिज़ाइन के विशाल क्षेत्र में दो अविभाज्य घटक हैं।

# DSG Support Multi Solution

उपभोक्ता व्यवहार और उपयोगकर्ता इंटरैक्शन का मुख्य उद्देश्य तकनीकी उत्पादों का उपयोग करने में उपयोगकर्ता की खुशी को बढ़ाना है और साथ ही वे उपयोगकर्ता-उत्पाद इंटरैक्शन के विभिन्न हिस्सों को संबोधित करते हैं। उनके अंतरों का वर्णन अनुशासन में काम करने वाले पेशेवरों के लिए एक आधारशिला है, न केवल स्वीकार करने के लिए बल्कि यात्रा मार्गों को व्यवस्थित करने के लिए जो सहज और स्पष्ट हैं।

## इंटरैक्शन डिज़ाइन क्या है?

इंटरएक्टिव डिज़ाइन (IXD) उपयोगकर्ता अनुभव डिज़ाइन (UX) के भीतर एक संकीर्ण विशेषीकृत डोमेन है, जहाँ विशेषज्ञता डिजिटल उत्पादों और आउटपुट के डिज़ाइन में है, जिसमें वेबसाइट, मोबाइल ऐप और सॉफ्टवेयर उपयोगकर्ता इंटरफ़ेस शामिल हैं। इसमें उपयोगकर्ता की इंटरफ़ेस संरचना और व्यवहार दोनों का निर्माण शामिल है ताकि उपयोगकर्ता की सहज और सफल क्रियाएँ और साथ ही प्रभावी प्रयोज्यता प्राप्त हो सके। इंटरैक्शन डिज़ाइनर, जो इंटरफ़ेस घटकों, नियंत्रणों और डिजिटल उत्पाद के साथ उपयोगकर्ताओं की सहभागिता को विकसित करने के लिए जिम्मेदार हैं, इस प्रक्रिया के भी प्रभारी हैं। इसमें बटन, मेनू, फॉर्म, नेविगेशन सिस्टम और अन्य इनपुट का निर्माण शामिल है जो सहज, सुरक्षित और देखने में मनभावन हैं। इसके अलावा, इंटरैक्शन डिज़ाइनर कार्य प्रवाह और क्रियाओं के क्रम के निर्माता हैं जो डिजिटल उत्पाद के अंदर उपयोगकर्ताओं के कार्य लक्ष्यों की स्व-प्राप्ति को संभव बनाते हैं।

## इंटरैक्शन डिज़ाइन की विशेषताएं

- **उपयोगकर्ता-केंद्रित दृष्टिकोण:** उपयोगकर्ताओं की ज़रूरतों, व्यवहार और स्वाद को निष्पक्ष रूप से संबोधित करना इंटरैक्शन डिज़ाइन की भूमिका है। उपयोगकर्ता केंद्रित डिज़ाइन निर्णय लेने के लिए एक रूपरेखा प्रदान करता है जिसमें सुविधा पूर्णता और उच्च उपयोगकर्ता रिसेप्शन प्राप्त करने के लिए उपयोगकर्ताओं के लक्ष्यों, प्रेरणा और कार्यों को समझना शामिल है।
- **सहज ज्ञान युक्त इंटरफ़ेस:** इंटरैक्शन डिज़ाइनर हमेशा ऐसे इंटरफ़ेस बनाने में लगे रहते हैं जो जटिल न हों और जिनका पालन करना आसान हो। यह सामग्री को इस तरह से व्यवस्थित करके प्राप्त किया जाता है कि उपयोगकर्ता इसे आसानी से समझ सकें, रूपरेखा को स्पष्ट रूप से बताकर, और कार्यों और इंटरैक्शन को सुविधाजनक बनाने के लिए दृश्य संकेत प्रदान करके।
- **रिस्पॉन्सिव डिज़ाइन:** यह वह जगह है जहाँ इंटरैक्शन डिज़ाइन एक महत्वपूर्ण भूमिका निभाता है ताकि इंटरफ़ेस विकसित किया जा सके जो कई डिवाइस और स्क्रीन पर प्रतिक्रिया दे सके। यहाँ, यह एप्लिकेशन को उपयोगकर्ता के अनुभव को प्रभावित किए बिना विभिन्न डिवाइस पर संचालित करने की अनुमति देता है, चाहे वे डेस्कटॉप कंप्यूटर, टैबलेट या स्मार्टफोन का उपयोग कर रहे हों।

# DSG Support Multi Solution

- **फीडबैक तंत्र:** उपयोगकर्ताओं को फीडबैक देना उन्हें परिणाम बताने और उन्हें सही तरीके से निर्देश देने के लिए महत्वपूर्ण है कि वे बातचीत प्रक्रिया के माध्यम से कैसे आगे बढ़ सकते हैं। इंटरैक्शन डेवलपर्स दृश्य, श्रवण और स्पर्शनीय फीडबैक अनुप्रयोगों को परिभाषित करते हैं जो उपयोगी क्रियाओं, त्रुटियों और स्थिति अपडेट को उजागर करते हैं।
- **इंटरएक्टिव तत्व:** इंटरैक्शन डिज़ाइन में ग्राफिकल यूजर इंटरफ़ेस (GUI) तत्वों का विकास शामिल है, उदाहरण के लिए, बटन, लिंक, फॉर्म और मेनू। इन तत्वों को दृष्टिगत रूप से अलग किया जाना चाहिए, आसानी से पहचाना जा सकता है और उपयोगकर्ता संचालन के अनुसार प्रतिक्रिया करनी चाहिए, ताकि सरल उपयोगकर्ता इंटरैक्शन की सुविधा मिल सके।

## UX डिजाइन क्या है?

उपयोगकर्ता अनुभव डिजाइन (UX डिजाइन) एक बहु-विषयक क्षेत्र है जिसमें सभी विषयों का समावेश होता है जिसमें उत्पादों, प्रणालियों या सेवाओं का डिजाइन, उत्पादन या विकास शामिल होता है, जिसका मूल उद्देश्य उपयोगकर्ताओं को सबसे सार्थक, प्रासंगिक और आकर्षक अनुभव प्रदान करना होता है। UX डिजाइन में उपयोगकर्ता पथ, बाहरी-व्यावसायिक इंटरैक्शन डिजाइन, डेटा आर्किटेक्चर, प्रयोज्यता, ग्राफिक डिजाइन और HCI के विविध भाग शामिल होते हैं। UX डिज़ाइनर का उद्देश्य किसी उत्पाद या सेवा के साथ बातचीत करते समय उपयोगकर्ता के सामान्य अनुभव को बेहतर बनाना होता है और यह सुनिश्चित करके किया जा सकता है कि उपयोगकर्ता की ज़रूरतें पूरी हों, उत्पाद उपयोग में आसान हों और उपयोगकर्ताओं को अतिरिक्त मूल्य प्रदान करें।

## UX डिज़ाइन की विशेषताएं

- उपयोगकर्ता अनुभव (यूएक्स) डिजाइन का उद्देश्य विभिन्न विशेषताओं का निर्माण करना है जो डिजिटल उत्पादों या सेवाओं के साथ बातचीत करते समय उपयोगकर्ताओं के लिए जीवन भर उपयोगी और आनंददायक अनुभव का निर्माण करता है।
- **उपयोगकर्ता-केंद्रित दृष्टिकोण:** UX डिज़ाइन का पूरा विचार उपयोगकर्ता की ज़रूरतों की प्रकृति, जिस तरह से ये लोग अपने पर्यावरण के साथ बातचीत कर रहे हैं, और उनकी व्यक्तिगत प्राथमिकताओं को अच्छी तरह से समझने पर केंद्रित है। एक विशिष्ट डिज़ाइन उपयोगकर्ता अनुसंधान पर आधारित है, जिसका अर्थ है कि उत्पाद उपयोगकर्ता की ज़रूरतों को तदनुसार संबोधित करता है और अंतिम उपयोगकर्ता के लिए आवश्यक मूल्य प्रदान करता है।
- **शोध और विश्लेषण:** UX डिज़ाइनर उपयोगकर्ता की जनसांख्यिकी, आदतों और उनके द्वारा उपयोग किए जाने वाले उत्पाद को चुनने के कारणों के बारे में जानकारी प्राप्त करने के लिए शोध करते हैं। यह उपयोगकर्ताओं से बात करने और प्रयोज्यता परीक्षण, और सर्वेक्षण और विश्लेषण (मात्रात्मक विधियाँ) जैसी विधियाँ हो सकती हैं।

# DSG Support Multi Solution

- **सूचना वास्तुकला:** UX डिज़ाइनर ऐसी संरचना और व्यवस्था बनाते हैं जो आसान पहुँच के सिद्धांत का पालन करके सामग्री को सरल और परेशानी मुक्त बनाए रखेगी। इस प्रकार साइट मैप, वायरफ्रेम और साथ ही नेविगेशन सिस्टम बनाते हैं जो उपयोगकर्ताओं को इंटरैक्टिव और तार्किक यात्राएँ करने की अनुमति देते हैं।

## इंटरैक्शन डिज़ाइन और UX डिज़ाइन के बीच अंतर

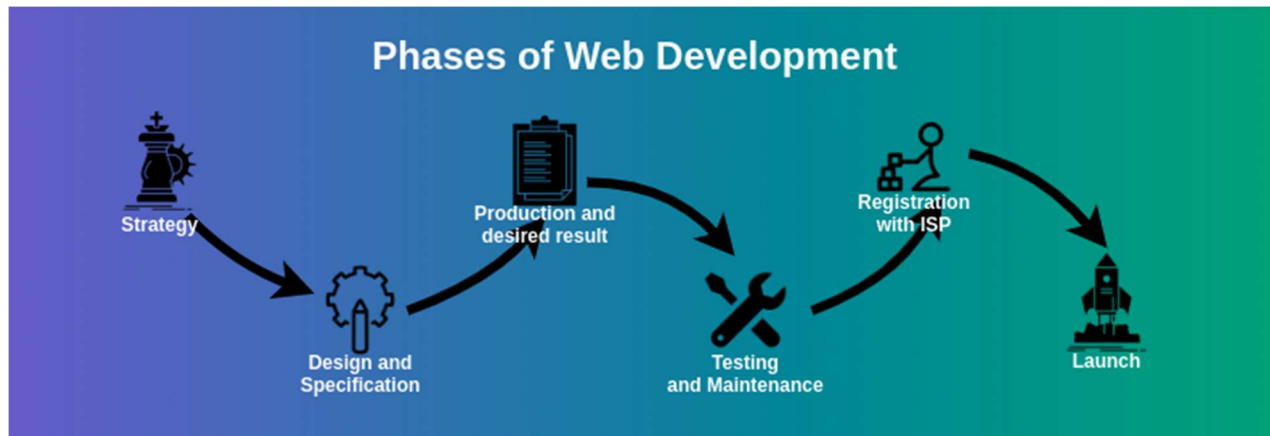
इंटरैक्शन डिज़ाइन	उपयोगकर्ता अनुभव डिज़ाइन
डिजिटल उत्पादों के भीतर इंटरफ़ेस तत्वों और सूक्ष्म-इंटरैक्शन को तैयार करने पर ध्यान केंद्रित करता है।	इसमें उपयोगकर्ता अनुसंधान, सूचना वास्तुकला, प्रयोज्यता परीक्षण और समग्र उत्पाद रणनीति सहित व्यापक दायरा शामिल है।
प्राथमिक फोकस - इंटरफ़ेस तत्व, माइक्रो-इंटरैक्शन और दृश्य डिज़ाइन।	प्राथमिक फोकस - प्रारंभिक संलग्नता से लेकर कार्य पूरा होने तक संपूर्ण उपयोगकर्ता यात्रा।
प्रकार - इंटरफ़ेस तत्व डिज़ाइन, माइक्रो-इंटरैक्शन, दृश्य डिज़ाइन।	प्रकार - उपयोगकर्ता अनुसंधान, सूचना वास्तुकला, प्रयोज्यता परीक्षण, उत्पाद रणनीति।
डिजिटल इंटरफ़ेस के भीतर प्रयोज्यता को बढ़ाना और उपयोगकर्ता की क्रियाओं का मार्गदर्शन करना।	उपयोगकर्ता यात्रा के सभी स्पर्श-बिंदुओं पर निर्बाध और आनंददायक अनुभव का निर्माण करना।
विस्तार-उन्मुख, व्यक्तिगत इंटरफ़ेस तत्वों और अंतःक्रियाओं के डिज़ाइन पर ध्यान केंद्रित करना।	समग्र, उपयोगकर्ता की आवश्यकताओं, व्यवहारों और प्रेरणाओं के व्यापक संदर्भ पर विचार करना।
विधियाँ - वायरफ्रेमिंग, प्रोटोटाइपिंग, विज़ुअल डिज़ाइन, एनीमेशन डिज़ाइन।	विधियाँ - उपयोगकर्ता साक्षात्कार, सर्वेक्षण, प्रयोज्यता परीक्षण, ए/बी परीक्षण, व्यक्तित्व विकास।
संकीर्ण, उपयोगकर्ता इंटरफ़ेस और इंटरैक्शन डिज़ाइन के विशिष्ट पहलुओं पर ध्यान केंद्रित करना।	व्यापक, इंटरफ़ेस डिज़ाइन सहित उपयोगकर्ता अनुभव के सभी पहलुओं को संबोधित करना।

# DSG Support Multi Solution

इंटरैक्शन डिज़ाइन	उपयोगकर्ता अनुभव डिजाइन
सहज और आकर्षक इंटरफ़ेस डिज़ाइन जो सहज उपयोगकर्ता इंटरैक्शन को सुविधाजनक बनाता है।	सुसंगत और सार्थक उपयोगकर्ता अनुभव जो उपयोगकर्ताओं के साथ प्रतिध्वनित होते हैं और उनकी आवश्यकताओं को पूरा करते हैं।
इंटरफ़ेस डिज़ाइन से परे व्यापक उपयोगकर्ता अनुभव मुद्दों को नजरअंदाज किया जा सकता है।	अनुसंधान की गहराई और उपयोगकर्ता अनुभव के सभी पहलुओं को संबोधित करने की व्यापकता के बीच संतुलन की आवश्यकता है।
सुसंगत दृश्य डिजाइन के माध्यम से बेहतर प्रयोज्यता, उन्नत ब्रांड प्रतिनिधित्व।	निर्बाध और आनंददायक अनुभव के माध्यम से उपयोगकर्ता संतुष्टि में वृद्धि, प्रतिधारण और वफादारी में सुधार।
उदाहरण - मोबाइल ऐप इंटरफ़ेस डिज़ाइन, वेबसाइट नेविगेशन मेनू, बटन डिज़ाइन।	उदाहरण - ई-कॉमर्स वेबसाइट उपयोगकर्ता यात्रा अनुकूलन, यात्रा बुकिंग प्लेटफॉर्म उपयोगकर्ता अनुभव संवर्द्धन।

## Planning in Website

वेब डेवलपमेंट एक ऐसा शब्द है जिसमें वेब प्रोजेक्ट या वेबसाइट विकसित करने में शामिल सभी प्रक्रियाएँ शामिल हैं। इसमें वेब प्रोजेक्ट की योजना, डिजाइनिंग, परीक्षण और लॉन्चिंग जैसे विभिन्न चरण शामिल हैं। वेब डेवलपमेंट प्रक्रिया के लिए विशेषज्ञों की एक टीम की आवश्यकता होती है जो वेबसाइट बनाने के लिए आवश्यक विभिन्न कार्यों को लागू करने के लिए जिम्मेदार होती है



वेब विकास में वेब परियोजना विकसित करने के लिए आवश्यक विभिन्न चरण निम्नलिखित हैं:

# DSG Support Multi Solution

**रणनीति:** किसी डेवलपर के लिए वेब डेवलपमेंट प्रक्रिया में पहला कदम वेब पेज या वेब साइट विकसित करने के लिए रणनीति बनाना है। रणनीति चरण में, वेब डेवलपर को निम्नलिखित कार्य करने होते हैं:

