

GİRİŞ

Günümüz dijital dünyasında her geçen gün kullanıcı ihtiyaçları artmakta ve alışkanlıkları değişmekte, daha karmaşık ve çok yönlü bir hale gelmektedir. Kullanıcıların, farklı ekran boyutlarında farklı cihazları kullanırken sorun yaşamamaları günümüz teknolojik çağında olmazsa olmaz bir gereklilik olmuştur. Sorun yaşayan, kötü deneyimi tecrübe eden kullanıcılar ilgili web siteleri ya da uygulamalarını kullanmayı çok beklemeden terk edecektir. Bu noktada duyarlı web tasarımı imdadımıza yetişiyor. Duyarlı web tasarımı ister masaüstü bilgisayar, ister akıllı telefon, ister tablet olsun, herhangi bir cihazda en iyi deneyimi sorunsuz sağlamak için görüntülenecek web sitesi veya uygulamaların içeriklerini, düzenlerini, işlevlerini otomatik olarak ayarlamalarına imkân tanıyacak şekilde tasarlanmasını ve oluşturulmasını içermektedir.

Kullanılan cihaza göre içeriği optimize etme prensibine sahip olan duyarlı web tasarımı, bir uygulamanın veya bir web sitesinin farklı cihazlarda ve farklı ekran boyutlarında uyumlu ve sorunsuz bir şekilde görüntülenmesini sağlamaktadır. Duyarlı web tasarımının işlevselliği ve kullanılabilirliği içeren esneklik, grid ve medya sorguları gibi temel ilkeleri mevcuttur. Esneklik, web sitesi ya da uygulamalardaki içeriğin ve tasarımın farklı cihazlara ve ekran boyutlarına uyum sağlamasını mümkün kılmaktadır. Grid sistemleri sayfa düzenini organize etmekte ve dinamik olarak uyarlama işlemini yapmaktadır. Medya sorguları ise tarayıcının özelliklerini sorgulayarak stil ve düzeni dinamik olarak uyarlamayı sağlamaktadır.

İnternet erişimi önceleri sadece masaüstü bilgisayarlar üzerinden kullanılmaktaydı. Artık kullanıcılar web içeriklerine akıllı telefonlar, akıllı televizyonlar, tabletler, akıllı saatler gibi farklı cihazlar üzerinden erişmektedirler. Bu kadar farklı cihazın ortaya çıkıp kullanılmasıyla tasarımcılara ve geliştiricilere uyumlu, kullanıcı dostu, sorunsuz bir deneyim sunma gerekliliği ortaya çıkmıştır. Duyarlı web tasarımı tüm bu ihtiyaçlara cevap veren bir yaklaşım sunmaktadır.

DUYARLI WEB TASARIMI NEDİR?

Duyarlı web tasarımı, bir web sitesinin ya da bir uygulamanın bilgisayar, telefon, tablet gibi farklı cihazlarda ve farklı ekran boyutlarında uygun bir şekilde görüntülenmesini sağlayan bir tasarım yaklaşımıdır.

Duyarlı Web tasarımı, tasarım ve geliştirmenin ekran boyutu, platform ve yönelime bağlı olarak kullanıcının davranışına ve ortamına yanıt vermesi gerektiğini öne sürmektedir. Herhangi bir kullanıcı dizüstü bilgisayarından bir web sitesine girip daha sonra bir tablete ya da akıllı telefona geçip aynı web sitesine girdiğinde, web sitesi çözünürlük, görüntü boyutu gibi özelliklere uyum sağlamak için dinamik olarak geçiş yapmalıdır. Diğer bir ifadeyle web sitesi kullanıcının cihaz tercihlerine göre otomatik olarak yanıt verecek teknolojiye sahip olmalıdır. Bu sürekli üretilip piyasaya sunulan her yeni aygıt için farklı bir tasarım ve geliştirme aşaması ihtiyacını ortadan kaldıracaktır.

DUYARLI WEB TASARIMI KAVRAMLARI

Duyarlı web tasarımı, web sitelerinin insanların internete erişmek için kullandıkları çok çeşitli cihazlara uyum sağlamasına olanak tanınmaktadır. İşletmeler için daha iyi kullanıcı deneyimleri, gelişmiş erişilebilirlik ve uygun maliyetli çözümler sağlamaktadır. Duyarlı web tasarımı ile ilgili önemli olan bazı kavramlar şu şekildedir:

- Akışkan Izfara (Grid) Düzeni
- Duyarlı Görüntüler
- Medya Sorguları
- Kesme Noktaları
- Mobil Öncelikli Yaklaşım
- İçerik Hiyerarşisi
- Geliştirilmiş Kullanıcı Deneyimi
- Maliyet Etkinliği

DUYARLI WEB TASARIMININ FAYDALARI

Duyarlı web tasarımı farklı cihazlarda ve ekran boyutlarında tutarlı bir kullanıcı deneyimi sağlamaktadır. Duyarlı bir tasarımla, kullanıcılar bir cihazdan diğerine geçerken kullanıcı deneyimini kaçırmadan sorunsuz bir şekilde içeriği görebilir ve içerikten faydalanabilirler.

İnsanlar dünya genelinde internete erişmek için binlerce farklı cihaz kullanabilmektedir. Web sitelerinin bilgisayar, tablet, akıllı saat gibi birçok cihazda aynı anda kusursuz, uyumlu ve sorunsuz

görünmesi gerekmektedir. Bu kullanıcıların memnuniyeti için çok önemlidir. Siteler ya da uygulamalara farklı cihazlardan erişmek isteyen ziyaretçilerin sorun yaşaması durumunda rakiplere karşı dezavantajlı duruma düşülmektedir. Kötü tasarlanmış bir web sitesini daha küçük ekranlarda görüntülemeye çalışıp hayal kırıklığı yaşayan kullanıcı için o sitedeki her şey bozuk ve yersiz görünür. Duyarlı tasarımlar kullanılmadığında dönüşümler ve yatırım getirileri en üst düzeye çıkarılamaz, hedef kitlenin çoğuna ulaşılamaz.

Duyarlı web tasarımı faydalarını aşağıdaki başlıklar halinde sıralayabiliriz:

- Geliştirilmiş Mobil Uyumluluk
- Maliyet Etkinliği
- Geliştirilmiş SEO Performansı
- Daha Kolay Web Sitesi Yönetimi
- Geliştirilmiş Mobil Uyumluluk

Web siteleri mobil cihaza uyum sağlamıyorsa, sitelerini mobil ziyaretçilere göre düzenleyen rakiplere karşı dezavantajlı duruma düşmektedir.

Maliyet Etkinliği

Duyarlı bir web tasarımı yaklaşımı benimsemek, daha düşük geliştirme ve bakım masrafının yanı sıra, gelişmiş kullanıcı deneyimi ve etkileşimi, daha iyi SEO performansı ve daha fazla erişim ve erişilebilirlik yoluyla maliyet etkinliği sunmaktadır. Bu durum sadece para tasarrufu sağlamakla kalmaz, aynı zamanda işletmeler için daha yüksek bir yatırım getirisi ile sonuçlanır.

Geliştirilmiş SEO Performansı

Duyarlı web tasarımı, herhangi bir web sitesinin arama motoru optimizasyonu (SEO) performansını artırmada önemli rol oynamaktadır. Google gibi arama motorları, mobil kullanıcılar için daha iyi bir kullanıcı deneyimi sağladıkları için arama sonuçlarında mobil uyumlu web sitelerine öncelik vermektedirler.

Duyarlı web tasarımının uygulanması yalnızca kullanıcı deneyimini iyileştirmekle kalmaz, aynı zamanda SEO performansını da artırmaktadır. Duyarlı web tasarımı mobil uyumlu tasarıma odaklanmakta ve cihazlar arasında birleşik bir web sitesi deneyimi sağlamaktadır. Bu sebeple web sitelerinin arama motoru sonuçlarındaki görünürlüğünü artırılmaktadır.

Daha Kolay Web Sitesi Yönetimi

"Daha Kolay Web Sitesi Yönetimi", duyarlı web tasarımı yaklaşımıyla hazırlanmış bir web sitesinin bakımı ve güncellenmesindeki kolaylık ve verimliliği ifade etmektedir. Web yöneticileri için genel yönetim sürecini basitleştirip, daha az zaman alıcı hale getirmektedir. Genel olarak, duyarlı web tasarımı sayesinde daha kolay web sitesi yönetimi, işletmeler ve web sitesi sahipleri için daha fazla üretkenlik, daha az iş yükü ve maliyet tasarrufu sağlanmaktadır.

DUYARLI WEB TASARIMINDA KULLANILAN YENİLİKÇİ TEKNOLOJİLER

Duyarlı web tasarımında kullanılan yenilikçi teknolojiler, duyarlı web tasarımının temelini oluşturan ve web deneyimini geliştirmeye yönelik çeşitli araçları içeren, web tasarımında kullanıcı deneyimini geliştirmek, performansı artırmak ve farklı cihazlarda uyumluluk sağlamak için kullanılan önemli araçlardır. Yenilikçi teknolojilerle tasarlanmış bir web sitesi veya uygulama, bu teknolojileri etkili bir şekilde kullanarak kullanıcılara daha etkileşimli ve tatmin edici bir deneyim sunmaktadır. Duyarlı web tasarımında sıkça kullanılan yenilikçi teknolojilerden bazıları şu şekildedir:

- CSS Framework'leri
- Flexbox ve Grid Layout
- Duyarlı Görüntüler (Responsive Images)
- JavaScript Framework'leri
- Progressive Web Apps (PWA)
- Service Workers
- WebAssembly (Wasm)
- CSS Framework'leri

Bir CSS Framework, kullanıma hazır bir CSS kütüphanesidir. Bir projeye sıfırdan başlamak yerine, CSS Framework geliştiricilere tekrarlayacakları ve ince ayarlar yapabilecekleri bir kullanıcı ara yüzünü hızlıca oluşturabilmeleri için araçlar sunmaktadır. Bununla birlikte CSS Framework standartlara daha uygun web siteleri oluşturulmasını sağlamaktadır.

Tailwind, Bootstrap, Foundation gibi CSS framework'leri, duyarlı web tasarımını kolaylaştıran ve hızlandıran önemli araçlardır. Bu framework'ler, önceden tasarlanmış bileşenler ve grid sistemleri sunarak tasarımcılara ve geliştiricilere esneklik sağlamaktadır.

Tailwind: Tailwind, web sitelerini hızlı ve kolayca şekillendirmek için kullanılmaktadır. Temel olarak özel kullanıcı ara yüzlerini hızlıca oluşturmak için bir yardımcı program öncelikli CSS Framework'üdür. Tailwind ile benzersiz bir kullanıcı ara yüzü oluşturmak için küçük bileşenleri bir araya getirmek yeterli olmaktadır. Tailwind ham bir CSS dosyasını alır ve bu dosyayı bir yapılandırma dosyası üzerinden işlemekte ve çıktı üretmektedir.

Bootstrap: Bootstrap, web ara yüzleri tasarlamaya yönelik açık kaynaklı bir ön uç Framework olup, web

tasarımları oluşturabilmeyi kolaylaştıran "kullanıma hazır" HTML, CSS ve JavaScript kodu koleksiyonundan oluşmaktadır.

Bootstrap kullanılarak, arkasındaki karmaşık kodu anlamak zorunda kalmadan çekici bir web görünümü oluşturulabilmektedir. İhtiyaç duyulan temel düzeyde, oldukça basit HTML, CSS bilgisi, eğer etkileşimli efektler oluşturulmak isteniyorsa biraz da JavaScript bilgisidir. Bootstrap açık kaynaklı olup kullanımını ücretsizdir.

Foundation: Foundation 2011 Eylül'de ZURB tarafından oluşturulmuştur. Herhangi bir cihazda çalışan duyarlı web siteleri, uygulamalar, e-postalar oluşturmayı kolaylaştıran açık kaynaklı ve duyarlı bir ön uç Framework olarak adlandırılmaktadır. Foundation tamamen duyarlı web uygulamaları oluşturulmasına imkân sağlamaktadır.

Flexbox ve Grid Layout

Flexbox ve Grid, sayfa düzenini ve bileşen yerleşimini yönetmekte büyük kolaylık sağlamaktadır. Özellikle grid sistemleri, sayfa tasarımını farklı ekran boyutlarına uyumlu hale getirmek için etkili bir araçtır.

Flexbox: Flexbox, hizalama yapılarını kontrol etmek için öğeler arasında dağıtılmış boşluklara sahip esnek ve verimli düzenlere sahip tek boyutlu bir düzen modeli olarak adlandırılmaktadır.

Grid Layout: Grid Layout, bir web sayfasındaki veya uygulamadaki öğelerin iki boyutlu düzeni için tasarlanmış bir CSS düzen yöntemi olarak adlandırılmaktadır. Grid layout Html'de değil CSS'de tanımlanan grid yapıları oluşturmak için bir yöntem sunmaktadır.

Duyarlı Görüntüler (Responsive Images)

Duyarlı görüntüler, cihaz çözünürlüğüne, ekran boyutuna, yönüne, ağ bağlantısına ve sayfa düzenine bağlı olarak doğru görüntüyü yüklemek için kullanılan teknikler bütünü olarak adlandırılmaktadır.

Tarayıcı, sayfa düzenine uyması için görüntüyü uzatmamalı ve yükleme zaman ve bant genişliği israfına neden olmamalıdır. Görüntüler hızlı yüklendiğinden ve insan gözüne net görüldüğünden kullanıcı deneyimini iyileştirmektedir.

JavaScript Framework'leri

JavaScript ile yazılmış kodları içeren kütüphaneler topluluğu olup yazılım geliştiricilerin işlerini çok kolaylaştırmaktadırlar. JavaScript framework'leri, kullanıcı ara yüzlerini ve sayfalar arasında geçişleri yönetmek için kullanılır. Bu framework'ler, tek sayfa uygulamalarını (Single Page Applications - SPA) geliştirmek için yaygın olarak tercih edilir.