- लक्ष्य और उद्देश्य तय करना
- विकासशील टीम
- समस्या से संबंधित उचित विश्लेषण करें और विश्लेषण की समीक्षा करें
- कार्यों की सूची तैयार करें
- वेब टीम को विकास हेतु परियोजना का प्रस्ताव

**डिजाइन और विनिर्देश:** रणनीति बनाने के बाद, वेब डेवलपमेंट प्रक्रिया में अगला चरण योजनाबद्ध कार्य विकसित करना है। वेब डेवलपर को शेड्यूल और विनिर्देश निर्धारित करने होते हैं। इस चरण में कार्य इस प्रकार हैं।

- दृष्टिकोण विकसित करना
- उपयोग के लिए आवश्यक सामग्री की योजना बनाना
- परियोजना के लिए मोटा डिजाइन बनाना
- यदि कच्चे डिजाइन में कोई महत्वपूर्ण संशोधन नहीं किया गया हो तो कच्चे डिजाइन से अंतिम डिजाइन बनाना।
- विकास के लिए परियोजना का एक प्रोटोटाइप तैयार करें
- प्रोटोटाइप का परीक्षण करें

यदि प्रोटोटाइप पूरा हो जाता है, तो अगले चरण चरण-3 पर जाएं अन्यथा प्रोटोटाइप पूरा होने तक चरण 2 को दोहराएं।

**वांछित परिणाम का उत्पादन:** वेब विकास प्रक्रिया के इस चरण में, वास्तविक कार्यात्मक साइट का निर्माण किया जाता है। प्रोटोटाइप के उचित परीक्षण के बाद, डेवलपर को वास्तविक लाइव वेब प्रोजेक्ट विकसित करने पर काम करना होता है। वास्तविक लाइव वेब प्रोजेक्ट क्लाइंट की आवश्यकताओं के अनुसार बनाया जाता है। वेब डेवलपर को वेब प्रोजेक्ट में सभी सुविधाएँ बनाने के लिए डिजाइन चरण से सभी स्थितियों पर विचार करना होता है। इस चरण में वेबसाइट का फ्रंट एंड डेवलपमेंट और बैक एंड डेवलपमेंट दोनों शामिल हैं। फ्रंट एंड डेवलपमेंट में वेब मानकों के अनुसार HTML, CSS आदि जैसी बुनियादी तकनीकों के साथ कोड लिखना शामिल है। यह आम तौर पर पहले होम पेज और फिर अन्य पेजों को विकसित करके शुरू होता है। इस चरण में कंटेंट मैनेजमेंट सिस्टम, डेटाबेस और फ्रेमवर्क को स्थापित और कॉन्फिगर करके बैक एंड डेवलपमेंट भी पूरा किया जाता है। रणनीति और डिजाइन चरण में अंतिम रूप दिए गए सभी चरणों को पूरा करने के बाद, जिसके द्वारा मूल वेबसाइट कार्यात्मक हो जाती है, अगले चरण में इसका परीक्षण किया जाता है।



# DSG Support Multi Solution

**परीक्षण और रखरखाव:** परीक्षण वेब विकास प्रक्रिया में एक महत्वपूर्ण चरण है। वेब प्रोजेक्ट के पूरा होने के बाद क्लाइंट की आवश्यकताओं को सुनिश्चित करने के लिए डेवलपर्स और परीक्षकों द्वारा परीक्षण किया जाता है। इस चरण में, वेबसाइट की गुणवत्ता आश्वासन और ब्राउज़र संगतता मुद्दों की जाँच की जाती है। परीक्षक सभी विकसित सुविधाओं का परीक्षण करते हैं और लिखित कोड की वैधता का आश्वासन देते हैं। इस चरण में परीक्षण और विकास दोनों टीमों द्वारा एकीकरण परीक्षण, प्रतिगमन परीक्षण, कार्यात्मक परीक्षण, स्मोक परीक्षण, लोड परीक्षण और प्रदर्शन परीक्षण जैसे विभिन्न प्रकार के परीक्षण किए जाते हैं। परीक्षण के प्रकार और वेब प्रोजेक्ट के आधार पर परीक्षण मैनुअल रूप से या स्वचालित रूप से किया जा सकता है। यदि वांछित और संतोषजनक परिणाम नहीं मिलते हैं, तो बग को हटाने के लिए उचित कार्रवाई की जाती है।

**ISP के साथ पंजीकरण:** परीक्षण और रखरखाव पूरा होने और परियोजना से सभी बग्स को हटाने के बाद, अगला चरण या चरण वेब प्रोजेक्ट को वैध बनाने के लिए क्षेत्रीय ISP के साथ वेब प्रोजेक्ट को पंजीकृत करना है। वेबसाइट को सर्वर पर अपलोड करने के बाद वेब प्रोजेक्ट क्लाइंट को डिलीवर किया जाता है। वेबसाइट को होस्टिंग सर्वर पर होस्ट करने के लिए फ़ाइल ट्रांसफ़र प्रोटोकॉल (FTP) का उपयोग किया जाता है। क्लाइंट को उस ISP का चयन और निर्णय लेना होता है जो डोमेन नाम पंजीकरण और वेब होस्टिंग सेवाएँ प्रदान करता है। इन खातों को सेटअप करने और ISP के साथ पंजीकरण करने के बाद ताकि वेब प्रोजेक्ट को ISP सर्वर पर एक सटीक डोमेन स्पेस मिल सके।

**लॉन्च:** यह वेब डेवलपमेंट प्रक्रिया का अंतिम चरण है। ISP के साथ पंजीकृत होने के बाद प्रोजेक्ट लॉन्च किया जाता है। लॉन्च होने के बाद, वेब प्रोजेक्ट को विशेष डोमेन के उपयोगकर्ताओं द्वारा सार्वजनिक रूप से एक्सेस किया जाता है। लॉन्च चरण में किए जाने वाले कार्य इस प्रकार हैं।

- डेटा का स्थानांतरण
- सर्वर का शुभारंभ
- कोड का विलय
- डोमेन नाम पुनर्निर्देशित करना

## Directory Structure

ब्रैकेट्स वास्तव में वह सब है जो हमें वेबसाइट बनाने और कोडिंग शुरू करने के लिए चाहिए, लेकिन ऐसा करने से पहले, हमें वेबसाइट को रखने के लिए एक जगह की आवश्यकता होती है, अर्थात एक फ़ोल्डर संरचना जिसमें हम सभी HTML, CSS, जावास्क्रिप्ट और छवि फ़ाइलों को रख सकें।

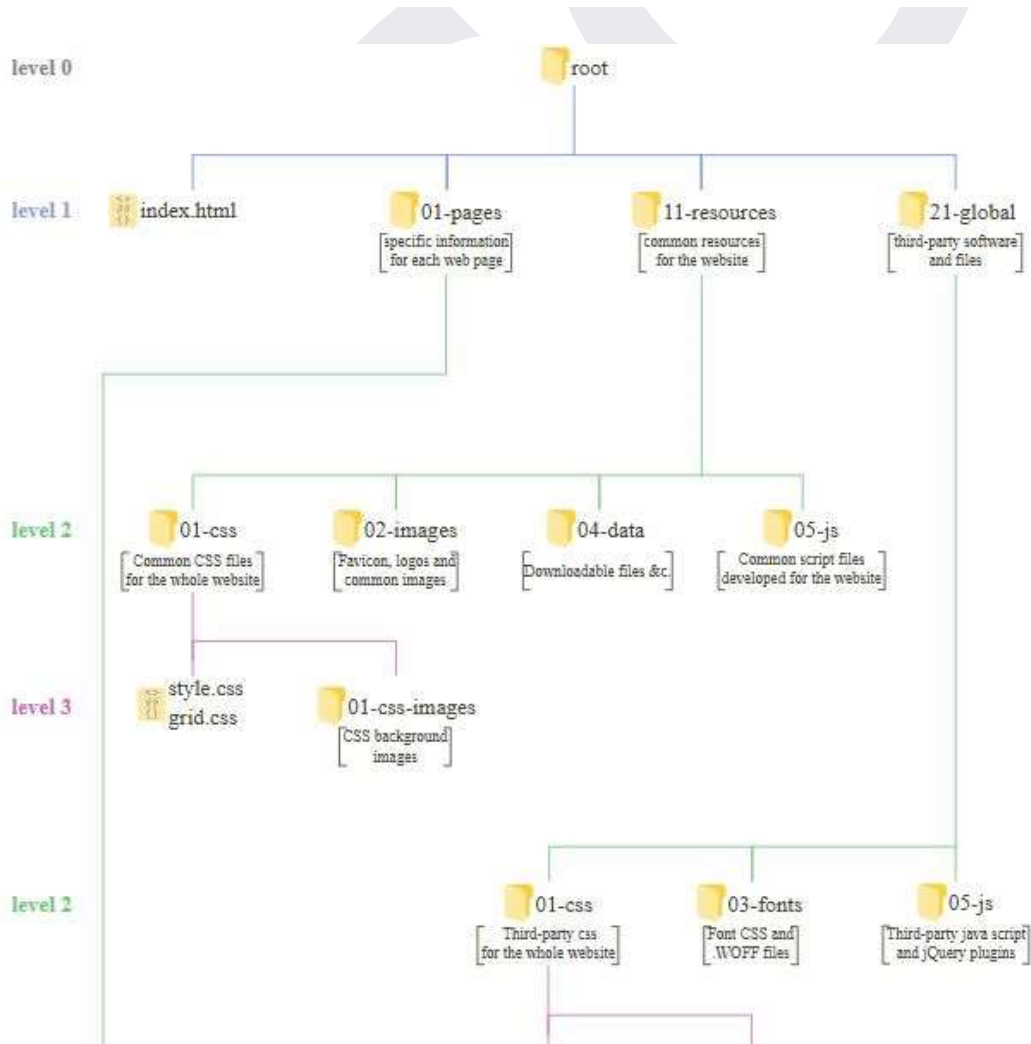


# DSG Support Multi Solution

यह फ़ोल्डर संरचना, संक्षेप में, वेबसाइट है। यह फ़ोल्डर संरचना और इसकी सामग्री ही है जो अंततः किसी सर्वर पर अपलोड की जाएगी और फिर इसे उन लाखों वेब उपयोगकर्ताओं द्वारा देखा जाएगा जो ऐसी ही वेबसाइट का इंतज़ार कर रहे हैं।

फ़ोल्डर संरचना की आवश्यकता नहीं है, सभी फ़ाइलों को एक ही फ़ोल्डर में रखना पूरी तरह से संभव है और वेबसाइट बिल्कुल वैसी ही दिखेगी और काम करेगी और उपयोगकर्ता को कभी भी अंतर का पता नहीं चलेगा। हालाँकि, फ़ोल्डर संरचना लागू करने से (विशेष रूप से एक बड़ी वेबसाइट पर - जो मुझे लगता है कि यह बन रही है) बेहतर संगठन होता है - अगर साइट पर अनुसरण करने में आसान संरचना है तो आम तौर पर चीजों को ढूँढना आसान होता है (और फिर मेरी संरचना है - बस मजाक कर रहा हूँ)।

मोटे तौर पर मैंने वेबसाइट की संरचना इस प्रकार की है



# DSG Support Multi Solution

## नामकरण परंपराएँ

वेबसाइट पर फ़ाइलों और फ़ोल्डरों के नामकरण के लिए कुछ परंपराएँ हैं और इनका व्यापक रूप से पालन किया जाता है (तर्क के भीतर, आप हमेशा वेब पर अपवाद पा सकते हैं)। ये परंपराएँ हैं:

1. फ़ाइलों और फ़ोल्डरों का नामकरण करते समय हमेशा लोअर केस का उपयोग करें
2. फ़ाइल और फ़ोल्डर नामों में रिक्त स्थान का उपयोग न करें (इसके बजाय डैश का उपयोग करें)
3. केवल वर्ण [az], संख्या [0-9] और डैश/हाइफ़न [-] का उपयोग करें

मेरा मानना है कि ये परंपराएँ इस तथ्य से उपजी हैं कि बहुत सारे वेब सर्वर (वे चीजें जो वेबसाइट को होस्ट करती हैं) UNIX पर आधारित हैं; जिसका अर्थ है कि फ़ाइलें और फ़ोल्डर नाम केस सेंसिटिव हैं।

विंडोज में `Index.HTML` नामक फ़ाइल और `index.html` नामक फ़ाइल (या वास्तव में `InDeX.hTmL` या इसके किसी भी संयोजन) के बीच कोई अंतर नहीं है। यदि आप `Index.html` को ऐसे फ़ोल्डर में कॉपी करने का प्रयास करते हैं जिसमें `index.html` है तो यह बाद वाले को पहले वाले से बदल देगा।

दूसरी ओर, UNIX `Index.HTML` और `index.html` को दो पूर्णतः अलग-अलग फाइलों के रूप में मानेगा और वे एक ही निर्देशिका में खुशी-खुशी सह-अस्तित्व में रह सकते हैं। इसलिए भ्रम से बचने के लिए सभी फ़ाइलें और फ़ोल्डर्स लोअरकेस<sup>†1</sup> में हैं।

अगला स्थान रिक्त स्थान है। रिक्त स्थान के साथ समस्या यह है कि इन्हें अलग-अलग वेब सर्वर द्वारा अलग-अलग तरीके से व्यवहार (एनकोड) किया जा सकता है (यहां तक कि विंडोज कमांड लाइन भी फ़ाइल नामों में रिक्त स्थान पसंद नहीं करती है, इन्हें उद्धरण चिह्नों से घिरा होना चाहिए)। सबसे अच्छी बात यह है कि फ़ाइल और फ़ोल्डर नामों में रिक्त स्थान से बचें।

रिक्त स्थान को बदलने के लिए डैश का उपयोग करें, जैसे `01-00-introduction.html`, न कि `01 00 Introduction.html`। फ़ाइल नामों में अंडरस्कोर होना पूरी तरह से संभव है - लेकिन ऐसा न करें - यह गलत अभ्यास है और होस्ट नामों में इसका उपयोग करते समय कुछ भ्रम होता है (कम से कम में

# DSG Support Multi Solution

भ्रमित था) (मुझे यकीन नहीं है कि होस्ट नाम क्या है, यह डोमेन नाम से अलग है, लेकिन मुझे नहीं पता कि कैसे)। वैसे भी, यह रिक्त स्थान के स्थान पर डैश (हाइफ़ेन) है।

## HTML Introduction

HTML का मतलब हाइपरटेक्स्ट मार्कअप लैंग्वेज है। यह वेब पर सामग्री बनाने और संरचना करने के लिए उपयोग की जाने वाली मानक भाषा है।

- HTML एक मार्कअप भाषा है, प्रोग्रामिंग भाषा नहीं, जिसका अर्थ है कि यह पाठ को एनोटेट करके यह परिभाषित करती है कि वेब ब्राउज़र द्वारा इसे कैसे संरचित और प्रदर्शित किया जाए।
- यह सभी वेबसाइटों के निर्माण खंड का निर्माण करता है और शैली के लिए CSS और अन्तर्क्रियाशीलता के लिए जावास्क्रिप्ट द्वारा पूरित होता है।

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>My First Webpage</title>
```