Her bir JavaScript çatısı, yazılım geliştirmede farklı alanlar ve farklı amaçlar için önceden oluşturulmuş kodlar sunarak geliştiriciye zaman kazandırmaktadır. Yazılım geliştiricilerin birlikte çalışabileceği çok fazla sayıda JavaScript Framework kütüphanesi vardır.

Progressive Web Apps (PWA, Aşamalı web uygulaması)

Progressive Web Apps(PWA), bir mobil uygulama gibi görünen ve davranan bir web sitesi olarak adlandırılmaktadır.

PWA'lar, son kullanıcının bir uygulama mağazasını ziyaret etmesini, bir satın alma işlemi yapmasını ve yazılımı yerel olarak indirmesini gerektirmeden yerel mobil cihaz özelliklerinden yararlanmak için oluşturulmuştur. Bir PWA bir arama motoru sorgusu ile bulunabilir ve bir tarayıcı aracılığıyla hemen erişilebilir.

Service Workers

Servis Worker, kullanıcı etkileşimi olmadan bağımsız olarak tarayıcı arka planında çalışan bir betiktir. Ayrıca, kullanıcı tarafında çalışan bir proxy'yi andırır. Bu script ile sayfanın ağ trafiğini takip edebilir, anlık bildirimleri yönetebilir ve Cache API ile "offline first" web uygulamaları geliştirebilirsiniz.

Servis Worker ile sayfanın tüm ağ trafiği ve anlık bildirimler yönetilebilmektedir. Service Worker ve Cache API ile herhangi bir istek / yanıt çifti önbelleğe alınabilmekte ve bu çevrimdışı içeriğe istenilen zamanda erişebilmektedir.

WebAssembly (Wasm)

Genellikle Wasm olarak adlandırılan WebAssembly, özellikle web tarayıcıları için tasarlanmış ikili bir komut biçimi olarak adlandırılmaktadır. Kodun neredeyse yerel hızda yürütülmesini sağlayan düşük seviyeli bir sanal makinedir. Wasm, web'e yüksek performanslı uygulamalar getirmek amacıyla Mozilla, Google, Microsoft ve Apple gibi büyük tarayıcı satıcıları arasında ortak bir çaba olarak geliştirilmiştir. WebAssembly'nin en önemli avantajlarından biri, birden fazla programlama diliyle uyumlu olması ve bu sayede web geliştirme için çok yönlü bir çözüm olmasıdır.

GİRİŞ

Web dünyası teknolojik gelişmelere paralel olarak sürekli bir değişim ve gelişim halindedir. Web dünyasındaki bu değişim ve gelişime öncülük eden teknolojilerden birisi de HTML5'tir. HTML5, internet üzerindeki içeriği daha erişilebilir, daha etkileşimli ve zengin hale getirmek hedefiyle geliştirilen bir standarttır. İlk olarak 2014 yılında W3C tarafından resmi olarak kabul edilmiş olan HTML5, web geliştiricilerine yeni özellikler, geliştirilmiş performans ve daha iyi bir kullanıcı deneyimi sunabilmeyi amaçlamaktadır. HTML5, web dünyasında devrim niteliğinde bir dönüşüm başlatmıştır ve geleceğin standartlarına ışık tutan bir teknoloji olarak adlandırılmaktadır. HTML5 teknolojisinin ana hedeflerinden birisi, zengin medya desteğini güçlendirmektir. HTML5, ses ve video içeriklerini entegre etmede daha önceki versiyonlara kıyasla daha hızlı ve sorunsuz bir performans sunmaktadır. HTML5 entegre etme sürecini basitleştirmekte ve harici eklentilere ihtiyaç duymadan video ve ses içeriklerini desteklemektedir. Bu durum kullanıcıların daha etkileşimli ve çeşitli medya içeriklerine çok daha hızlı erişebilmesini mümkün kılmaktadır. HTML5'in bir diğer öncelikli hedeflerinden birisi de Mobil uyum ve duyarlılıktır. Mobil cihazların yaygınlaşmasıyla birlikte farklı cihazlarda ve ekran boyutlarında tutarlı bir deneyim sunmak günümüz çoklu platform ortamında olmazsa olmaz gerekliliklerdendir. Bu ihtiyacı karşılamak için HTML5 mobil uyumlu web siteleri oluşturmayı kolaylaştırmaktadır. HTML5'in güçlü yönlerinden birisi de tarih seçicileri, yerel doğrulama özellikleri ve gelişmiş input tipleri gibi özelliklere sahip olmasıdır. Bunlar kullanıcı dostu ve hata toleranslı form deneyimleri sunmaktadır.

HTML5 NEDİR?

HTML'in ilk versiyonu 1993 yılında yazılmıştır. O zamandan günümüze kadar çok daha gelişmiş olmasına rağmen günümüzde kullanılan son versiyonu HTML5'ten önce HTML4 son sürüm olup ve hala sınırlamaları vardı. HTML4'ün en büyük sorunu web geliştirici ve tasarımcılarının HTML'de desteklenmeyen içerik veya özellikler eklemek istemeleriydi. HTML'in desteği olmayan bu durumlarda kullanıcıların tarayıcı eklentileri yüklemeleri gerekmekteydi. Bu eklentileri yüklemeyen kullanıcılar ilgili içeriklere erişemeyeceklerdi. HTML5 bu standart dışı tescilli teknolojilere olan bağımlılığı ortadan kaldırmak için tasarlanmıştır. HTML5 ile çevrimdışı çalışan, yüksek çözünürlüklü video ve animasyonları destekleyen ve coğrafi olarak kullanıcıların nerede bulunduğunu bilen web uygulamaları oluşturulabilmektedir.

HTML5, web siteleri ve web uygulamaları oluşturmak için kullanılan Hypertext Markup Language'in (HTML) en son sürümüdür. HTML5'in resmi spesifikasyonu 2014 yılında yayınlanmıştır ve en yaygın kullanılan sürüm haline gelmiştir. HTML5, geliştiricilerin daha dinamik ve etkileşimli web sayfaları oluşturmaya olanak tanıyan birçok yeni öğe, nitelik ve API (Uygulama Programlama Ara yüzleri) sunmaktadır.

HTML5, web sayfalarının yapısını ve sunumunu tanımlayan kod olan Hypertext Markup Language'in en son standardıdır. Üç koddan oluşur: Yapıyı koruyan HTML; sunumu etkileyen CSS; ve web tarayıcılarında etkileşimli efektler oluşturan JavaScript.

HTML5, masaüstü bilgisayarlar, dizüstü bilgisayarlar, akıllı telefonlar ve tabletler de dahil olmak üzere çok çeşitli cihazlarla uyumlu olacak şekilde tasarlanmış ve geliştiricilerin birden fazla cihazda sorunsuz bir şekilde çalışan platformlar arası web uygulamaları oluşturmalarını kolaylaştırmıştır.

HTML5 NE İÇİN KULLANILIR?

Bir işaretleme dili olan HTML5, öncülleri HTML 1'den HTML4'e kadar olduğu gibi web içeriği oluşturmak ve yapılandırmak için kullanılır. HTML5, modern, dinamik ve ilgi çekici web içeriği ve uygulamaları oluşturmak için gerekli olan çok yönlü ve yaygın olarak kullanılan bir teknolojidir. HTML5 çok çeşitli amaçlar için kullanılmaktadır, bunlardan bazıları aşağıda sıralanmıştır:

Web siteleri , web uygulamaları oluşturma

Mobil geliştirme

Çapraz platform uyumluluğu

Erişilebilirlik

Video ve ses oynatma

E-ticaret

HTML5 İLE GELEN YENİLİKLER

HTML5 kendisinden önceki HTML sürümlerine göre birçok yeni özellik ve iyileştirme sağlayarak geliştiricilerin dinamik, etkileşimli ve erişilebilir web içeriği ve uygulamaları oluşturmasını kolaylaştırmıştır. Bununla birlikte HTML5 her proje için her zaman en iyi seçenektir denilemez. Bazı projeler için normal HTML yeterli olabilirken bazıları için farklı bir teknoloji daha uygun olabilmektedir.

HTML5 ile gelen yeniliklerden bazıları aşağıdaki gibi sıralanmıştır:

Yeni öğeler ve nitelikler

Geliştirilmiş multimedya desteği

Canvas öğesi

Geliştirilmiş formlar

Çevrimdışı depolama

Geliştirilmiş erişilebilirlik

Daha verimli kodlama

Çapraz platform uyumluluğu

HTML5 SAYFA YAPISI

HTML5 kodlarını yazıp çalıştırmak için Windows tabanlı bilgisayarlar için geliştirilmiş ücretsiz, açık kaynaklı bir HTML editörü olan Notepad++ programı kullanılabilir. Bunun yanı sıra Atom HTML Editörü, NetBeans, Microsoft Expression Studio, Sublime Text, Adobe Dreamweaver CC gibi editörler de kullanılmaktadır.

HTML5'in genel sayfa yapısında kullanılan etiketler:

HTML5'DE KALDIRILAN ETİKETLER

HTML4'ten sonraki sürüm olan HTML5'e geçişte bazı etiketler kaldırılmıştır. Bu kaldırılan etiketlerin bir kısmının yerine aynı işlevi gerçekleştiren yeni etiketler gelmiştir.

HTML5'te kaldırılan etiketler ,, , ,

,

, , , ,

, , olarak sıralanabilir.

ANLAMSAL HTML5 ETİKETLERİ

HTML5 Etiketleri

HTML5, çok fazla sayıda web tarayıcısı tarafından desteklenen çok sayıda yeni öğe getirmiştir.

HTML5'in temel unsurları aşağıda listelenmiştir :

: Bu yeni bölümlenme öğesi, belirli bir içeriği bir makalenin parçası olarak işaretlemek için kullanılmaktadır.

: Genellikle belge kenar çubuklarında kullanılan

ögesi, içinde bulunan içerikten başka bir içeriği tanımlamaktadır.

: Ses kaynaklarını ayırmak için bir veya daha fazla etiketiyle birlikte kullanılabilen

etiketi, kullanıcıların bir sayfanın müzik veya ses akışı gibi ses içeriğini yerleştirmesine olanak tanımaktadır.

: Çift yönlü izolasyon anlamına gelen bu öğenin işlevi, öğenin dışındaki metinden farklı bir yönde biçimlendirilebilecek bir metin parçasını izole etmektir.

: Bu öğe, grafikler çizmek ve bir sayfaya görüntü eklemek için JavaScript kullanılmaktadır. Kenarlıklar ve metin varsayılan olarak dahil edilmez ve etiketler şeffaftırlar.

: HTML kodları içinde veri saklamak için kullanılan etiketi, web sayfalarında kullanıcıya görünmeyen, ancak sunucuyla iletişim halinde olan verileri depolamak ve iletmek için kullanılmaktadır.

: Bu öğe, veri girildiğinde tarayıcının kullanıcılara sunacağı önceden ayarlanmış seçeneklerden oluşan bir açılır liste belirterek öğesi için bir 'otomatik tamamlama' özelliği sağlamaktadır.

ve :

etiketi HTML5 ile gelen yeni bir öğe olup içeriği gizlemek ve açmak için kullanılmaktadır. Bu etiket, etiketin içerdiği içeriğin başlangıcını ve sonunu tanımlamakta ve bir kullanıcı tıklayarak veya bir klavye kısa yolu kullanarak içeriğin açılmasını veya gizlenmesini sağlamaktadır.

ögesi,

açılış etiketi ve

kapanış etiketi olmak üzere iki bileşenden oluşmaktadır. İçindeki içeriği gizlemek veya açmak için etiketi kullanılmaktadır. etiketi, kullanıcıya içeriğin özeti veya başlığı olarak görüntülenmektedir.

: ögesi, bir iletişim kutusu veya alt pencere tanımlayarak bir web sayfasında açılır iletişim kutuları ve modal pencereler oluşturmayı kolaylaştırmaktadır.

: ögesi genel olarak ses veya video gibi multimedya içeriği biçimini alan üçüncü taraf uygulamaları gömmek için kullanılmaktadır.

ve

:

etiketi, web sayfalarında grafikler, resimler, videolar, tablolar gibi medya öğelerini gruplar ve bu öğeler için açıklama veya alt yazı sağlayan bir etikettir.

etiketi,

öğesinin içinde bulunarak medya öğesiyle ilgili açıklamayı sağlamaktadır.

: Bu öğe, bir bölüm ya da sayfa için bir altbilgi tanımlamaktadır. Genel olarak yazar hakkında bilgi, telif hakları, iletişim bilgileri, site haritası gibi ilgili okumalar içerir.

:

öğesi genellikle sayfanın başlığı ve başlığı ile ilgili bilgileri içermektedir. Tipik olarak, tanıtıcı bilgiler ya da gezinme bağlantıları için bir kapsayıcıyı ana hatlarıyla belirtmektedir.

: öğesi, formlarda bir anahtar çifti oluşturucu alanı belirtmek için kullanılmaktadır. Amacı, kullanıcılara güvenli bir kimlik doğrulama yöntemi sağlamaktır.

:

öğesi, sayfanın ana içeriğini ana hatlarıyla belirtir ve bu içerik söz konusu belgeye özgü olmalıdır.

: öğesi, bir paragraf içinde vurgulanması veya işaretlenmesi gereken metni tanımlamaktadır.

: Gösterge olarak da bilinen öğesi, önceden tanımlanmış bir aralıktaki skaler ölçümleri ve kesirli değerleri tanımlamak için kullanılmaktadır.

:

öğesi, web sitesinin genellikle geçerli sayfada başka bir noktaya veya farklı bir sayfaya yönlendiren gezinme bağlantılarına ayrılmış bölümlerini ana hatlarıyla belirtmektedir. Yaygın öğeleri arasında tablolar, menüler ve dizinler bulunmaktadır.