```
</head>
```

```
<body>
```

```
<h1>Welcome to My Webpage</h1>
```

```
<p>This is my first paragraph of text!</p>
```

```
</body>
```

```
</html>
```

HTML क्यों सीखें?

HTML सीखने के 5 सामान्य कारण यहां दिए गए हैं :

1. **वेबसाइट बनाएँ** : HTML किसी भी वेबसाइट को बनाने के लिए बुनियादी बिल्डिंग ब्लॉक है। HTML सीखना आपको वेब डेवलपमेंट में करियर बनाने में मदद कर सकता है।
2. **सामग्री अनुकूलित करें** : आपको अपनी आवश्यकताओं के अनुरूप वेब पेज, ईमेल या टेम्पलेट्स को संपादित या संशोधित करने की अनुमति देता है।
3. **समझें कि वेब कैसे काम करता है** : यह आपको यह समझने में मदद करता है कि इंटरनेट कैसे काम करता है और वेब पेज कैसे संरचित होते हैं।
4. **रोजगार के अवसर** : श्रम सांख्यिकी ब्यूरो के अनुसार 2022-2032 के बीच वेब डेवलपर्स के लिए रोजगार 16% बढ़ेगा, जो सभी व्यवसायों में औसत से बहुत तेज है।

# DSG Support Multi Solution

5. आसानी से सीखें : HTML शुरुआती लोगों के लिए अनुकूल है, जो इसे कोडिंग और प्रौद्योगिकी की दुनिया में पहला कदम बनाने में मदद करता है

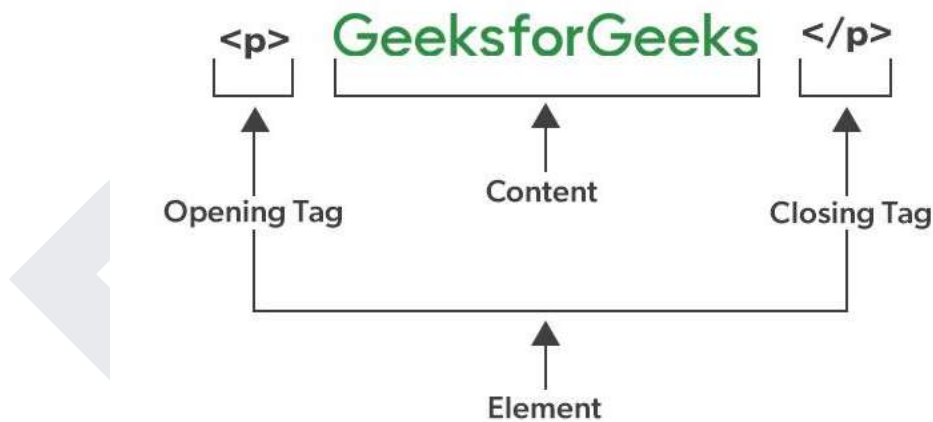
## HTML की विशेषताएं

- इसे सीखना आसान है और इसका प्रयोग भी आसान है।
- यह प्लेटफॉर्म-स्वतंत्र है।
- चित्र, वीडियो और ऑडियो को वेब पेज पर जोड़ा जा सकता है।
- हाइपरटेक्स्ट को पाठ में जोड़ा जा सकता है।
- यह एक मार्कअप भाषा है.

## HTML तत्व और HTML टैग

HTML एलिमेंट और HTML टैग्स संबंधित हैं लेकिन अलग-अलग हैं। **HTML एलिमेंट** एक संपूर्ण संरचना है, जिसमें ओपनिंग टैग, कंटेंट (यदि कोई हो) और क्लोजिंग टैग (यदि लागू हो) शामिल है।

दूसरी ओर, **A टैग** कोण कोष्ठक ( ) में संलग्न वास्तविक कीवर्ड या नाम है < > जो ब्राउज़र को बताता है कि किस प्रकार की सामग्री की अपेक्षा की जाए।

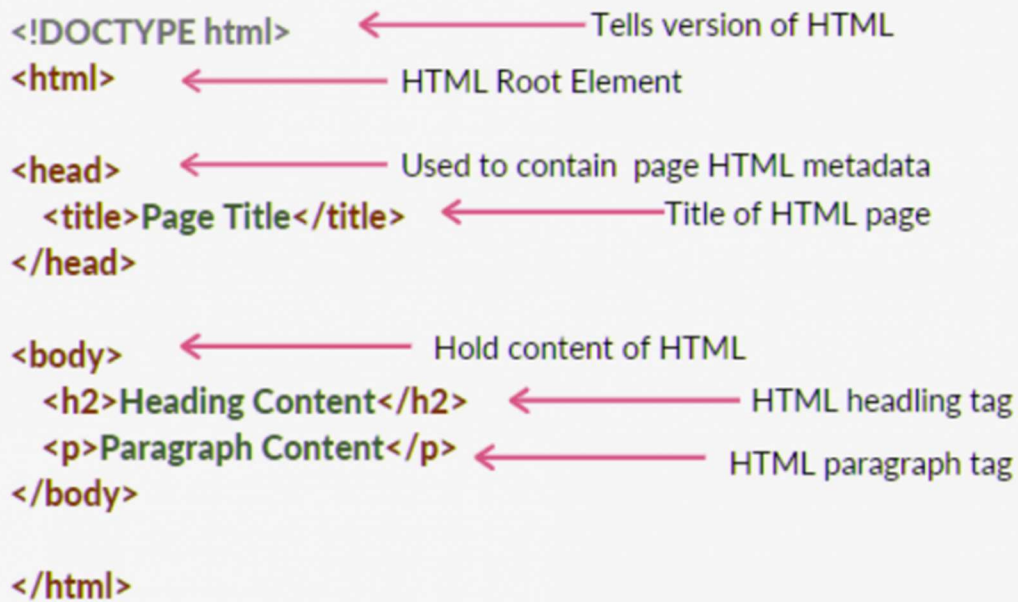


## HTML पृष्ठ संरचना

HTML पेज की मूल संरचना नीचे दर्शाई गई है। इसमें आवश्यक बिल्डिंग-ब्लॉक तत्व (यानी डॉकटाइप घोषणा, HTML, हेड, शीर्षक और बॉडी तत्व) शामिल हैं, जिन पर सभी वेब पेज बनाए जाते हैं।

# DSG Support Multi Solution

## HTML Page Structure



- [<!DOCTYPE html>](#) – यह दस्तावेज़ प्रकार की घोषणा है, टैग नहीं। यह घोषणा करता है कि दस्तावेज़ एक HTML5 दस्तावेज़ है।
- [<html>](#) - इसे HTML रूट एलिमेंट कहा जाता है। अन्य सभी एलिमेंट इसके अंदर समाहित होते हैं।
- [<head>](#) – हेड टैग में वेबपेज के लिए “पर्दे के पीछे” तत्व शामिल होते हैं। हेड के अंदर के तत्व वेबपेज के फ्रंट एंड पर दिखाई नहीं देते हैं। <head> के अंदर के आम तत्वों में ये शामिल हैं:
  - [<title>](#) : ब्राउज़र टैब पर प्रदर्शित शीर्षक को परिभाषित करता है।
  - [<meta>](#) : वर्ण सेट या व्यूपोर्ट सेटिंग्स जैसी जानकारी प्रदान करता है।
  - [<link>](#) : बाहरी स्टाइलशीट या संसाधनों को लिंक करता है।
  - [<style>](#) : आंतरिक CSS शैलियाँ एम्बेड करता है।
  - [<script>](#) : कार्यक्षमता के लिए जावास्क्रिप्ट एम्बेड करता है।
- [<title>](#) – शीर्षक वह है जो आपके ब्राउज़र के शीर्ष पर प्रदर्शित होता है जब आप किसी वेबसाइट पर जाते हैं और इसमें उस वेबपेज का शीर्षक होता है जिसे आप देख रहे हैं।
- [<h2>](#) – <h2> टैग एक द्वितीय-स्तरीय शीर्षक टैग है।
- [<p>](#) – <p> टैग पाठ के एक पैराग्राफ का प्रतिनिधित्व करता है।
- [<body>](#) - बॉडी टैग का उपयोग वेबपेज की सभी दृश्यमान सामग्री को संलग्न करने के लिए किया जाता है। दूसरे शब्दों में, बॉडी सामग्री वह है जो ब्राउज़र फ्रंट एंड पर दिखाएगा।

# DSG Support Multi Solution

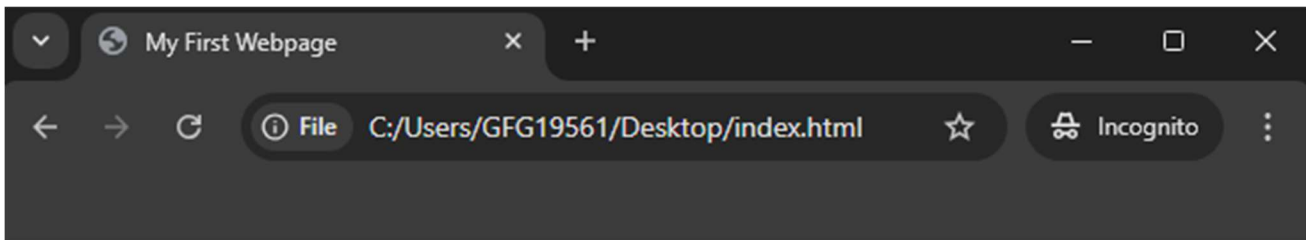
HTML टेक्स्ट एडिटर का उपयोग करके HTML दस्तावेज़ बनाया जा सकता है। “.html” या “.htm” एक्सटेंशन का उपयोग करके टेक्स्ट फ़ाइल को सेव करें। HTML दस्तावेज़ के रूप में सहेजे जाने के बाद, फ़ाइल को ब्राउज़र में वेबपेज के रूप में खोला जा सकता है।

*नोट: बेसिक/बिल्ट-इन टेक्स्ट एडिटर नोटपैड (विंडोज) और टेक्स्टएडिट (मैकओएस) हैं। अन्य उन्नत टेक्स्ट एडिटर में सबलाइम टेक्स्ट, विजुअल स्टूडियो कोड, फ़ोला आदि शामिल हैं।*

## वेब ब्राउज़र

अन्य प्रोग्रामिंग भाषाओं के विपरीत, HTML कम्पाइलर पर आउटपुट नहीं दिखाता है। वेब ब्राउज़र HTML कोड के परिणाम दिखाते हैं।

यह HTML फ़ाइलों को पढ़ता है और HTML टैग की मदद से यह निर्धारित करता है कि सामग्री को कैसे दिखाया जाए। किसी भी वेब ब्राउज़र ( **Google, Safari, Mozilla Firefox** , आदि) का उपयोग **.HTML फ़ाइल को खोलने और परिणाम देखने के लिए** किया जा सकता है।



## Welcome to My Webpage

This is my first paragraph of text!

*index.html*

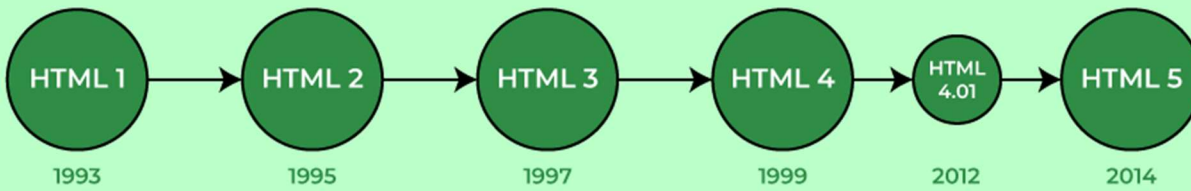
## HTML इतिहास

वर्तमान में हम HTML5 का उपयोग कर रहे हैं , जो HTML का नवीनतम और सबसे उन्नत संस्करण है। HTML को सर्वप्रथम **टिम बर्नर्स-ली** द्वारा 1991 में वेब पर दस्तावेजों को साझा करने और संरचना करने के तरीके के रूप में बनाया गया था।

सबसे पहला संस्करण **HTML 1.0** था , जो एक बुनियादी और सीमित संस्करण था। हालाँकि, पहला मानकीकृत संस्करण, **HTML 2.0** , 1995 में प्रकाशित हुआ था, जिसने वेब विकास की नींव रखी जैसा कि हम आज जानते हैं

# DSG Support Multi Solution

## HTML Released Year



## HTML के लाभ

- HTML का उपयोग वेबसाइट बनाने के लिए किया जाता है।
- यह सभी ब्राउज़रों द्वारा समर्थित है।
- इसे अन्य भाषाओं जैसे [सीएसएस](#) , [जावास्क्रिप्ट](#) आदि के साथ एकीकृत किया जा सकता है।

## HTML के नुकसान

- HTML केवल स्थिर वेब पेज बना सकता है। गतिशील वेब पेज और तर्क के लिए, हमें [जावास्क्रिप्ट](#) की आवश्यकता है ।
- एक सरल वेब पेज बनाने के लिए बड़ी मात्रा में कोड लिखना पड़ता है।

## HTML Form Tag

<form> टैग का उपयोग उपयोगकर्ता इनपुट के लिए [HTML](#) फॉर्म बनाने के लिए किया जाता है। यह टेक्स्ट फ़ील्ड, चेकबॉक्स और बटन जैसे विभिन्न इनपुट तत्वों के लिए एक कंटेनर के रूप में कार्य करता है, जिससे सर्वर पर सबमिशन के लिए डेटा का संग्रह या क्लाइंट-साइड स्क्रिप्ट के माध्यम से प्रोसेसिंग करना संभव हो जाता है।

```
<html>
<body>
  <form action="/submit" method="POST">
    <h2>User Information</h2>
    <label for="fname">First Name</label>
    <input type="text"
      id="fname"
      name="fname"
      placeholder="Enter your first name" required><br><br>
```



# DSG Support Multi Solution

```
<label for="lname">Last Name</label>
<input type="text"
      id="lname"
      name="lname"
      placeholder="Enter your last name" required><br><br>

<input type="submit"
      value="Submit">

</form>
</body>
</html>
```

सामान्य फॉर्म तत्व

- [<इनपुट>](#)
- [<बटन>](#)
- [<चुनें>](#)
- [<टेक्स्टएरिया>](#)
- [<लेबल>](#)
- [<फील्डसेट>](#)
- [<किंवदंती>](#)
- [<डेटा सूची>](#)
- [<आउटपुट>](#)

## HTML Table Tag

HTML टेबल्स आपको वेब पेज पर डेटा को पंक्तियों और स्तंभों में व्यवस्थित करने की अनुमति देती है, जिससे शेड्यूल, सांख्यिकी या अन्य संरचित डेटा जैसी जानकारी को स्पष्ट प्रारूप में प्रदर्शित करना आसान हो जाता है।

- **<tr>** तालिका पंक्तियाँ परिभाषित करने के लिए,
- **<th>** तालिका शीर्षलेखों के लिए, और
- **<td>** तालिका डेटा कक्षों के लिए

प्रत्येक <tr> एक पंक्ति का प्रतिनिधित्व करता है, और प्रत्येक पंक्ति के भीतर, <th> या <td> टैग उस पंक्ति के कक्षों का प्रतिनिधित्व करते हैं, जिनमें पाठ, चित्र, सूचियाँ या यहाँ तक कि कोई अन्य तालिका भी हो सकती है।

# DSG Support Multi Solution

```
<!-- index.html -->
<!DOCTYPE html>
<html>

<body>
  <table>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Age</th>
    </tr>
    <tr>
      <td>Priya</td>
      <td>Sharma</td>
      <td>24</td>
    </tr>
    <tr>
      <td>Arun</td>
      <td>Singh</td>
      <td>32</td>
    </tr>
    <tr>
      <td>Sam</td>
      <td>Watson</td>
      <td>41</td>
    </tr>
  </table>
</body>