: öğesi, JavaScript ya da başka bir komut dosyası tarafından gerçekleştirilen bir hesaplamanın çıktısını temsil etmektedir.

: Genellikle JavaScript ile birlikte kullanılan öğesi, tamamlanan iş miktarı ya da indirme süresi gibi bir görevin ilerleme durumunu görselleştirmektedir.

, , ve : etiketi, özellikle Japonca öğrenenler için çok faydalıdır, çünkü karakterlerin telaffuzlarını açıklamakta ve okunmasını kolaylaştırmaktadır. HTML etiketi, “rp” kısaltması tarafından temsil edilen “ruby parenthesis” anlamına gelmektedir. Bu etiket, metin içinde “ruby” formatını kullanarak küçük bir yazıyı veya bir karakterin yanında parantezlerle küçük bir yazıyı göstermek için kullanılmaktadır. HTML etiketi, “ruby text” in kısaltılması olup, yan yana yerleştirilmiş metinlerdeki karakterlerin üzerindeki açıklamaları belirtmek için kullanılan bir HTML etikettir.

:

öğesi, üstbilgi ve altbilgi dahil olmak üzere web sayfalarının belirli bölümlerini tanımlamaktadır.

: Bu öğe SVG grafikleri için bir kap oluşturur. Kutular, yollar, daireler, grafik görüntüler ve metin çizmek için çok sayıda yöntemi bulunmaktadır.

: öğesi, tarih ve saati kullanıcılar tarafından okunabilir bir biçimde görüntülemektedir.

: Bu öğe, bir web sayfasına video içeriği yerleştirmek için kullanılmaktadır.

: öğesi, bir metin satırı içinde, gerektiğinde satır sonu eklenebilecek noktaları belirtmektedir.

Katmanlı Stil Sayfaları (ing.: Cascading Style Sheets, CSS), web sayfalarının görünümünü ve düzenini belirlemek için kullanılan bir stil dilidir. CSS, HTML gibi bir işaretleme dili olarak kabul edilir ve başlıca işlevi bir web sayfasındaki öğelerin (başlık, paragraf, liste, bağlantı, tablo, resim, form elemanı vb.) renkleri, boyutları, kenar boşlukları, arka planlar gibi görsel özelliklerin nasıl görüneceğini kontrol etmektir. Bu işlemi CSS seçiciler vasıtasıyla gerçekleştirir.

CSS seçiciler (ing.: CSS selectors), web sayfalarındaki HTML öğelerini seçmek ve onlara stil uygulamak için kullanılan kod parçacıklarıdır. Bu seçiciler, belirli öğeleri seçme ve onlara istenilen görünümü kazandırmayı sağlar. Temelde, CSS ile web sayfasındaki öğelerin tasarımını kontrol etmek için kullanılırlar.

Aşağıdaki CSS kodu bir web sayfası içerisindeki tüm başlıklarını seçer ve içerisinde yazılan CSS kurallarını bu etikete uygular.

```
h1 {  
color: blue;  
}
```

Bu kod uygulandığı sayfadaki tüm başlıklarının mavi renkli olmasını sağlar

CSS seçiciler, ayrıca web sayfası öğelerin hiyerarşisini anlamak için kullanılabilir. Örneğin, bir öğenin içindeki alt öğeleri seçmek veya belirli bir sınıfa sahip bir öğenin içindeki alt öğeleri seçmek mümkündür. Böylece daha ayrıntılı seçimler yapılabilir. Web sayfalarının tasarımı sürecinde (özellikle duyarlı web tasarımı) CSS seçicileri anlamak ve onları doğru bir şekilde kullanmak, web sayfasının görünümünü ve farklı cihazlara göre davranışını kontrol etmek konusunda önemlidir.

HTML ELEMAN SEÇİCİ

HTML eleman seçici, web sayfası içeriğindeki öğeleri etiket ismine (ing.: tag name) göre seçerek CSS kurallarını uygular. Web sayfası içeriğinde etiketi ile işaretlenmiş bir paragraf ve etiketi ile işaretlenmiş bir başlık varsa aşağıdaki kod ile tüm etiketlerinin ve etiketlerinin bu kurallardan etkileneceği aşağıdaki şekilde CSS kodu yazmak mümkündür.

```
p {  
....  
}
```

```
h1 {  
....  
}
```

Bu şekilde 120 civarındaki tüm HTML etiketlerine uygulanabilir.

ID SEÇİCİ

CSS dilinde, belirli HTML öğelerine etikete özgü bazı özellikler eklemek ve belirli öğeleri belirgin bir şekilde tanımlamak için "ID seçici" veya "kimlik seçici" kullanılmaktadır. Bir HTML belgesinde (her web sayfası HTML belgesi olarak bilinir) her bir öğe benzersiz bir kimliğe sahip olabilir. Bu kimlik "id" özelliği ile tanımlanır ve HTML belgesinde bu öğenin diğer öğelerden ayrışması sağlanır. HTML içerisinde ID özelliği "ana-bölge" olan bir öğesi varsa aşağıdaki CSS kodu bu öğesini seçer ve içerisinde tanımlı CSS kodlarını bu öğesine uygular.

```
#ana-bolge {  
background-color: lightgray;  
padding: 20px;  
}
```

CLASS SEÇİCİ

Bir web sayfasında bulunan öğelere uygulanmak üzere tekrar eden bir dizi CSS kuralı bulunabilir. Bu durumda bu kuralların uygulanacak olduğu HTML öğelerini seçmek için ayrı ayrı CSS kuralı yazmak gerekecektir. Ancak bu yöntem fazladan kod yazılmasına ve bakımı zorlaşan web sayfaları oluşturulmasına sebep olabilir. Buna bir çözüm olarak, tekrar eden CSS kuralları gruplanır ve her bir gruba ayrı bir sınıf ismi verilir. Bu şekilde bir HTML belgesinde aynı sınıf ismini taşıyan birden çok HTML öğesi olabilir ve tekrar eden kodlar farklı etiketlere uygulanabilir. Aşağıdaki örnekte 2 adet etiketi bulunmaktadır. Her iki paragraf öğesi de bilgi-mesajı adlı bir sınıfa sahiptir. Bu iki paragrafa uygulanacak CSS seçicileri yazmak için aşağıdaki CSS kodu yazılabilir.

HTML Kodu

Bu bir bilgi mesajıdır.
Bu da bir başka bilgi mesajıdır.

CSS Kodu

```
.bilgi-mesaji {  
background-color: #dff0d8;  
border: 1px solid #3c763d;  
padding: 10px;  
color: #3c763d;  
}
```

Yukarıdaki koddan da görüleceği gibi class seçici (ing. class selector) kullanmak için . (nokta) sembolü ile başlayan ve ardından sınıf adı gelen bir tanımlayıcı kullanılır (bilgi-mesajı). Burada her iki p etiketini seçmek için etiket seçici kullanılabilirdi. Ancak HTML belgesinde daha fazla p etiketi varsa tüm p etiketleri bundan etkilenebilirdi.

Class seçici kullanmanın en önemli avantajı da aynı CSS kurallarının farklı HTML etiketlerine uygulanabilmesi imkanındır. Böylece belirli bir CSS kuralını paylaşan ve birden çok öğeye uygulanan genel kurallar tanımlamak mümkün olur.

ETİKET DÜZEYİNDE SINIFLAR

Etiket düzeyinde seçiciler, belirli bir etiket için değil de sadece belli bir sınıfa ayrılmış etiketlere CSS kurallarını uygulamak için kullanılmaktadır.

Her etikete farklı CSS uygulamanın bir yolu etiketi sınıflara ayırmaktır.

HTML Kodu

Bu paragrafa farklı bir stil uygulanacak.

Bu paragrafa ise başka bir stil uygulanacak.

CSS Kodu

```
p.paragraf1 {  
color: blue;  
font-size: 18px;  
}  
  
p.paragraf2 {  
color: green;  
font-weight: bold;  
}
```

Burada “paragraf1” ve “paragraf2” sınıfları sadece etiketlerine uygulanabilecek şekilde hedefleme yapılmaktadır. Bu sayede etiketler de kendi aralarında farklı sınıflara ayrılabilir. Yukarıda sadece etiketi için etiket düzeyinde sınıf kodları verilmiştir. Elbetteki bu yaklaşım diğer HTML etiketleri için de kullanılabilir.

Sınıflara ayırma işleminin etiket düzeyinde yapılmasının bazı sebepleri vardır. Bu sebeplerin en önemlisi etiket düzeyinde sınıflandırma yapıldığında bu sınıfların diğer etiketlere uygulanamamasıdır.

Tasarımcıya bu şekilde sınırlama getirilmesi (çoğu durumda tasarımcı bu sınırlamayı kendi kendine yapar) hatalı seçici kullanarak web sayfalarının görünümünün değişmesine engel olmaktadır.

EVRENSEL ETİKET SEÇİCİ

CSS dilinde, belirli bir HTML ögesini veya sınıfları seçmeden tüm öğeleri seçmek için kullanılan "Universal Selector" veya "Evrensel Seçici" bulunmaktadır. Evrensel seçici, tüm HTML öğelerini seçmek için kullanılan bir seçicidir ve belirli bir öge türüne, sınıfa veya ID'ye bağlı kalmadan tüm öğeleri seçmek için kullanılmaktadır. Evrensel seçici, * (yıldız) sembolü ile temsil edilir.

```
* {  
margin: 0;  
padding: 0;  
box-sizing: border-box;  
}
```

Yukarıdaki CSS kodu, tüm HTML öğelerine sıfır iç kenar boşluğu, iç dolgu ve kenar kutu modelini sınırlayarak genel bir sıfırlama uygular. Böylece belirli öğelere veya sınıflara özel bir CSS kuralı uygulamadan önce varsayılan bir temel CSS kuralı belirlenebilmektedir.

Evrensel seçici kullanmanın en önemli avantajı, web tarayıcıların varsayılan CSS kurallarını sıfırlamak için kullanılabilmesidir. Ancak, evrensel seçiciyi aşırı kullanmak performans sorunlarına neden olabilir. Çünkü tüm öğeleri seçmek ve stil uygulamak tarayıcının iş yükünü artırabilir. Bu nedenle evrensel seçici içerisinde çok fazla CSS kuralı yazmaktan kaçınmak en verimli yoldur ve genellikle belirli bir hedefe odaklanan daha özel seçiciler (id seçici veya class) kullanmak, performans açısından daha etkilidir.

SEÇİCİ GRUPLAMA

Birden çok seçiciyi birleştirmek ve aynı stili bir grup öğeye uygulamak için gruplama seçiciler kullanılmaktadır. Gruplama seçici, birden çok CSS seçicisinin virgülle ayrılarak birleştirilmesi şeklinde kullanılmaktadır.

```
h1, p {  
color: gray;  
font-family: 'Arial', sans-serif;  
margin-bottom: 10px;  
}
```

Yukarıdaki CSS kodu, hem başlıklarına hem de

paragraflarına aynı renk, yazı tipi ve alt boşluk miktarını uygular. Burada etiket düzeyinde seçici örneği verilmiştir. Ancak gruplama seçiciler id ve sınıf seçicilerle de kullanılmaktadır.

CSS kuralları her etiket, id veya sınıf için ayrı ayrı yazarak da uygulanabilir. Bu durumda neden gruplama seçici kullanıldığı sorulabilir. Gruplama seçicileri, aynı CSS kurallarını paylaşan HTML öğeleri varsa daha temiz kod yazılmasını, bu kuralların tek bir yerde yönetilmesini sağlar, bu da bakımı kolaylaştırır ve yazılan CSS kodlarının daha anlaşılır olmasını sağlar.

KOMBİNATOR SEÇİCİLER

CSS dilinde birden çok CSS seçici kullanılabilir. Önceki bölümlerde görüldüğü gibi aynı CSS kuralları, seçiciler arasında virgül(,) kullanarak birden çok seçiciye uygulanabilmektedir. Birden çok seçici arasındaki ilişkilere göre kullanılacak farklı CSS seçiciler vardır. VSS seçiciler arasındaki ilişkiyi açıklayan kavrama kombinator (birleştirici) denilmektedir. CSS dilinde 4 farklı çeşit birleştirici bulunmaktadır.

soy seçici (boşluk karakteri), (ing.: descendant selector)
çocuk (child) seçici (>)

komşu kardeş seçici (ing.: adjacent sibling selector, +)
genel kardeş seçici (general sibling selector, ~)

Soy seçici (ing.: descendant selector) kullanılırken “boşluk karakteri kullanılır” ve belirli bir ögenin içindeki tüm alt öğeleri seçmek için kullanılır. Örneğin “.ust-bölge li” seçicisi ile class özelliği “ust-bölge” olan içerisinde yer alan tüm

- öğeleri seçilir.

Çocuk seçici (ing.: child selector) kullanılırken “>” sembolü kullanılmaktadır. Bu sembol belirli bir ögenin içerisindeki 1. derece çocuk öğeleri (doğrudan alt öğeleri) seçerek CSS

kuralları uygular. Yukarıda görülen soy seçiciden farkı şudur: Soy seçici üst ögenin içerisindeki tüm öğeleri seçerken çocuk seçici sadece 1. derecede çocuk öğeleri seçmektedir. Komşu kardeş seçici (ing.: adjacent sibling selector) kullanılırken “+” sembolü kullanılmaktadır. Bu sembol belirli bir öğeden hemen sonra gelen kardeş öğeleri seçerek CSS kuralları uygular. Bu seçici genellikle bir web sayfasındaki bir bölümde verilen bir listenin ilk öğesine farklı bir stil uygulanmak istediğinde ya da içerisinde bir çok haber özeti bölümü olan içeriklerden ilk haber özetine farklı bir stil uygulamak gibi farklı uygulamalarda kullanılabilir. Bazı durumlarda tüm kardeş öğeleri seçerek bir stil uygulanmak istenebilir. Bu durumda genel kardeş seçici (ing.: general sibling selector) kullanılmaktadır. Komşu kardeş seçici “~” (tilda) sembolü kullanılmaktadır. Bu sembol belirli bir öğe ile aynı seviyede olan kardeş öğeleri seçerek CSS kuralları uygular.