</html>
```

# DSG Support Multi Solution

आउटपुट:

Firstname	Lastname	Age
Priya	Sharma	24
Arun	Singh	32
Sam	Watson	41

- `<table>`: यह टैग टेबल की शुरुआत करता है। आरंभिक `<table>` और समापन `</table>` टैग के बीच की सभी चीज़ें मिलकर टेबल बनाती हैं।
- `<tr>`: इसका मतलब है “टेबल रो”। प्रत्येक `<tr>` टैग टेबल में एक पंक्ति को परिभाषित करता है।
- `<th>`: इसका मतलब है “टेबल हेडर”। इसका इस्तेमाल कॉलम के हेडर के लिए किया जाता है। इस मामले में, “**प्रथम नाम**”, “**अंतिम नाम**”, और “**आयु**” हेडर हैं। `<th>` टैग में टेक्स्ट आमतौर पर डिफॉल्ट रूप से बोल्ड और केंद्रित होता है।
- `<td>`: इसका मतलब है “टेबल डेटा”। इस टैग का इस्तेमाल हर कॉलम के नीचे मौजूद वास्तविक डेटा सेल के लिए किया जाता है। उदाहरण के लिए, “**प्रिया**” “**फर्स्टनेम**” हेडर के नीचे का डेटा है, “**शर्मा**” “**लास्टनेम**” के नीचे और “**24**” “**आयु**” के नीचे का डेटा है।
- पहले `<tr>` में तीन `<th>` तत्व हैं, जो स्तंभ शीर्षक निर्धारित करते हैं।
- बाद के `<tr>` टैग में प्रत्येक में तीन `<td>` तत्व होते हैं, जो तालिका में सूचीबद्ध प्रत्येक व्यक्ति के डेटा का प्रतिनिधित्व करते हैं।

ags used in HTML Tables

HTML Tags	Descriptions
<a href="#"><u>&lt;table&gt;</u></a>	Defines the structure for organizing data in rows and columns within a web page.
<a href="#"><u>&lt;tr&gt;</u></a>	Represents a <b>row</b> within an HTML table, containing individual cells.
<a href="#"><u>&lt;th&gt;</u></a>	Shows a table <b>header</b> cell that typically holds titles or headings.

# DSG Support Multi Solution

HTML Tags	Descriptions
<a href="#"><u>&lt;td&gt;</u></a>	Represents a standard <b>data</b> cell, holding content or data.
<a href="#"><u>&lt;caption&gt;</u></a>	Provides a title or description for the entire table.
<a href="#"><u>&lt;thead&gt;</u></a>	Defines the header section of a table, often containing column labels.
<a href="#"><u>&lt;tbody&gt;</u></a>	Represents the main content area of a table, separating it from the header or footer.
<a href="#"><u>&lt;tfoot&gt;</u></a>	Specifies the footer section of a table, typically holding summaries or totals.
<a href="#"><u>&lt;col&gt;</u></a>	Defines attributes for table <b>columns</b> that can be applied to multiple columns at once.
<a href="#"><u>&lt;colgroup&gt;</u></a>	Groups together a <b>set of columns</b> in a table to which you can apply formatting or properties collectively.

## Frame Tag

HTML `<frame>` टैग का उपयोग वेब ब्राउज़र विंडो को कई खंडों में विभाजित करने के लिए किया जाता है, जिनमें से प्रत्येक स्वतंत्र रूप से सामग्री लोड करने में सक्षम होता है। यह फ्रेमसेट टैग के भीतर फ्रेम के संग्रह का उपयोग करके प्राप्त किया जाता है।

**नोट:** `<frame>` टैग HTML 5 में अप्रचलित है।

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<frameset rows="20%, 60%, 20%">
```

```
<frame name="top"
```

# DSG Support Multi Solution

```
src="./attr1.html" />
<frame name="main"
  src="./gradient.html" />
<frame name="bottom"
  src="./colLast.html" />
<noframes>
  <p>The browser you are using does not support frames.</p>
</noframes>
</frameset>
</body>

</html>
```

उपरोक्त उदाहरण में मूल रूप से फ्रेमसेट टैग की पंक्ति विशेषता का उपयोग करके तीन क्षैतिज फ्रेम अर्थात् शीर्ष, मध्य और नीचे बनाने के लिए उपयोग किया गया है और नोफ्रेम टैग का उपयोग उन ब्राउज़र के लिए किया जाता है जो नोफ्रेम का समर्थन नहीं करते हैं।

वाक्यविन्यास

```
<frameset cols='50%, 50%'>
  <frame src='page1.html'>
  <frame src='page2.html'>
</frameset>
```

## HTML फ्रेम टैग विशेषताएँ

### HTML <फ्रेम> नाम विशेषता

इस विशेषता का उपयोग फ्रेम को नाम देने के लिए किया जाता है। यह एक फ्रेम को दूसरे से अलग करता है। इसका उपयोग यह इंगित करने के लिए भी किया जाता है कि दस्तावेज़ को किस फ्रेम में लोड किया जाना चाहिए।

### उदाहरण

```
<फ्रेम नाम = "शीर्ष" src = "C:/Users/dharam/Desktop/attr1.png" />
<फ्रेम नाम = "मुख्य" src = "C:/Users/dharam/Desktop/gradient3.png" />
<फ्रेम नाम = "नीचे" src = "C:/Users/dharam/Desktop/col_last.png" />
```

यहां हम तीन फ्रेम का उपयोग करते हैं जिनके नाम बाएं, मध्य और दाएं हैं।

### [HTML <frame> src विशेषता](#)

फ्रेम टैग में यह विशेषता मूल रूप से उस स्रोत फ़ाइल को परिभाषित करने के लिए उपयोग की जाती है जिसे फ्रेम में लोड किया जाना चाहिए। src का मान कोई भी URL हो सकता है।

# DSG Support Multi Solution

## उदाहरण

```
<फ्रेम नाम = "बाएं" src = "/html/left.htm" />
```

उपरोक्त उदाहरण में फ्रेम का नाम left है और स्रोत फ़ाइल फ्रेम में “/html/left.htm” से लोड की जाएगी।

## HTML <फ्रेम> मार्जिनविड्थ विशेषता

फ्रेम टैग में इस विशेषता का उपयोग बाएँ और दाएँ फ्रेम की सीमा और सामग्री के बीच पिक्सेल में रिक्त स्थान की चौड़ाई निर्दिष्ट करने के लिए किया जाता है।

## उदाहरण

```
<फ्रेम मार्जिनविड्थ="20">
```

## HTML <फ्रेम> मार्जिनहाइट विशेषता

फ्रेम टैग में इस विशेषता का उपयोग ऊपरी और निचले फ्रेम की सीमा और सामग्री के बीच पिक्सेल में रिक्त स्थान की ऊंचाई निर्दिष्ट करने के लिए किया जाता है।

## उदाहरण

```
<फ्रेम मार्जिनहाइट="20">
```

## HTML <फ्रेम> स्क्रॉलबार विशेषता

फ्रेम में स्क्रॉल बार की उपस्थिति को नियंत्रित करने के लिए फ्रेम टैग में स्क्रॉलबार विशेषता का उपयोग करें। इसका उपयोग मूल रूप से स्क्रॉलबार की उपस्थिति को नियंत्रित करने के लिए किया जाता है। इस विशेषता का मान हाँ, नहीं, ऑटो हो सकता है। जहाँ मान नहीं दर्शाता है वहाँ स्क्रॉल बार की कोई उपस्थिति नहीं होगी।

## उदाहरण

```
<फ्रेम स्क्रॉलबार="नहीं">
```

## List Tag

HTML सूची आपको वेब पेजों पर डेटा को क्रमबद्ध या अव्यवस्थित प्रारूप में व्यवस्थित करने की अनुमति देती है ताकि जानकारी को पढ़ना आसान हो और देखने में आकर्षक लगे। HTML सूचियाँ वेब विकास में संरचित, सुलभ सामग्री बनाने के लिए बहुत सहायक हैं।

## HTML सूचियों के प्रकार

HTML में तीन मुख्य प्रकार की सूचियाँ हैं:

1. **अव्यवस्थित सूचियाँ (<ul>):** इन सूचियों का उपयोग उन वस्तुओं के लिए किया जाता है जिन्हें किसी विशिष्ट क्रम में रखने की आवश्यकता नहीं होती है। सूची आइटम आमतौर पर बुलेट से चिह्नित होते हैं।

# DSG Support Multi Solution

2. **क्रमबद्ध सूचियाँ (<ol>):** इन सूचियों का उपयोग तब किया जाता है जब आइटम का क्रम महत्वपूर्ण होता है। क्रमबद्ध सूची में प्रत्येक आइटम को आम तौर पर संख्याओं या अक्षरों से चिह्नित किया जाता है।
3. **विवरण सूची (<dl>):** इन सूचियों का उपयोग शब्दों और उनके संगत विवरणों को शामिल करने के लिए किया जाता है।

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>Welcome To GeeksforGeeks Learning</h2>
```

```
<h5>List of available courses</h5>
```

```
<ul>
```

```
<li>Data Structures & Algorithm</li>
```

```
<li>Web Technology</li>
```

```
<li>Aptitude & Logical Reasoning</li>
```

```
<li>Programming Languages</li>
```

```
</ul>
```

```
<h5>Data Structures topics</h5>
```

```
<ol>
```

```
<li>Array</li>
```

```
<li>Linked List</li>
```

```
<li>Stacks</li>
```

```
<li>Queues</li>
```

```
<li>Trees</li>
```

```
<li>Graphs</li>
```

```
</ol>
```

```
</body>
```

```
</html>
```

आउटपुट:



# DSG Support Multi Solution

## Welcome To GeeksforGeeks Learning

### List of available courses

- Data Structures & Algorithm
- Web Technology
- Apptitude & Logical Reasoning
- Programming Languages

### Data Structures topics

1. Array
2. Linked List
3. Stacks
4. Queues
5. Trees
6. Graphs

Tag	Description
<a href="#">&lt;ul&gt;</a>	Defines an unordered list.
<a href="#">&lt;ol&gt;</a>	Defines an ordered list.
<a href="#">&lt;li&gt;</a>	Defines a list item.
<a href="#">&lt;dl&gt;</a>	Defines a description list.
<a href="#">&lt;dt&gt;</a>	Defines a term in a description list.
<a href="#">&lt;dd&gt;</a>	Details the term in a description list.

## HTML Image Tag

HTML <img> टैग का उपयोग वेब पेज में इमेज एम्बेड करने के लिए किया जाता है। यह एक खाली या स्व-समापन टैग है, जिसका अर्थ है कि इसमें कोई समापन टैग नहीं है। यह विभिन्न स्रोतों से इमेज प्रदर्शित करने की अनुमति देता है, जैसे कि किसी वेबसाइट पर फ़ाइलें या अन्य वेबसाइटों के URL।

```
<html>  
  <head></head>  
<body style="text-align: center;">  
  <h3>GeeksforGeeks logo</h3>
```

# DSG Support Multi Solution

```
<img src=
```

```
"https://media.geeksforgeeks.org/wpcontent/uploads/geeksforgeeks-13.png"
```

```
width="420" height="100"
```

```
alt="Geeksforgeeks.org">
```

```
</body>
```

```
</html>
```

## Syntax:-

```

```

## Introduction JavaScript

जावास्क्रिप्ट एक बहुमुखी, गतिशील रूप से टाइप की गई प्रोग्रामिंग भाषा है जिसका उपयोग इंटरैक्टिव वेब अनुप्रयोगों के लिए किया जाता है, जो क्लाइंट-साइड और सर्वर-साइड विकास दोनों का समर्थन करता है, और [HTML](#) , [CSS](#) और एक समृद्ध मानक लाइब्रेरी के साथ सहजता से एकीकृत होता है।

- जावास्क्रिप्ट एक [एकल-थ्रेडेड](#) भाषा है जो एक समय में एक ही कार्य निष्पादित करती है।
- यह एक [इंटरप्रेटेड भाषा](#) है , जिसका अर्थ है कि यह कोड को लाइन दर लाइन निष्पादित करती है।
- जावास्क्रिप्ट में वेरिएबल का डेटा प्रकार रन-टाइम पर तय किया जाता है, इसीलिए इसे डायनामिकली टाइपड कहा जाता है।

ब्राउज़र कंसोल में “हैलो, वर्ल्ड!” प्रोग्राम

“हैलो, वर्ल्ड!” प्रोग्राम किसी भी प्रोग्रामिंग भाषा के साथ शुरुआत करने का सबसे सरल तरीका है। यहाँ बताया गया है कि आप जावास्क्रिप्ट का उपयोग करके इसे कैसे लिख सकते हैं।

```
<html>
```

```
<head></head>
```

```
<body>
```

```
<h1>Check the console for the message!</h1>
```

```
<script>
```

```
// This is our first JavaScript program
```

```
console.log("Hello, World!");
```

```
</script>
```

```
</body>
```

```
</html>
```

# DSG Support Multi Solution

सर्वर कंसोल में “हैलो वर्ल्ड” प्रोग्राम

हम “Hello World” प्रोग्राम को HTML में एम्बेड किए बिना सीधे कंसोल टर्मिनल में प्रिंट कर सकते हैं। एक index.js फ़ाइल बनाएँ और उसमें कोड जोड़ें।