YALANCI SINIFLAR

Yalancı sınıflar (ing.: pseudo class) bir HTML öğesinin herhangi bir ID veya sınıf belirtmeden kendiliğinden var olan durumuna göre seçilmesini sağlar. Bu sayede istenen CSS kodları HTML içeriklerine uygulanmış olur. Kendiliğinden olan durumlara örnekler şu şekildedir:

bir bağlantının ziyaret edilmemiş olması (link)

bir bağlantının ziyaret edilmiş olması (visited)

bir bağlantının seçilmiş olması (active)

bir bağlantının HTML öğesinin fare imleci ile üzerine gelinmesi (hover)

Bir HTML öğesinin içinde bulunduğu bir üst ögenin ilk veya son elemanı olması (first-child, last-child)

Bir HTML öğesinin içinde bulunduğu bir üst ögenin baştan veya sondan n. son elemanı olması (nth-child, nth-last-child)

Yalancı sınıflar için CSS kodu oluşturmak aşağıdaki söz dizimine göre yapılmaktadır. Burada önemli olan “:” (iki nokta üst üste) karakterinin kullanılmasıdır.

```
selector:pseudo-class {  
property: value;  
...  
}
```

Burada “selector” yerine önceki bölümlerde görülen etiket, id, sınıf vb. seçiciler kullanılabilir. “pseudo-class” ifadesi yerine yukarıdaki durumlardan biri (visited, hover, first-child) kullanılmalıdır. Bu durumlar tarayıcı tarafından tanınması için doğru yazılmaları gerekmektedir.

YALANCI ÖĞELER

Bazı durumlarda web sayfası öğelerinin alt parçaları içinde CSS kuralları tanımlanabilmektedir. Bu işlem yalancı öğeler ile mümkün olabilmektedir. Yalancı öğeler (ing.:pseudo elements) HTML belgesinde yer alan öğelerin bir parçası için seçme işlemi gerçekleştirir. Bu sayede web sayfasındaki öğelerin istenen alt parçalarına CSS kuralları eklenebilir.

Yaygın olarak kullanılan bazı yalancı öğeler aşağıdaki gibidir:

::before ve ::after → Bu yalancı öğe, bir web sayfası içeriğinden hemen öncesinde ve sonrasında ek içerik eklemek için kullanılır.

::first-line ve ::first-letter → Bir web sayfası öğesinin ilk satırını ve ilk karakterini seçerek CSS kuralı uygulanmasını sağlar.

::selection → Bu yalancı öğe, web sayfası kullanıcısının herhangi bir HTML belgesindeki herhangi bir metni seçmesi durumunda seçilen metne uygulanacak CSS kurallarını belirlemek için kullanılır.

::placeholder → Bu yalancı öğe, form elemanlarındaki yer tutucu metni hedeflemek için kullanılır.

Duyarlı web tasarımı, web sitelerinin farklı cihazlarda veya ekran boyutlarında düzgün bir şekilde görüntülenmesini sağlamak için kullanılan bir tasarım yaklaşımıdır. Bunun için CSS medya sorguları ve konteyner sorguları kullanılmaktadır. Medya ve konteyner sorguları sayesinde her cihaz veya ekran için ayrı ayrı web tasarımı yapmak yerine tek bir tasarım ile tüm ekran boyutları ve cihaz türlerine görsel olarak uyumlu görünecek web siteleri tasarlamak mümkün olmaktadır. Bu nedenle medya ve konteyner sorguları duyarlı web tasarımı yaparken kullanılan en önemli konulardan biridir. CSS medya sorguları (ing.: media queries), web sitelerinin farklı cihazlarda ve ekran boyutlarında nasıl görüntüleneceğini ve davranacağını belirlemek için kullanılan CSS kod bloklarıdır. Konteyner sorguları (ing.: container queries) ise medya sorularının aksine ekran çözünürlüğünü sorgulamaz. Bunun yerine web sitesinde yer alan bir öğenin belirli bir boyutu veya oranı aştığında veya altına düştüğünde stil kurallarını uygulamak için kullanılır.

@MEDYA SORGUSUNUN KULLANIMI

CSS medya sorgusu web sitelerinin farklı cihaz ve ekran boyutlarına göre sorgulama yapar ve her sorgu için CSS kod bloğu oluşturur. Bu amaçla “@” simgesi ile başlayan “@media” sorgusu kullanılır. Genel “@media” sorgu sözdizimi aşağıdaki gibidir:

```
@media not|only mediatype and (expressions) {  
...  
CSS Kodları;  
}
```

Sorgu ifadesi “@media” ile başlar, ardından “not” (değil) veya “only” (sadece) operatörü gelir.

all
tüm cihaz türleri için kullanılır (varsayılan)

screen
bilgisayar, tablet, telefon gibi cihazlar için kullanılır

print
yazıcılar için kullanılır

speech
sesli okunabilen ekran okuyucular için kullanılır

“@media all” sorgusu, genel olarak tüm cihazlar için CSS kodları tanımlamayı sağlar. Yani, ekranlar, yazıcılar, sesli tarayıcılar ve diğer tüm cihazlar için stil kodlamalarını. “all” cihaz türü varsayılandır ve tüm tarayıcılarda kullanılır.

“@media screen” sorgusu CSS dilinde bilgisayarlar, tabletler, geniş ekranlar ve mobil telefon ekranlarına uygun stil tanımlamalarını yapmak için kullanılır. Bu sorgu, özellikle tarayıcı ekranları için tasarlanmış web sitelerinde veya uygulamalarda kullanışlıdır. @media screen sorgusu, web sayfasının ekranlarda nasıl görüntüleneceğini ve davranacağını belirlemek için kullanılır. Aşağıdaki CSS kodu ekran genişliğine göre bir stil ayarlaması yapmaktadır.

```
@media screen and (max-width: 800px) {  
body {  
background-color: lightgray;  
}  
h1 {  
font-size: 20px;  
}  
}
```

Yukarıdaki örnekte, ekran genişliği maksimum 800 piksel olan (800 pikselden daha dar) ekranlar için uygulanacak CSS kodlarıdır.

“@media print” sorgusu, CSS dilinde medya sorguları aracılığıyla yazdırma işlemi için stil tanımlamalarını yapmak için kullanılır. Bu sorgu, web sayfalarının kağıda veya diğer yazdırılabilir ortamlara nasıl aktarılacağını belirler. “@media print” sorgusu, belgelerin yazdırıldığında veya PDF dosyalarına dönüştürüldüğünde nasıl görüneceğini ve davranacağını kontrol etmek için kullanılır.