```
/*
```

**This is a multi-line comment.**

**It can span several lines.**

```
*/
```

```
console.log("Hello, World!"); // Prints Hello, World! to the console
```

इस उदाहरण में

- **कंसोल.लॉग():** कंसोल.लॉग() विधि का उपयोग ब्राउज़र के डेवलपर कंसोल पर संदेश प्रिंट करने के लिए किया जाता है। संदेश “हैलो, वर्ल्ड!” प्रदर्शित देखने के लिए कंसोल खोलें (आमतौर पर F12 या Ctrl + Shift + J के साथ)।
- **संहिता में टिप्पणियाँ :**
  - **बहु-पंक्ति टिप्पणी:** /\* \*/ सिंटैक्स का उपयोग कई पंक्तियों में टिप्पणी लिखने के लिए किया जाता है।
  - **एकल-पंक्ति टिप्पणी:** // सिंटैक्स का उपयोग छोटी, इनलाइन टिप्पणियों के लिए किया जाता है, जैसे कंसोल.लॉग फ़ंक्शन को समझाने वाली टिप्पणी।

जावास्क्रिप्ट की मुख्य विशेषताएं

- **क्लाइंट-साइड स्क्रिप्टिंग:** जावास्क्रिप्ट उपयोगकर्ता के [ब्राउज़र पर चलता है](#), इसलिए [सर्वर](#) के साथ संवाद किए बिना इसका प्रतिक्रिया समय तेज़ होता है।
- **बहुमुखी:** जावास्क्रिप्ट का उपयोग सरल गणनाओं से लेकर जटिल सर्वर-साइड अनुप्रयोगों तक, कार्यों की एक विस्तृत श्रृंखला के लिए किया जा सकता है।
- **घटना-संचालित:** जावास्क्रिप्ट वास्तविक समय में उपयोगकर्ता की क्रियाओं (क्लिक, कीस्ट्रॉक्स) पर प्रतिक्रिया दे सकता है।
- **एसिंक्रोनस:** जावास्क्रिप्ट उपयोगकर्ता इंटरफ़ेस को स्थिर किए बिना [सर्वर से डेटा लाने जैसे कार्यों को संभाल सकता है](#)।
- **समृद्ध पारिस्थितिकी तंत्र:** जावास्क्रिप्ट पर निर्मित कई [लाइब्रेरी](#) और [फ्रेमवर्क](#) हैं , जैसे कि [रिएक्ट](#) , [एंगुलर](#) और [व्यू.जेएस](#) , जो विकास को तेज और अधिक कुशल बनाते हैं।

# DSG Support Multi Solution

## जावास्क्रिप्ट की क्लाइंट साइड और सर्वर साइड प्रकृति

- **क्लाइंट-साइड** : इसमें ब्राउज़र और उसके DOM को नियंत्रित करना, क्लिक और फॉर्म इनपुट जैसे उपयोगकर्ता ईवेंट को संभालना शामिल है। AngularJS, ReactJS और VueJS जैसी लाइब्रेरी का आमतौर पर इस्तेमाल किया जाता है।
- **सर्वर-साइड** : इसमें डेटाबेस के साथ इंटरैक्ट करना, फाइलों में हेरफेर करना और प्रतिक्रियाएँ उत्पन्न करना शामिल है। Node.js और Express.js जैसे फ्रेमवर्क के साथ, जावास्क्रिप्ट का उपयोग सर्वर साइड पर भी व्यापक रूप से किया जाता है।
- **अनिवार्य प्रोग्रामिंग** : यह इस बात पर केंद्रित है कि कार्यों को कैसे निष्पादित किया जाए, गणना के प्रवाह को नियंत्रित किया जाए। इसमें प्रक्रियात्मक और ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग जैसे दृष्टिकोण शामिल हैं, जो अक्सर क्रियाओं को संभालने के लिए async/await जैसी संरचनाओं का उपयोग करते हैं।
- **घोषणात्मक प्रोग्रामिंग** : यह इस बात पर ध्यान केंद्रित करता है कि क्या किया जाना चाहिए, न कि यह कि इसे कैसे किया जाना चाहिए। यह वांछित परिणाम का वर्णन करने पर जोर देता है, जैसे कि एरो फंक्शन के साथ, इसे प्राप्त करने के चरणों का विवरण दिए बिना।

## जावास्क्रिप्ट की मुख्य विशेषताएं

- **क्लाइंट-साइड स्क्रिप्टिंग**: जावास्क्रिप्ट उपयोगकर्ता के [ब्राउज़र पर चलता है](#), इसलिए [सर्वर](#) के साथ संवाद किए बिना इसका प्रतिक्रिया समय तेज़ होता है।
- **बहुमुखी**: जावास्क्रिप्ट का उपयोग सरल गणनाओं से लेकर जटिल सर्वर-साइड अनुप्रयोगों तक, कार्यों की एक विस्तृत श्रृंखला के लिए किया जा सकता है।
- **घटना-संचालित**: जावास्क्रिप्ट वास्तविक समय में उपयोगकर्ता की क्रियाओं (क्लिक, कीस्ट्रॉक्स) पर प्रतिक्रिया दे सकता है।
- **एसिंक्रोनस**: जावास्क्रिप्ट उपयोगकर्ता इंटरफ़ेस को स्थिर किए बिना [सर्वर से डेटा लाने जैसे कार्यों को संभाल सकता है](#)।
- **समृद्ध पारिस्थितिकी तंत्र**: जावास्क्रिप्ट पर निर्मित कई [लाइब्रेरी](#) और [फ्रेमवर्क](#) हैं, जैसे कि [रिएक्ट](#), [एंगुलर](#) और [व्यू.जेएस](#), जो विकास को तेज और अधिक कुशल बनाते हैं।

## जावास्क्रिप्ट की क्लाइंट साइड और सर्वर साइड प्रकृति

- **क्लाइंट-साइड** : इसमें ब्राउज़र और उसके DOM को नियंत्रित करना, क्लिक और फॉर्म इनपुट जैसे उपयोगकर्ता ईवेंट को संभालना शामिल है। AngularJS, ReactJS और VueJS जैसी लाइब्रेरी का आमतौर पर इस्तेमाल किया जाता है।

# DSG Support Multi Solution

- **सर्वर-साइड** : इसमें डेटाबेस के साथ इंटरैक्ट करना, फ़ाइलों में हेरफेर करना और प्रतिक्रियाएँ उत्पन्न करना शामिल है। Node.js और Express.js जैसे फ्रेमवर्क के साथ, जावास्क्रिप्ट का उपयोग सर्वर साइड पर भी व्यापक रूप से किया जाता है।
- **अनिवार्य प्रोग्रामिंग** : यह इस बात पर केंद्रित है कि कार्यों को कैसे निष्पादित किया जाए, गणना के प्रवाह को नियंत्रित किया जाए। इसमें प्रक्रियात्मक और ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग जैसे दृष्टिकोण शामिल हैं, जो अक्सर क्रियाओं को संभालने के लिए `async/await` जैसी संरचनाओं का उपयोग करते हैं।
- **घोषणात्मक प्रोग्रामिंग** : यह इस बात पर ध्यान केंद्रित करता है कि क्या किया जाना चाहिए, न कि यह कि इसे कैसे किया जाना चाहिए। यह वांछित परिणाम का वर्णन करने पर जोर देता है, जैसे कि एरो फ़ंक्शन के साथ, इसे प्राप्त करने के चरणों का विवरण दिए बिना।

## JavaScript Versions

Let's take a look at the different versions of ECMAScript, their release years, and the key features they introduced



# DSG Support Multi Solution

Version	Name	Release Year	Features
ES1	ECMAScript 1	1997	Initial Release
ES2	ECMAScript 2	1998	Minor Editorial Changes
ES3	ECMAScript 3	1999	<p><b>Added:</b></p> <ul style="list-style-type: none"> <li>• <a href="#">Regular Expression</a></li> <li>• <a href="#">try/catch</a></li> <li>• <a href="#">Exception Handling</a></li> <li>• <a href="#">switch case</a> and <a href="#">do-while</a></li> </ul>
ES4	ECMAScript 4		Abandoned due to conflicts
ES5	ECMAScript 5	2009	<p><b>Added:</b></p> <ul style="list-style-type: none"> <li>• <a href="#">JavaScript "strict mode"</a></li> <li>• <a href="#">JSON support</a></li> <li>• <a href="#">JS getters and setters</a></li> </ul>
<a href="#">ES6</a>	<a href="#">ECMAScript 2015</a>	2015	<p><b>Added:</b></p> <ul style="list-style-type: none"> <li>• <a href="#">let and const</a></li> <li>• <a href="#">Class declaration</a></li> <li>• <a href="#">import and export</a></li> <li>• <a href="#">for..of loop</a></li> <li>• <a href="#">Arrow functions</a></li> </ul>
ES7	ECMAScript 2016	2016	<p><b>Added:</b></p> <ul style="list-style-type: none"> <li>• <a href="#">Block scope for variable</a></li> <li>• <a href="#">async/await</a></li> <li>• <a href="#">Array.includes function</a></li> <li>• <a href="#">Exponentiation Operator</a></li> </ul>
ES8	ECMAScript 2017	2017	<p><b>Added:</b></p> <ul style="list-style-type: none"> <li>• <a href="#">Object.values</a></li> <li>• <a href="#">Object.entries</a></li> <li>• <a href="#">Object.getOwnPropertyDescriptors</a></li> </ul>
ES9	ECMAScript 2018	2018	<p><b>Added:</b></p> <ul style="list-style-type: none"> <li>• <a href="#">spread operator</a></li> <li>• <a href="#">rest parameters</a></li> </ul>

# DSG Support Multi Solution

ES10	ECMAScript 2019	2019	<b>Added:</b> <ul style="list-style-type: none"><li>• <a href="#">Array.flat()</a></li><li>• <a href="#">Array.flatMap()</a></li><li>• <a href="#">Array.sort is now stable</a></li></ul>
ES11	ECMAScript 2020	2020	<b>Added:</b> <ul style="list-style-type: none"><li>• <a href="#">BigInt primitive type</a></li><li>• <a href="#">nullish coalescing operator</a></li></ul>
ES12	ECMAScript 2021	2021	<b>Added:</b> <ul style="list-style-type: none"><li>• <a href="#">String.replaceAll() Method</a></li><li>• <a href="#">Promise.any() Method</a></li></ul>
ES13	ECMAScript 2022	2022	<b>Added:</b> <ul style="list-style-type: none"><li>• Top-level await</li><li>• New class elements</li><li>• Static block inside classes</li></ul>
ES14	ECMAScript 2023	2023	<b>Added:</b> <ul style="list-style-type: none"><li>• <a href="#">toSorted method</a></li><li>• <a href="#">toReversed method</a></li><li>• <a href="#">findLast, and findLastIndex methods on Array.prototype and TypedArray.prototype</a></li></ul>

## JavaScript Variable

जावास्क्रिप्ट में वेरिएबल्स को var, let या const का उपयोग करके घोषित किया जा सकता है। जावास्क्रिप्ट गतिशील रूप से टाइप किया गया है, इसलिए [वेरिएबल प्रकारों को स्पष्ट प्रकार परिभाषाओं के बिना रनटाइम पर निर्धारित किया जाता है।](#)

- [जावास्क्रिप्ट var कीवर्ड](#)
- [जावास्क्रिप्ट let कीवर्ड](#)
- [जावास्क्रिप्ट const कीवर्ड](#)

```
var a = 10 // Old style
```

```
let b = 20; // Preferred for non-const
```

```
const c = 30; // Preferred for const (cannot be changed)
```

```
console.log(a);
```

```
console.log(b);
```

```
console.log(c);
```



# DSG Support Multi Solution

## उत्पादन

10  
20  
30

## जावास्क्रिप्ट में चर घोषित करना

### 1. जावास्क्रिप्ट var कीवर्ड

var जावास्क्रिप्ट में एक कीवर्ड है जिसका उपयोग चर घोषित करने के लिए किया जाता है और यह फ़ंक्शन-स्कोप और होइस्टेड है, जिससे पुनर्घोषणा की अनुमति मिलती है लेकिन इससे अप्रत्याशित बग उत्पन्न हो सकते हैं।

```
var a = "Hello Geeks";  
var b = 10;  
console.log(a);  
console.log(b);
```

### 2. जावास्क्रिप्ट let कीवर्ड

let जावास्क्रिप्ट में एक कीवर्ड है जिसका उपयोग चर घोषित करने के लिए किया जाता है और यह ब्लॉक-स्कोप है और शीर्ष पर नहीं चढ़ाया जाता है, परिवर्तनीय चर के लिए उपयुक्त है

```
let a = 12  
let b = "gfg";  
console.log(a);  
console.log(b);
```

### 3. जावास्क्रिप्ट const कीवर्ड

const जावास्क्रिप्ट में एक कीवर्ड है जिसका उपयोग वेरिएबल्स को घोषित करने के लिए किया जाता है और यह ब्लॉक-स्कोप, अपरिवर्तनीय बाइंडिंग है जिसे पुनः असाइन नहीं किया जा सकता है, हालांकि ऑब्जेक्ट्स को अभी भी उत्परिवर्तित किया जा सकता है।

```
const a = 5  
let b = "gfg";  
console.log(a);  
console.log(b);
```

## चरों के नामकरण के नियम

जावास्क्रिप्ट में चरों का नामकरण करते समय, इन नियमों का पालन करें

- चर नाम किसी अक्षर, अंडरस्कोर (`_`) या डॉलर चिह्न (`$`) से शुरू होना चाहिए।
- इसके बाद के अक्षर अक्षर, संख्या, अंडरस्कोर या डॉलर चिह्न हो सकते हैं।
- चर नाम केस-सेंसिटिव होते हैं (उदाहरण के लिए, आयु और आयु अलग-अलग चर हैं)।

# DSG Support Multi Solution

- आरक्षित कीवर्ड (जैसे फ़ंक्शन, क्लास, रिटर्न, आदि) का उपयोग वैरिएबल नाम के रूप में नहीं किया जा सकता।

```
let userName = "Suman"; // Valid
let $price = 100; // Valid
let _temp = 0; // Valid
let 123name = "Ajay"; // Invalid
let function = "gfg"; // Invalid
```

## जावास्क्रिप्ट में वैरिएबल शैडोइंग

**वैरिएबल शैडोइंग** तब होता है जब किसी निश्चित स्कोप (जैसे, कोई फ़ंक्शन या ब्लॉक) के भीतर घोषित वैरिएबल का नाम बाहरी स्कोप में मौजूद वैरिएबल के समान होता है। आंतरिक वैरिएबल अपने स्कोप के भीतर बाहरी वैरिएबल को **ओवरराइड** करता है ।

```
let n = 10; // Global scope

function gfg() {
  let n = 20; // Shadows the global 'n' inside this function
  console.log(n); // Output: 20
}

gfg();
console.log(n); // Output: 10 (global 'n' remains unchanged)
```

## उत्पादन

```
20
10
```

- आंतरिक n अपने दायरे में बाहरी n को छाया देता है।
- बाहरी n अभी भी फ़ंक्शन के बाहर पहुँच योग्य है।

इसके बारे में अधिक पढ़ने के लिए लेख पढ़ें - [जावास्क्रिप्ट में वैरिएबल शैडोइंग](#)

## जावास्क्रिप्ट में चर का दायरा

स्कोप आपके कोड में वैरिएबल की पहुँच को निर्धारित करता है। जावास्क्रिप्ट निम्न प्रकार के स्कोप का समर्थन करता है

### 1. वैश्विक दायरा

किसी भी फ़ंक्शन या ब्लॉक के बाहर घोषित किए गए वैरिएबल वैश्विक रूप से स्कोप किए जाते हैं। जबकि var, let, और const सभी को फ़ंक्शन के बाहर घोषित किए जाने पर वैश्विक स्कोप हो सकता है, उनका व्यवहार अलग-अलग होता है:

- var को ब्राउज़र में विंडो ऑब्जेक्ट में जोड़ा जाता है।

# DSG Support Multi Solution

- let और const विंडो ऑब्जेक्ट से संलग्न नहीं होते, जिससे वे आधुनिक उपयोग के लिए अधिक सुरक्षित हो जाते हैं।

```
var globalVar = "I am global";
let globalLet = "I am also global";
const globalConst = "I am global too";
```

## 2. कार्य क्षेत्र

किसी फ़ंक्शन के अंदर घोषित किए गए वेरिएबल केवल उसी फ़ंक्शन के भीतर ही एक्सेस किए जा सकते हैं। यह var, let और const पर लागू होता है:

```
function test() {
  var localVar = "I am local";
  let localLet = "I am also local";
  const localConst = "I am local too";
}
console.log(localVar); // Error: not defined
```

## 3. ब्लॉक स्कोप

किसी ब्लॉक के अंदर let या const के साथ घोषित किए गए वेरिएबल्स (जैसे, {} के अंदर) ब्लॉक-स्कोपड होते हैं, जिसका अर्थ है कि उन्हें ब्लॉक के बाहर एक्सेस नहीं किया जा सकता है। हालाँकि, var ब्लॉक-स्कोपड नहीं है और ब्लॉक के बाहर लीक हो जाएगा।

```
{
  let blockVar = "I am block-scoped";
  const blockConst = "I am block-scoped too";
}
console.log(blockVar); // Error: not defined
```

जावास्क्रिप्ट में वेरिएबल्स के बारे में रोचक तथ्य

**1. let या const को var से ज़्यादा प्राथमिकता दी जाती है:** शुरुआत में, जावास्क्रिप्ट में सभी वेरिएबल var कीवर्ड का इस्तेमाल करके लिखे गए थे, लेकिन ES6 में let और const कीवर्ड पेश किए गए। var के साथ मुख्य समस्या स्कोपिंग है।

**2. var फ़ंक्शन स्कोपड है:** यदि [फ़ंक्शन](#) के भीतर है तो ब्लॉक के बाहर एक्सेस किया जा सकता है।

```
if (true) {
  var x = 10;
}
```

```
// Accessible outside the block
// because we are in same function
console.log(x);
```

## उत्पादन

# DSG Support Multi Solution

3. **let** और **const** ब्लॉक स्कोपड हैं: समान फ़ंक्शन के अंदर होने पर भी ब्लॉक के बाहर एक्सेस नहीं किया जा सकता

```
if (true) {  
  let y = 20;  
  const z = 30;  
}  
console.log(y, z); // ReferenceError
```

आउटपुट:

हैंगअप (SIGHUP)

```
/home/guest/sandbox/Solution.js:5  
console.log(y, z); // ReferenceError  
      ^
```

4. **var** को उसी दायरे में पुनः घोषित किया जा सकता है, लेकिन **let** और **const** को नहीं।

```
var x = 10;  
var x = 20; // Allowed  
  
let y = 30;  
let y = 40; // SyntaxError  
  
const z = 50;  
const z = 60; // SyntaxError
```

उत्पादन

सिंटैक्स त्रुटि: पहचानकर्ता 'y' पहले ही घोषित किया जा चुका है

5. हम **array** या ऑब्जेक्ट के तत्वों को बदल सकते हैं, भले ही उन्हें **const** के रूप में घोषित किया गया हो।

```
const ob = { a: 10 };  
ob.a = 20; // Allowed  
  
const arr = [10, 20, 30]  
arr[2] = 40  
console.log(arr) // Allowed
```

*/\* TypeError in the below lines*

```
obj = { b: 30 };  
arr = [50, 100] */
```

उत्पादन

```
[ 10, 20, 40 ]
```

**var**, **let**, या **const** का उपयोग कब करें

- यदि मान को नहीं बदला जाना चाहिए तो हम **const** का उपयोग करके चर घोषित करते हैं

# DSG Support Multi Solution

- यदि हम परिवर्तनीय मान चाहते हैं तो हमें let का उपयोग करना चाहिए अन्यथा हम const का उपयोग नहीं कर सकते
- हम var का उपयोग केवल तभी करते हैं जब हम पुराने ब्राउज़र का समर्थन करते हैं।

चरों के दायरे के बारे में अधिक जानने के लिए इस लेख को देखें [जावास्क्रिप्ट में चर के दायरे को समझना](#)  
जावास्क्रिप्ट में let, var, और const कीवर्ड के गुणों की तुलना:

Property	var	let	const
<a href="#">Scope</a>	Function scoped	Block scoped	Block scoped
Updation	Mutable	Mutable	Immutable
Redeclaration	Can be redeclared	Cannot be redeclared	Cannot be redeclared
<a href="#">Hoisting</a>	Hoisted at top	Hoisted at top	Hoisted at top
Origins	Pre ES2015	ES2015(ES6)	ES2015(ES6)
Support	Supported in the old version of Browser	Not supported in the old version of the Browser	Not supported in the old version of the Browser

## Data Type JavaScript

जावास्क्रिप्ट में, प्रत्येक मान का एक डेटा प्रकार होता है, जो इसकी प्रकृति (जैसे, संख्या, स्ट्रिंग, बूलियन) और संचालन को परिभाषित करता है। डेटा प्रकारों को आदिम (जैसे, स्ट्रिंग, संख्या) और गैर-आदिम (जैसे, ऑब्जेक्ट, एरे) में वर्गीकृत किया जाता है।

### आदिम डेटा प्रकार

#### 1. संख्या

जावास्क्रिप्ट में Number डेटा प्रकार में पूर्णांक और फ्लोटिंग-पॉइंट संख्याएँ दोनों शामिल हैं। Infinity, -Infinity और NaN जैसे विशेष मान क्रमशः अनंत मान और कम्प्यूटेशनल त्रुटियाँ दर्शाते हैं।

```
let n1 = 2;  
console.log(n1)
```

```
let n2 = 1.3;  
console.log(n2)
```

# DSG Support Multi Solution

```
let n3 = Infinity;  
console.log(n3)
```

```
let n4 = 'something here too' / 2;  
console.log(n4)
```

## उत्पादन

2  
1.3  
अनंत  
नेन

## 2. स्ट्रिंग

जावास्क्रिप्ट में एक [स्ट्रिंग](#) वर्णों की एक श्रृंखला है जो उद्धरण चिहनों से घिरी होती है। जावास्क्रिप्ट में तीन प्रकार के उद्धरण चिह्न होते हैं, जो हैं।

```
let s1 = "Hello There";  
console.log(s1);
```

```
let s2 = 'Single quotes work fine';  
console.log(s2);
```

```
let s3 = `can embed ${s1}`;  
console.log(s3);
```

## उत्पादन

नमस्ते  
एकल उद्धरण ठीक काम करते हैं  
नमस्ते वहाँ एम्बेड कर सकते हैं

जावास्क्रिप्ट में 'सिंगल' और "डबल" कोट्स के बीच कोई अंतर नहीं है। बैकटिक्स अतिरिक्त कार्यक्षमता प्रदान करते हैं क्योंकि उनकी मदद से हम उनके अंदर चर एम्बेड कर सकते हैं।

## 3. बूलियन

बूलियन प्रकार [के](#) केवल दो मान होते हैं अर्थात् सत्य और असत्य ।

```
let b1 = true;  
console.log(b1);
```

```
let b2 = false;  
console.log(b2);
```

# DSG Support Multi Solution

## उत्पादन

सत्य

असत्य

## 4. शून्य

विशेष शून्य मान किसी भी डिफ़ॉल्ट डेटा प्रकार से संबंधित नहीं है। यह अपना स्वयं का एक अलग प्रकार बनाता है जिसमें केवल शून्य मान होता है।

```
let age = null;  
console.log(age)
```

## उत्पादन

व्यर्थ

'null' डेटा प्रकार एक विशेष मान को परिभाषित करता है जो कुछ नहीं, या रिक्त मान को दर्शाता है।

## 5. अपरिभाषित

एक चर जिसे घोषित किया गया है लेकिन किसी मान के साथ आरंभ नहीं किया गया है, उसे स्वचालित रूप से अपरिभाषित मान असाइन किया जाता है। इसका मतलब है कि चर मौजूद है, लेकिन इसे कोई मान असाइन नहीं किया गया है।

```
let a;  
console.log(a);
```

## उत्पादन

अपरिभाषित

## 6. प्रतीक (ES6 में प्रस्तुत)

ES6 में पेश किए गए प्रतीक, अद्वितीय और अपरिवर्तनीय आदिम मान हैं जिनका उपयोग ऑब्जेक्ट गुणों के लिए पहचानकर्ता के रूप में किया जाता है। वे ऑब्जेक्ट में अद्वितीय कुंजियाँ बनाने में मदद करते हैं, जिससे अन्य गुणों के साथ टकराव को रोका जा सकता है।

```
let s1 = Symbol("Geeks");  
let s2 = Symbol("Geeks");  
console.log(s1 == s2);
```

## उत्पादन



# DSG Support Multi Solution

असत्य

## 7. बिगइंट (ES2020 में प्रस्तुत)

[BigInt](#) एक अंतर्निहित ऑब्जेक्ट है जो  $2^{53}$  से बड़ी पूर्ण संख्याओं को दर्शाने का एक तरीका प्रदान करता है। सबसे बड़ी संख्या जिसे जावास्क्रिप्ट विश्वसनीय रूप से Number प्राइमेटिव के साथ दर्शा सकता है वह  $2^{53}$  है, जिसे MAX\_SAFE\_INTEGER स्थिरांक द्वारा दर्शाया जाता है।

```
let b = BigInt("0b10101010010101010011111111111111111111");
console.log(b);
```

## गैर-आदिम डेटा प्रकार

आदिम डेटा प्रकारों से प्राप्त डेटा प्रकारों को गैर-आदिम डेटा प्रकार के रूप में जाना जाता है। इसे व्युत्पन्न डेटा प्रकार या संदर्भ डेटा प्रकार के रूप में भी जाना जाता है।

### 1. वस्तु

जावास्क्रिप्ट [ऑब्जेक्ट](#) कुंजी-मूल्य जोड़े हैं जिनका उपयोग डेटा संग्रहीत करने के लिए किया जाता है, जिन्हें {} या new कीवर्ड के साथ बनाया जाता है। वे मौलिक हैं क्योंकि जावास्क्रिप्ट में लगभग सब कुछ एक ऑब्जेक्ट है।

```
let gfg = {
  type: "Company",
  location: "Noida"
}
console.log(gfg.type)
```

## उत्पादन

कंपनी

### 2. सारणी

ऐसे एक विशेष प्रकार का ऑब्जेक्ट है जिसका उपयोग मानों के क्रमबद्ध संग्रह को संग्रहीत करने के लिए किया जाता है, जो किसी भी डेटा प्रकार का हो सकता है।

```
let a1 = [1, 2, 3, 4, 5];
console.log(a1);
```

```
let a2 = [1, "two", { name: "Object" }, [3, 4, 5]];
console.log(a2);
```

## उत्पादन

```
[ 1, 2, 3, 4, 5 ]
```

```
[ 1, 'दो', { नाम: 'ऑब्जेक्ट' }, [ 3, 4, 5 ] ]
```

# DSG Support Multi Solution

## 3. कार्य

जावास्क्रिप्ट में फ़ंक्शन पुनः प्रयोज्य कोड का एक ब्लॉक होता है जिसे कॉल किए जाने पर एक विशिष्ट कार्य करने के लिए डिज़ाइन किया गया है ↓

```
// Defining a function to greet a user
function greet(name) { return "Hello, " + name + "!"; }
// Calling the function
console.log(greet("Ajay"));
```

### उत्पादन

नमस्ते अजय!

## 4. दिनांक ऑब्जेक्ट

जावास्क्रिप्ट में दिनांक [ऑब्जेक्ट का](#) उपयोग दिनांकों और समय के साथ काम करने के लिए किया जाता है, जिससे दिनांक निर्माण, हेरफेर और स्वरूपण की सुविधा मिलती है।

```
// Creating a new Date object for the current date and time
let currentDate = new Date();
// Displaying the current date and time
console.log(currentDate);
```

### उत्पादन

2025-03-08T06:23:33.202Z

## 5. नियमित अभिव्यक्ति

जावास्क्रिप्ट में RegExp ([रेगुलर एक्सप्रेशन](#)) एक ऑब्जेक्ट है जिसका उपयोग स्ट्रिंग्स में मिलान करने वाले टेक्स्ट के लिए खोज पैटर्न को परिभाषित करने के लिए किया जाता है।

```
// Creating a regular expression to match the word "hello"
let pattern = /hello/;
// Testing the pattern against a string
let result = pattern.test("Hello, world!"); // Returns true because "Hello" matches the pattern
console.log(result);
```

### उत्पादन

असत्य

# DSG Support Multi Solution

डेटा प्रकारों के बारे में रोचक तथ्य

**1. गतिशील रूप से टाइप किया गया:** जावास्क्रिप्ट चर किसी विशिष्ट डेटा प्रकार से बंधे नहीं होते हैं। मुख्य रूप से डेटा प्रकार को मान के साथ संग्रहीत किया जाता है (चर नाम के साथ नहीं) और रन टाइम पर तय और जाँच की जाती है।

```
let x = 42;
console.log(x)
```

```
x = "hello";
console.log(x)
```

```
x = [1, 2, 3]
console.log(x)
```

## उत्पादन

```
42
```

```
नमस्ते
```

```
[ 1, 2, 3 ]
```

**2. सब कुछ एक ऑब्जेक्ट है (एक तरह से):** जावास्क्रिप्ट में, [फंक्शन ऑब्जेक्ट हैं](#), [सरणियाँ ऑब्जेक्ट हैं](#), और यहां तक कि आदिम मान भी अस्थायी रूप से ऑब्जेक्ट की तरह व्यवहार कर सकते हैं जब आप उन पर गुणों तक पहुंचने का प्रयास करते हैं।

```
let s = "hello";
console.log(s.length);
```

```
// Example with a number
let x = 42;
console.log(x.toString());
```

```
// Example with a boolean
let y = true;
console.log(y.toString());
```

```
/* Internal Working of primitives
   to be treated as objects
```

```
// Temporary wrapper object
let temp = new String("hello");
```

```
console.log(temp.Length); // 5
```

```
// The wrapper is discarded after use
temp = null; */
```

# DSG Support Multi Solution

## उत्पादन

```
5
42
सत्य
```

**3. NaN अपने आप के बराबर नहीं है:** NaN का मतलब है “नॉट-ए-नंबर”, इसका उपयोग कम्प्यूटेशनल त्रुटि को दर्शाने के लिए किया जाता है। NaN तकनीकी रूप से संख्या प्रकार का है।

```
console.log(typeof NaN);
console.log(NaN === NaN);
```

## उत्पादन

```
संख्या
असत्य
```

**4. एक प्रतीक कभी भी दूसरे के बराबर नहीं होता** :Symbol एक अद्वितीय और अपरिवर्तनीय डेटा प्रकार है जिसका उपयोग अक्सर निजी गुण और विधियाँ बनाने के लिए किया जाता है। प्रतीक कभी भी किसी अन्य प्रतीक के बराबर नहीं होते।

```
let s1 = Symbol("abc");
let s2 = Symbol("abc");
console.log(s1 === s2);
```

## उत्पादन

```
असत्य
```

**5. अपरिभाषित और शून्य:** अपरिभाषित एक चर का प्रतिनिधित्व करता है जिसे घोषित किया गया है लेकिन असाइन नहीं किया गया है, जबकि शून्य एक स्पष्ट असाइनमेंट है जो "कोई मान नहीं" का प्रतिनिधित्व करता है।

**6. पूर्णांक केवल फ्लोटिंग पॉइंट संख्याएँ हैं।** केवल एक ही प्रकार की संख्या है जो पूर्णांक और फ्लोटिंग पॉइंट संख्या दोनों को कवर करती है।

```
let x = 42; // Integer
let y = 42.5; // Floating-point
```

```
console.log(typeof x);
console.log(typeof y);
```

## उत्पादन

# DSG Support Multi Solution

संख्या

संख्या

**7. एक अक्षर भी एक स्ट्रिंग है।** अक्षरों के लिए कोई अलग प्रकार नहीं है। एक अक्षर भी एक स्ट्रिंग है।

```
let s1 = "gfg"; // String
let s2 = 'g'; // Character
```

```
console.log(typeof s1);
console.log(typeof s2);
```

उत्पादन

gfg

g

## JavaScript Opreators

जावास्क्रिप्ट ऑपरेटर ऐसे प्रतीक या कीवर्ड हैं जिनका उपयोग मानों और चरों पर संचालन करने के लिए किया जाता है। वे जावास्क्रिप्ट अभिव्यक्तियों के निर्माण खंड हैं और विभिन्न तरीकों से डेटा में हेरफेर कर सकते हैं।

जावास्क्रिप्ट द्वारा समर्थित विभिन्न ऑपरेटर हैं।

1. जावास्क्रिप्ट अंकगणितीय ऑपरेटर

अंकगणितीय ऑपरेटर गणितीय गणनाएँ करते हैं जैसे जोड़, घटाव, गुणा आदि।

```
const sum = 5 + 3; // Addition
const diff = 10 - 2; // Subtraction
const p = 4 * 2; // Multiplication
const q = 8 / 2; // Division
console.log(sum, diff, p, q);
```

उत्पादन

8 8 8 4

- + दो संख्याओं को जोड़ता है.
- - पहली संख्या से दूसरी संख्या घटाता है।
- \* दो संख्याओं को गुणा करता है.
- / पहली संख्या को दूसरी से विभाजित करता है।

# DSG Support Multi Solution

## 2. जावास्क्रिप्ट असाइनमेंट ऑपरेटर

[असाइनमेंट ऑपरेटर का](#) उपयोग वेरिएबल को मान असाइन करने के लिए किया जाता है। वे मान असाइन करने से पहले जोड़ या गुणा जैसे ऑपरेशन भी कर सकते हैं।

```
let n = 10;  
n += 5;  
n *= 2;  
console.log(n);
```

### उत्पादन

30

- = किसी चर को मान निर्दिष्ट करता है।
- += परिणाम को चर में जोड़ता है और असाइन करता है।
- \*= गुणा करता है और परिणाम को चर में निर्दिष्ट करता है।

## 3. जावास्क्रिप्ट तुलना ऑपरेटर

[तुलना ऑपरेटर](#) दो मानों की तुलना करते हैं और एक बूलियन (सत्य या असत्य) लौटाते हैं। वे सशर्त कथनों में निर्णय लेने के लिए उपयोगी होते हैं।

```
console.log(10 > 5);  
console.log(10 === "10");
```

### उत्पादन

सत्य

असत्य

- > जाँचता है कि क्या बायाँ मान दाएँ मान से अधिक है।
- === सख्त समानता (प्रकार और मूल्य दोनों) की जांच करता है।
- अन्य ऑपरेटरों में <, <=, >=, और !== शामिल हैं।

## 4. जावास्क्रिप्ट लॉजिकल ऑपरेटर्स

[तुलना ऑपरेटर](#) मुख्य रूप से तार्किक संचालन करने के लिए उपयोग किए जाते हैं जो मानों के बीच समानता या अंतर निर्धारित करते हैं।

```
const a = true, b = false;  
console.log(a && b); // Logical AND  
console.log(a || b); // Logical OR
```

### उत्पादन

# DSG Support Multi Solution

असत्य

सत्य

- यदि दोनों ऑपरेंड सत्य हैं तो `&&` सत्य लौटाता है।
- `||` यदि कम से कम एक ऑपरेंड सत्य है तो `true` लौटाता है।
- `!` बूलियन मान को अस्वीकार करता है।

## 5. जावास्क्रिप्ट बिटवाइज़ ऑपरेटर

[बिटवाइज़ ऑपरेटर](#) संख्याओं के बाइनरी निरूपण पर कार्य करते हैं।

```
const res = 5 & 1; // Bitwise AND
console.log(res);
```

## उत्पादन

1

- `&` एक बिटवाइज़ AND निष्पादित करता है।
- `|` एक बिटवाइज़ OR निष्पादित करता है।
- `^` बिटवाइज़ XOR निष्पादित करता है।
- `~` एक बिटवाइज़ NOT निष्पादित करता है।

## 6. जावास्क्रिप्ट टर्नरी ऑपरेटर

टर्नरी [ऑपरेटर](#) सशर्त कथनों का संक्षिप्त रूप है। इसमें तीन ऑपरेंड लगते हैं।

```
const age = 18;
const status = age >= 18 ? "Adult" : "Minor";
console.log(status);
```

## उत्पादन

वयस्क

शर्त ? अभिव्यक्ति1 : यदि शर्त सत्य है तो अभिव्यक्ति2 अभिव्यक्ति1 का मूल्यांकन करता है, अन्यथा अभिव्यक्ति2 का मूल्यांकन करता है।

## 7. जावास्क्रिप्ट कॉमा ऑपरेटर

[कॉमा ऑपरेटर \(,\)](#) मुख्य रूप से अपने ऑपरेंड का बाएं से दाएं क्रमिक रूप से मूल्यांकन करता है और सबसे दाएं ऑपरेंड का मान लौटाता है।

```
let n1, n2
const res = (n1 = 1, n2 = 2, n1 + n2);
console.log(res);
```



# DSG Support Multi Solution

## उत्पादन

3

- प्रत्येक अभिव्यक्ति का मूल्यांकन बाएं से दाएं किया जाता है।
- अभिव्यक्ति का अंतिम परिणाम सबसे दाहिना मान है।

## 8. जावास्क्रिप्ट यूनरी ऑपरेटर

यूनरी ऑपरेटर एकल ऑपरेंड पर कार्य करते हैं (जैसे, वृद्धि, कमी)।

```
let x = 5;
console.log(++x); // Pre-increment
console.log(x--); // Post-decrement (Output: 6, then x becomes 5)
```

## उत्पादन

6

6

- ++ मान को 1 से बढ़ाता है.
- — मान को 1 से घटाता है.
- typeof चर का प्रकार लौटाता है.

## 9. जावास्क्रिप्ट रिलेशनल ऑपरेटर्स

जावास्क्रिप्ट रिलेशनल ऑपरेटर्स का उपयोग इसके ऑपरेंड की तुलना करने और उनके बीच संबंध निर्धारित करने के लिए किया जाता है। वे तुलना परिणाम के आधार पर बूलियन मान (सत्य या असत्य) लौटाते हैं।

```
const obj = { length: 10 };
console.log("length" in obj);
console.log([] instanceof Array);
```

## उत्पादन

सत्य

सत्य

- **in** जाँचता है कि क्या ऑब्जेक्ट में कोई गुण मौजूद है।
- **instanceof** जाँचता है कि क्या ऑब्जेक्ट कन्स्ट्रक्टर का उदाहरण है।

## 10. जावास्क्रिप्ट बिगइंट ऑपरेटर

बिगइंट ऑपरेटर सुरक्षित पूर्णांक सीमा से परे संख्याओं के साथ गणना की अनुमति देते हैं।

# DSG Support Multi Solution

```
const big1 = 123456789012345678901234567890n;  
const big2 = 987654321098765432109876543210n;  
console.log(big1 + big2);
```

## उत्पादन

```
111111111101111111110111111111100एन
```

- जोड़, घटाव और गुणा जैसे कार्य बिगइंट के साथ काम करते हैं।
- BigInt लिटरल को दर्शाने के लिए **n** प्रत्यय का उपयोग करें ।

## 11. जावास्क्रिप्ट स्ट्रिंग ऑपरेटर

[जावास्क्रिप्ट स्ट्रिंग ऑपरेटर्स](#) में संयोजन (+) और संयोजन असाइनमेंट (+=) शामिल हैं, जिनका उपयोग स्ट्रिंग्स को जोड़ने या स्ट्रिंग्स को अन्य डेटा प्रकारों के साथ संयोजित करने के लिए किया जाता है।

```
const s = "Hello" + " " + "World";  
console.log(s);
```

## उत्पादन

```
हैलो वर्ल्ड
```

- + स्ट्रिंग को संयोजित करता है.
- += किसी मौजूदा स्ट्रिंग में जोड़ता है.

## 12. जावास्क्रिप्ट चेनिंग ऑपरेटर (?.)

वैकल्पिक [चेनिंग](#) ऑपरेटर, यदि गुण मौजूद नहीं है तो, त्रुटियों को फेंके बिना, गहराई से नेस्टेड गुणों तक सुरक्षित पहुंच की अनुमति देता है।

```
const obj = { name: "Aman", address: { city: "Delhi" } };  
console.log(obj.address?.city);  
console.log(obj.contact?.phone);
```

## उत्पादन

```
दिल्ली
```

```
अपरिभाषित
```

- ?. सुरक्षित रूप से किसी संपत्ति या विधि तक पहुँचता है।
- यदि संपत्ति मौजूद नहीं है तो अपरिभाषित लौटाता है.

## Function JavaScript

# DSG Support Multi Solution

जावास्क्रिप्ट में फ़ंक्शन कोड के पुनः प्रयोज्य ब्लॉक हैं जिन्हें विशिष्ट कार्य करने के लिए डिज़ाइन किया गया है। वे आपको कोड को व्यवस्थित करने, पुनः उपयोग करने और मॉड्यूलर बनाने की अनुमति देते हैं। यह इनपुट ले सकता है, क्रियाएँ कर सकता है और आउटपुट लौटा सकता है।

```
function sum(x, y) {  
    return x + y;  
}  
console.log(sum(6, 9));
```

## उत्पादन

15

### फ़ंक्शन सिंटैक्स और कार्यप्रणाली

फ़ंक्शन परिभाषा को कभी-कभी फ़ंक्शन घोषणा या फ़ंक्शन कथन भी कहा जाता है। जावास्क्रिप्ट में फ़ंक्शन बनाने के नियम नीचे दिए गए हैं:

- फ़ंक्शन कीवर्ड से शुरू करें, उसके बाद,
- उपयोगकर्ता-परिभाषित फ़ंक्शन नाम (उपर्युक्त उदाहरण में, नाम **sum** है )
- कोष्ठकों में बंद और अल्पविराम से अलग किए गए मापदंडों की सूची (उपर्युक्त उदाहरण में, पैरामीटर **x** और **y** हैं )
- फ़ंक्शन के मुख्य भाग को बनाने वाले कथनों की सूची, जो कर्ली ब्रेसज़ {} में संलग्न है (उपर्युक्त उदाहरण में, कथन “return x + y” है)।

### वापसी विवरण

कुछ स्थितियों में, हम कुछ ऑपरेशन करने के बाद फ़ंक्शन से कुछ मान वापस करना चाहते हैं। ऐसे मामलों में, हम रिटर्न का उपयोग करते हैं। यह एक वैकल्पिक कथन है। उपरोक्त फ़ंक्शन में, “sum()” परिणाम के रूप में दो का योग लौटाता है।

### फ़ंक्शन पैरामीटर

पैरामीटर्स किसी फ़ंक्शन को दिए गए इनपुट होते हैं। उपरोक्त उदाहरण में, sum() दो पैरामीटर्स लेता है, x और y.

### फ़ंक्शन कॉल करना

फ़ंक्शन को परिभाषित करने के बाद, अगला चरण फ़ंक्शन का उपयोग करने के लिए उन्हें कॉल करना है। हम फ़ंक्शन नाम का उपयोग करके फ़ंक्शन को कॉल कर सकते हैं, जिसे कोष्ठक के बीच संलग्न पैरामीटर के मान से अलग किया जाता है।

```
// Function Definition  
function welcomeMsg(name) {
```

# DSG Support Multi Solution

```
return ("Hello " + name + " welcome to GeeksforGeeks");
}

let nameVal = "User";

// calling the function
console.log(welcomeMsg(nameVal));
```

## उत्पादन

नमस्कार उपयोगकर्ता, GeeksforGeeks में आपका स्वागत है

## कार्य क्यों?

- कार्यों का उपयोग कई बार किया जा सकता है, जिससे अतिरेकता कम हो जाती है।
- जटिल समस्याओं को प्रबंधनीय टुकड़ों में तोड़ें।
- कार्यान्वयन विवरण छिपाकर जटिलता का प्रबंधन करें.
- समस्याओं को हल करने के लिए स्वयं को पुनरावर्ती रूप से बुला सकते हैं।

## फ़ंक्शन इनवोकेशन

आपके द्वारा लिखा गया फ़ंक्शन कोड जब भी कॉल किया जाएगा, निष्पादित हो जाएगा।

- किसी घटना द्वारा ट्रिगर किया गया (उदाहरण के लिए, किसी उपयोगकर्ता द्वारा बटन क्लिक करना).
- जब जावास्क्रिप्ट कोड से स्पष्ट रूप से बुलाया जाता है।
- स्वचालित रूप से निष्पादित, जैसे कि [स्व-आह्वान कार्यों](#) में ।

## फ़ंक्शन अभिव्यक्ति

यह फ़ंक्शन नाम के बिना फ़ंक्शन घोषणा के समान है। [फ़ंक्शन अभिव्यक्तियों को](#) एक चर असाइनमेंट में संग्रहीत किया जा सकता है।

## वाक्यविन्यास:

```
let geeksforGeeks= फ़ंक्शन(paramA, paramB) {
  // कथनों का सेट
}

const mul = function (x, y) {
  return x * y;
};
console.log(mul(4, 5));
```

## उत्पादन

20

# DSG Support Multi Solution

## तीर फ़ंक्शन

[एरो फ़ंक्शन](#) , ES6 में प्रस्तुत फ़ंक्शन लिखने के लिए एक संक्षिप्त सिंटैक्स है , और वे अपने स्वयं के this संदर्भ को बाध्य नहीं करते हैं।

### वाक्यविन्यास:

फ़ंक्शन\_नाम = (तर्क1, तर्क2, ..) => अभिव्यक्ति

```
const a = ["Hydrogen", "Helium", "Lithium", "Beryllium"];
```

```
const a2 = a.map(function (s) {  
  return s.length;  
});
```

```
console.log("Normal way ", a2);
```

```
const a3 = a.map((s) => s.length);
```

```
console.log("Using Arrow Function ", a3);
```

## उत्पादन

सामान्य तरीका [ 8, 6, 7, 9 ]

एरो फ़ंक्शन का उपयोग करना [ 8, 6, 7, 9 ]

## तत्काल लागू फ़ंक्शन अभिव्यक्ति (IIFE)

[IIFE फ़ंक्शन](#) उनकी परिभाषा के तुरंत बाद निष्पादित किए जाते हैं। इनका उपयोग अक्सर पृथक स्कोप बनाने के लिए किया जाता है।

```
(function () {  
  console.log("This runs immediately!");  
})();
```

## उत्पादन

यह तुरंत चलता है!

## कॉलबैक फ़ंक्शन

कॉलबैक [फ़ंक्शन को](#) किसी अन्य फ़ंक्शन के लिए तर्क के रूप में पास किया जाता है और उस फ़ंक्शन के पूरा होने के बाद निष्पादित किया जाता है।

```
function num(n, callback) {  
  return callback(n);  
}
```

```
const double = (n) => n * 2;
```

# DSG Support Multi Solution

```
console.log(num(5, double));
```

## उत्पादन

10

## अनाम कार्य

[अनाम फ़ंक्शन](#) बिना नाम वाले फ़ंक्शन होते हैं। इन्हें अक्सर अन्य फ़ंक्शन के लिए तर्क के रूप में उपयोग किया जाता है।

```
setTimeout(function () {  
    console.log("Anonymous function executed!");  
}, 1000);
```

## नेस्टेड फ़ंक्शन

अन्य फ़ंक्शन के भीतर परिभाषित फ़ंक्शन को [नेस्टेड फ़ंक्शन](#) कहा जाता है। उनके पास अपने मूल फ़ंक्शन के चर तक पहुंच होती है।

```
function outerFun(a) {  
    function innerFun(b) {  
        return a + b;  
    }  
    return innerFun;  
}
```

```
const addTen = outerFun(10);  
console.log(addTen(5));
```

## उत्पादन

15

## शुद्ध कार्य

[शुद्ध फ़ंक्शन](#) समान इनपुट के लिए समान आउटपुट लौटाते हैं और साइड इफ़ेक्ट उत्पन्न नहीं करते हैं। वे अपने दायरे से बाहर स्टेट को संशोधित नहीं करते हैं, जैसे कि वैश्विक चर को संशोधित करना, तर्क के रूप में पारित ऑब्जेक्ट की स्थिति को बदलना, या I/O ऑपरेशन करना।

```
function pureAdd(a, b) {  
    return a + b;  
}
```

```
console.log(pureAdd(2, 3));
```

## उत्पादन

5

# DSG Support Multi Solution

## कार्यों की मुख्य विशेषताएं

- **पैरामीटर और तर्क:** फ़ंक्शन पैरामीटर (प्लेसहोल्डर) स्वीकार कर सकते हैं और तर्क (मान) के साथ कॉल किए जा सकते हैं।
- **रिटर्न मान:** फ़ंक्शन रिटर्न कीवर्ड का उपयोग करके मान लौटा सकते हैं।
- **डिफ़ॉल्ट पैरामीटर:** फ़ंक्शन पैरामीटर को डिफ़ॉल्ट मान निर्दिष्ट किए जा सकते हैं।

## जावास्क्रिप्ट में फंक्शन्स के लाभ

- **पुनः प्रयोज्यता:** एक बार कोड लिखें और उसका कई बार उपयोग करें।
- **मॉड्यूलरिटी:** जटिल समस्याओं को छोटे, प्रबंधनीय टुकड़ों में तोड़ना।
- **बेहतर पठनीयता:** फ़ंक्शन कोड को समझना आसान बनाते हैं।
- **रखरखाव:** संपूर्ण कोडबेस को प्रभावित किए बिना एक स्थान पर परिवर्तन किए जा सकते हैं।

## सही फ़ंक्शन प्रकार का चयन करना

- **नियमित पुनः प्रयोज्य फ़ंक्शन** के लिए फ़ंक्शन घोषणाओं का उपयोग करें।
- **संक्षिप्त, एक-पंक्ति फ़ंक्शन** के लिए तीर फ़ंक्शन का उपयोग करें।
- **तुरंत चलने वाले कोड** के लिए IIFE का उपयोग करें।
- **API कॉल** जैसे एसिंक्रोनस ऑपरेशन के लिए कॉलबैक फ़ंक्शन का उपयोग करें।
- **बिना किसी दुष्प्रभाव के पूर्वानुमानित व्यवहार** के लिए शुद्ध फ़ंक्शन का उपयोग करें।

## Messaging in JavaScript

JavaScript has three kind of popup boxes: Alert box, Confirm box, and Prompt box.

### Alert Box

An alert box is often used if you want to make sure information comes through to the user.

When an alert box pops up, the user will have to click "OK" to proceed.

### Syntax

```
window.alert("sometext");
```

The `window.alert()` method can be written without the window prefix.

### Example

```
alert("I am an alert box!");
```



# DSG Support Multi Solution

[Try it Yourself »](#)

## Confirm Box

A confirm box is often used if you want the user to verify or accept something.

When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.

If the user clicks "OK", the box returns **true**. If the user clicks "Cancel", the box returns **false**.

## Syntax

```
window.confirm("sometext");
```

The `window.confirm()` method can be written without the window prefix.

## Example

```
if (confirm("Press a button!")) {  
  txt = "You pressed OK!";  
} else {  
  txt = "You pressed Cancel!";  
}
```

[Try it Yourself »](#)

## Prompt Box

A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

# DSG Support Multi Solution

## Syntax

```
window.prompt("sometext","defaultText");
```

The `window.prompt()` method can be written without the window prefix.

## Example

```
let person = prompt("Please enter your name", "Harry Potter");
let text;
if (person == null || person == "") {
  text = "User cancelled the prompt.";
} else {
  text = "Hello " + person + "! How are you today?";
}
```

[Try it Yourself »](#)

## Line Breaks

To display line breaks inside a popup box, use a back-slash followed by the character n.

## Example

```
alert("Hello\nHow are you?");
```

## JavaScript Events

**JavaScript Events** are **actions or occurrences** that happen in the browser. They can be triggered by various user interactions or by the browser itself.

```
<html>
<script>
  function myFun() {
    document.getElementById(
      "gfg").innerHTML = "GeeksforGeeks";
  }
</script>

<body>
  <button onclick="myFun()">Click me</button>
```

# DSG Support Multi Solution

```
<p id="gfg"></p>
</body>
</html>
```

- The onclick attribute in the <button> calls the myFun() function when clicked.
- The myFun() function updates the <p> element with id="gfg" by setting its innerHTML to "GeeksforGeeks".
- Initially, the <p> is empty, and its content changes dynamically on button click.

## Event Types

JavaScript supports a variety of event types. Common categories include:

- **Mouse Events:** click, dblclick, mousemove, mouseover, mouseout
- **Keyboard Events:** keydown, keypress, keyup
- **Form Events:** submit, change, focus, blur
- **Window Events:** load, resize, scroll

## Common JavaScript Events

Event Attribute	Description
<a href="#">onclick</a>	Triggered when an element is clicked.
<a href="#">onmouseover</a>	Fired when the mouse pointer moves over an element.
<a href="#">onmouseout</a>	Occurs when the mouse pointer leaves an element.
<a href="#">onkeydown</a>	Fired when a key is pressed down.
<a href="#">onkeyup</a>	Fired when a key is released.
<a href="#">onchange</a>	Triggered when the value of an input element changes.
<a href="#">onload</a>	Occurs when a page has finished loading.
<a href="#">onsubmit</a>	Fired when a form is submitted.
<a href="#">onfocus</a>	Occurs when an element gets focus.

# DSG Support Multi Solution

Event Attribute	Description
<a href="#">onblur</a>	Fired when an element loses focus.

// Mouse Event

```
document.addEventListener("mousemove", (e) => {  
  console.log(`Mouse moved to (${e.clientX}, ${e.clientY})`);  
});
```

// Keyboard Event

```
document.addEventListener("keydown", (e) => {  
  console.log(`Key pressed: ${e.key}`);  
});
```

- The mousemove event tracks cursor movement.
- The keydown event captures key presses.

Event Handling Methods

## 1. Inline HTML Handlers

```
<button onclick="alert('Button clicked!')">Click Me</button>
```

## 2. DOM Property Handlers

```
let btn = document.getElementById("myButton");  
btn.onclick = () => {  
  alert("Button clicked!");  
};
```

## 3. addEventListener() (Preferred)

```
btn.addEventListener("click", () => {  
  alert("Button clicked using addEventListener!");  
});
```

addEventListener() is the most versatile and recommended method as it supports multiple event listeners and removal of listeners.

Event Propagation

JavaScript events propagate in two phases:

- **Capturing Phase:** Event travels from the root to the target element.
- **Bubbling Phase:** Event travels from the target element back to the root.

```
document.querySelector("div").addEventListener("click", () => {  
  console.log("Div clicked");  
}, true); // Capturing phase
```

# DSG Support Multi Solution

```
button.addEventListener("click", (e) => {
  console.log("Button clicked");
  e.stopPropagation(); // Stops propagation
});
```

- Setting true in `addEventListener` makes it capture events during the capturing phase.
- [stopPropagation\(\)](#) halts further propagation.

## Event Delegation

[Event delegation](#) allows you to handle events efficiently by attaching a single listener to a parent element.

```
document.querySelector("ul").addEventListener("click", (e) => {
  if (e.target.tagName === "LI") {
    console.log(`Clicked on item: ${e.target.textContent}`);
  }
});
```

Events are delegated to list, reducing the need to add listeners to each list items.

## Preventing Default Behavior

Certain elements have default actions (e.g., links navigating to URLs).

Use [preventDefault\(\)](#) to override them.

```
document.querySelector("a").addEventListener("click", (e) => {
  e.preventDefault();
  console.log("Link click prevented");
});
```

`preventDefault()` stops the link from navigating.

## Practical Applications

### 1. Form Validation

```
<html>
<body>
  <h2>Form Validation</h2>
  <form id="example">
    <input type="text" placeholder="Enter something" id="formInput" />
    <button type="submit">Submit</button>
  </form>
  <script>
    document.querySelector("#example").addEventListener("submit", (e) => {
      let input = document.querySelector("#formInput");
      if (!input.value) {
        e.preventDefault();
      }
    });
  </script>
</body>
</html>
```

# DSG Support Multi Solution

```
    alert("Input cannot be empty");
  }
});
</script>
```

```
</body>
```

```
</html>
```

## 2. Dynamic Content

```
<html>
```

```
<body>
```

```
  <h2>Dynamic Content</h2>
```

```
  <button id="button">Add Element</button>
```

```
  <script>
```

```
    document.querySelector("#button").addEventListener("click", () => {
```

```
      let newDiv = document.createElement("div");
```

```
      newDiv.textContent = "New Element Added";
```

```
      newDiv.style.margin = "10px 0";
```

```
      document.body.appendChild(newDiv);
```

```
    });
```

```
  </script>
```

```
</body>
```

```
</html>
```

## 3. Interactive Lists

```
<html>
```

```
<body>
```

```
  <h2>Interactive Lists</h2>
```

```
  <ul id="lists">
```

```
    <li>Interactive Item 1</li>
```

```
    <li>Interactive Item 2</li>
```

```
    <li>Interactive Item 3</li>
```

```
  </ul>
```

```
  <script>
```

# DSG Support Multi Solution

```
let ul = document.querySelector("#lists");

ul.addEventListener("click", (e) => {
  if (e.target.tagName === "LI") {
    e.target.style.backgroundColor = "yellow";
  }
});
</script>

</body>

</html>
```

## Website

वेबसाइट कई वेब पेजों का संग्रह है, और वेब पेज डिजिटल फ़ाइलें हैं जो HTML (हाइपरटेक्स्ट मार्कअप लैंग्वेज) का उपयोग करके लिखी जाती हैं। अपनी वेबसाइट को दुनिया के हर व्यक्ति तक पहुँचाने के लिए, इसे चौबीसों घंटे इंटरनेट से जुड़े कंप्यूटर पर संग्रहीत या होस्ट किया जाना चाहिए। ऐसे कंप्यूटर को **वेब सर्वर** के रूप में जाना जाता है।

वेबसाइट के वेब पेज हाइपरलिंक और हाइपरटेक्स्ट से जुड़े होते हैं और एक ही इंटरफ़ेस और डिज़ाइन साझा करते हैं। वेबसाइट में कुछ अतिरिक्त दस्तावेज़ और फ़ाइलें भी हो सकती हैं जैसे कि छवियाँ, वीडियो या अन्य डिजिटल संपत्तियाँ।

इंटरनेट के हर क्षेत्र में प्रवेश करने के साथ, हम सभी प्रकार के कारणों और उद्देश्यों के लिए वेबसाइट देखते हैं। इसलिए, हम यह भी कह सकते हैं कि एक वेबसाइट को एक डिजिटल वातावरण के रूप में भी माना जा सकता है जो सूचना और समाधान प्रदान करने और लोगों, स्थानों और चीजों के बीच बातचीत को बढ़ावा देने में सक्षम है ताकि उस संगठन के लक्ष्यों का समर्थन किया जा सके जिसके लिए इसे बनाया गया था।

**वेबसाइट के घटक:** हम जानते हैं कि वेबसाइट वेब-सर्वर पर होस्ट किए गए वेबपेजों का एक संग्रह है। ये वेबसाइट बनाने के लिए घटक हैं।



# DSG Support Multi Solution

- **वेबहोस्ट:** होस्टिंग वह स्थान है जहाँ वेबसाइट भौतिक रूप से स्थित होती है। वेबपेजों (लिंक किए गए वेबपेजों) के समूह को वेबसाइट कहलाने का लाइसेंस तभी मिलता है जब वेबपेज वेबसर्वर पर होस्ट किया जाता है। वेबसर्वर फ़ाइलों का एक समूह होता है जो उपयोगकर्ता के कंप्यूटर पर तब भेजा जाता है जब वे वेबसाइट का पता निर्दिष्ट करते हैं।
- **पता:** किसी वेबसाइट का पता जिसे वेबसाइट का URL भी कहा जाता है। जब कोई उपयोगकर्ता किसी वेबसाइट को खोलना चाहता है तो उसे वेबसाइट का पता या URL वेब ब्राउज़र में डालना होता है, और पूछी गई वेबसाइट वेबसर्वर द्वारा डिलीवर कर दी जाती है।
- **होमपेज:** होम पेज किसी भी वेबपेज का एक बहुत ही आम और महत्वपूर्ण हिस्सा होता है। यह पहला वेबपेज होता है जो किसी विज़िटर के वेबसाइट पर आने पर दिखाई देता है। किसी वेबसाइट का होम पेज बहुत महत्वपूर्ण होता है क्योंकि यह वेबसाइट का लुक और फील सेट करता है और दर्शकों को वेबसाइट के बाकी पेजों पर ले जाता है।
- **डिज़ाइन:** यह वेबसाइट का अंतिम और समग्र रूप और अनुभव है जो नेविगेशन मेनू, ग्राफिक्स, लेआउट, नेविगेशन मेनू आदि जैसे उचित उपयोग और एकीकरण तत्वों का परिणाम है।
- **विषय-वस्तु :** वेबसाइट पर मौजूद हर वेब पेज मिलकर वेबसाइट की विषय-वस्तु बनाते हैं। वेबपेज पर अच्छी विषय-वस्तु वेबसाइट को अधिक प्रभावी और आकर्षक बनाती है।
- **नेविगेशन संरचना:** किसी वेबसाइट की नेविगेशन संरचना पृष्ठों का क्रम है, जो किससे लिंक करता है इसका संग्रह है। आमतौर पर, इसे कम से कम एक नेविगेशन मेनू द्वारा एक साथ रखा जाता है।

## वेबसाइट तक कैसे पहुँचें?

जब हम ब्राउज़र सर्च बार में कोई खास यूआरएल टाइप करते हैं, तो ब्राउज़र वेब सर्वर से पेज का अनुरोध करता है और वेब सर्वर ब्राउज़र को आवश्यक वेब पेज और उसकी सामग्री लौटा देता है। अब, यह स्टैटिक और डायनेमिक वेबसाइट के मामले में सर्वर द्वारा आवश्यक जानकारी लौटाने के तरीके से अलग है।

## वेबसाइट के प्रकार:

- स्थैतिक वेबसाइट
- गतिशील वेबसाइट

**स्टैटिक वेबसाइट:** स्टैटिक वेबसाइट में, सर्वर द्वारा वेब पेज लौटाए जाते हैं जो HTML, CSS या JavaScript जैसी सरल भाषाओं का उपयोग करके बनाए गए प्रीबिल्ट सोर्स कोड फ़ाइलें हैं। स्टैटिक वेबसाइट में सर्वर पर (उपयोगकर्ता के अनुसार) सामग्री की कोई प्रोसेसिंग नहीं होती है। वेब पेज सर्वर द्वारा बिना किसी बदलाव के लौटाए जाते हैं, इसलिए, स्टैटिक वेबसाइट तेज़ होती हैं। डेटाबेस के साथ कोई इंटरैक्शन नहीं होता है। साथ ही, वे कम खर्चीले होते हैं क्योंकि होस्ट को अलग-अलग भाषाओं के साथ सर्वर-साइड प्रोसेसिंग का समर्थन करने की आवश्यकता नहीं होती है।

# DSG Support Multi Solution

**डायनेमिक वेबसाइट:** डायनेमिक वेबसाइट में, वेब पेज सर्वर द्वारा लौटाए जाते हैं जिन्हें रनटाइम के दौरान प्रोसेस किया जाता है, इसका मतलब है कि वे पहले से बनाए गए वेब पेज नहीं हैं, बल्कि वे सर्वर-साइड स्क्रिप्टिंग भाषाओं जैसे PHP, Node.js, ASP.NET और सर्वर द्वारा समर्थित कई अन्य भाषाओं की मदद से उपयोगकर्ता की मांग के अनुसार रनटाइम के दौरान बनाए जाते हैं। इसलिए, वे स्थिर वेबसाइटों की तुलना में धीमी हैं लेकिन डेटाबेस के साथ अपडेट और इंटरैक्शन संभव हैं। डायनेमिक वेबसाइट का उपयोग स्टैटिक वेबसाइट पर किया जाता है क्योंकि स्टैटिक वेबसाइट की तुलना में अपडेट बहुत आसानी से किए जा सकते हैं (जहाँ हर पेज में बदलाव करना आवश्यक है) लेकिन डायनेमिक वेबसाइट में, एक बार एक सामान्य बदलाव करना संभव है, और यह सभी वेब पेजों में दिखाई देगा।

पूरे इंटरनेट पर विभिन्न प्रकार की वेबसाइटें हैं, हमने आपको एक संक्षिप्त विचार देने के लिए कुछ सबसे सामान्य श्रेणियों को चुना है -

- **ब्लॉग:** इस प्रकार की वेबसाइटें किसी व्यक्ति या व्यक्तियों के एक छोटे समूह द्वारा प्रबंधित की जाती हैं, वे किसी भी विषय को कवर कर सकते हैं - वे आपको फैशन टिप्स, संगीत टिप्स, यात्रा टिप्स, फिटनेस टिप्स दे सकते हैं। आजकल पेशेवर ब्लॉगिंग ऑनलाइन पैसे कमाने का एक बाहरी लोकप्रिय तरीका बन गया है।
- **ई-कॉमर्स:** ये वेबसाइट ऑनलाइन शॉप के नाम से जानी जाती हैं। ये वेबसाइट हमें उत्पादों की खरीद और उत्पादों और सेवाओं के लिए ऑनलाइन भुगतान करने की सुविधा देती हैं। स्टोर को स्टैंडअलोन वेबसाइट के रूप में भी संभाला जा सकता है।
- **पोर्टफोलियो:** इस प्रकार की वेबसाइटें फ्रीलांसर रिज्यूमे के विस्तार के रूप में कार्य करती हैं। यह संभावित ग्राहकों को आपके काम को देखने का एक सुविधाजनक तरीका प्रदान करता है, साथ ही आपको अपने कौशल या सेवाओं का विस्तार करने की अनुमति भी देता है।
- **ब्रोशर:** इस प्रकार की वेबसाइटें मुख्य रूप से छोटे व्यवसायों द्वारा उपयोग की जाती हैं, इस प्रकार की वेबसाइटें डिजिटल बिजनेस कार्ड के रूप में कार्य करती हैं, और संपर्क जानकारी प्रदर्शित करने और कुछ ही पृष्ठों के साथ सेवाओं का विज्ञापन करने के लिए उपयोग की जाती हैं।
- **समाचार और पत्रिकाएं:** इन वेबसाइटों को कम स्पष्टीकरण की आवश्यकता होती है, इस प्रकार की वेबसाइटों का मुख्य उद्देश्य अपने पाठकों को वर्तमान मामलों से अद्यतन रखना है जबकि पत्रिकाएं मनोरंजन पर ध्यान केंद्रित करती हैं।
- **सोशल मीडिया:** हम सभी कुछ प्रसिद्ध सोशल मीडिया वेबसाइट जैसे फेसबुक, ट्विटर, रेडिट और कई अन्य के बारे में जानते हैं। ये वेबसाइट आमतौर पर लोगों को अपने विचार, चित्र, वीडियो और अन्य उपयोगी घटक साझा करने के लिए बनाई जाती हैं।

# DSG Support Multi Solution

- **शैक्षिक:** शैक्षिक वेबसाइटें समझने में काफी सरल होती हैं, जैसा कि उनके नाम से ही पता चलता है। इन वेबसाइटों को ऑडियो या वीडियो या छवियों के माध्यम से जानकारी प्रदर्शित करने के लिए डिज़ाइन किया गया है।
- **पोर्टल:** इस प्रकार की वेबसाइटों का उपयोग स्कूल, संस्थान या किसी व्यवसाय के आंतरिक उद्देश्यों के लिए किया जाता है। इन वेबसाइटों में अक्सर एक लॉगिन प्रक्रिया होती है जो छात्रों को उनकी क्रेडेंशियल जानकारी तक पहुंचने की अनुमति देती है या कर्मचारियों को उनके ईमेल और अलर्ट तक पहुंचने की अनुमति देती है।

*Thanks & Regards*

*DSGSupport Multi Solution*