“@media speech” sorgusu, web sayfalarının sesli tarayıcılar ve ekran okuyucular gibi metin tabanlı sesli araçlar için nasıl görüntüleneceğini ve nasıl davranacağını belirlemek için kullanılır. Bu sorgu, web sayfalarının metin okuma cihazları için en uygun görünüm elde edilmesine yardımcı olur. Özellikle erişilebilirlik açısından önemlidir ve görsel içeriğin, yazılı metin ve sesli içeriğe dönüştürülmesine yardımcı olur.

Farklı cihaz türlerini sorgulamanın dışında ekran genişliklerine göre koşul ifadeleri yazılabilmektedir. Koşul ifadeleri yazarken “max-width” ya da “min-width” özellikleri kullanılmaktadır. Bu özelliklerin alacağı değerler “:” (iki nokta üst üste, CSS özellik ayarlama operatörü) karakterinden sonra piksel cinsinden verilmektedir. Örneğin;

```
min-width: 768px
```

```
max-width: 1024px
```

şeklinde maksimum veya minimum ekran genişlikleri verilerek sorgular yapılabilmektedir.

Küçük ekranlar için (genellikle mobil cihazlar) 600 pikselden az olması sorgusu yapılabilir (@media screen and (max-width: 600px) { ... }) ve bu kod bloğu içerisinde menüyü gizlemek ve metin yazı tipi boyutunu küçültmek için CSS kodları kullanılabilir.

Orta büyüklükteki ekranlar için (genellikle tablet cihazlar) 768 pikselden fazla ve 1024 pikselden az olması sorgusu yapılabilir (@media screen and (min-width: 768px) and (max-width: 1024px) { ... }).

Ekran boyutlarına göre tasarım yaparken dikkat edilmesi gereken bazı hususlar vardır. Öncelikle her ekran genişliği için ayrı ayrı sorgu yapmak yerine sayısal aralık vererek sorgulamalar yapılmalıdır. Bunun dışında tüm ekranlar için ortak olabilecek yazı fontu, rengi gibi ortak olabilecek bazı CSS kuralları herhangi bir sorgu içine alınmamalıdır, CSS kodları içerisindeki ortak alanına yazılmalıdır. Medya sorgu blokları içerisine sadece sayfa genişliği, font büyüklüğü, yerleşim için yazılan CSS kodları gibi farklı olabilecek stil kodları yazılmalıdır.

EKRAN BOYUTLARINA GÖRE TASARIM

Duyarlı web tasarımında farklı ekran türlerine göre iki farklı yaklaşım tercih edilmektedir: 1) Mobil öncelikli tasarım (ing.: mobile first design), 2) Masaüstü öncelikli yaklaşım (ing.: desktop first design). Mobil öncelikli yaklaşımda daha çok kullanıcı deneyimi ön plana çıkmaktadır ve artan mobil cihaz kullanımını göz önünde bulundurarak mobil cihaz ekranlarına göre tasarım yapmaya odaklanır. Bu yaklaşımda ilk olarak web sayfasının görünümünün mobil ekranlar için ayarlanmakta diğer ekran türleri için özellikler eklemek istendiğinde @media sorguları oluşturulmaktadır. Günümüzde daha çok mobil cihazlar ile web sayfalarının ziyaret edilmesi sebebiyle mobil öncelikli tasarım daha çok tercih edilmektedir.

Masaüstü öncelikli yaklaşımda ise büyük ekranlı cihazlara odaklanılır ve varsayılan olarak web sayfasının masaüstü cihazlarda görüntüleneceği kabul edilir. Bu yaklaşımda mobil cihazlar için tasarım yapılmak istendiğinde ekstrasdan @media sorguları yazılmalıdır.

Tasarım yaparken hangi yaklaşımın kullanılacağı, hedef kitleye ve proje gereksinimlerine bağlıdır. Eğer tasarım yapacağınız web sitesine hedef kitlesindeki kullanıcıların çoğu mobil cihazlardan erişim sağlıyorsa bu durumda mobil öncelikli tasarım tercih edilmelidir. Ayrıca hız ve performans önceliği söz konusu ise yine mobil öncelikli tasarım tercih edilmelidir. Eğer projeniz masaüstü cihazlara yönelik ise veya daha karmaşık içeriklere sahip web sayfaları içeriyorsa masaüstü öncelikli tasarım tercih etmek daha uygundur.

CİHAZ TÜRLERİNE ÖZEL TASARIM ÖNERİLERİ

Web tasarımcısı, mobil öncelikli tasarım, veya masaüstü öncelikle tasarım seçeneklerinden dilediğini seçerek web sitesi tasarımını gerçekleştirebilir. Ancak yaygın olarak kabul edilen ve dikkate alınması önerilen bazı tasarım önerileri vermek faydalı olacaktır.

Tasarım yapılırken ilk olarak düşünülmesi gereken kullanıcı deneyimidir. Kullanıcılar farklı cihaz türlerine göre farklı deneyimler yaşamaktadır. Dolayısıyla web tasarımcısı, her cihazın kendine özel avantajlarına ve kısıtlamalarına uymak durumundadır.

Farklı cihaz türleri ekran boyutları ve çözünürlük bakımından birbirinden farklıdır. Bunun yanında mobil cihazlar genellikle daha yavaş internet bağlantılarına sahiptir. Bu nedenle mobil öncelikli tasarımda web sayfasının yükleme hızı optimize edilmelidir. Gereksiz büyük görseller veya web sayfasında mobil cihazlarda gerek duyulmayan betikler (javascript kodlar vs.) kullanılmamalıdır.

Birçok web tarayıcı modern tasarım tekniklerini desteklese de farklı tarayıcılarda aynı tasarım farklı şekilde görüntülenebilmektedir. Bütün tarayıcılarda web sayfasının tutarlı bir şekilde görüntülendiğini

test etmek için tarayıcı uyumluluđuna dikkat etmek gerekmektedir.

Web sitesinde sunulacak içerik çok fazla etkileşim gerektiriyorsa ve karmaşık içerikleri barındırıyorsa bu durumda mobil öncelikli tasarım yapmak yerine mobil uygulama yapılması düşünülebilir.

KONTEYNER SORGULARI

Medya sorguları ekran çözünürlüğüne göre bir sorgulama yaparak stil uygulamaktadır. Ancak bazı durumlarda ekran çözünürlüğüne göre değil de kapsayıcı boyutuna göre stil kuralları uygulamak isteyebiliriz. Bu durumda konteyner sorguları kullanılmaktadır.

Konteyner sorguları medya sorguları gibi CSS dilinde duyarlı web tasarımı oluşturmak için kullanılan bir araçtır. Her iki sorgu türünde de sayfanın genişliğine bağlı olarak stiller uygulanmaktadır. Medya sorguları sayfanın tamamının genişliğine göre stil uygularken konteyner sorguları kapsayıcı elementin genişliğine göre stil uygular. Bu nedenle konteyner sorguları medya sorgularından daha fazla esneklik sağlar.

Konteyner sorguları eklemek için “@container” kuralı kullanılmaktadır. Genel söz dizimi aşağıdaki gibidir.

```
@container {  
  
}
```

Burada “@container” ifadesi bir konteyner kuralı yazacağımızı belirtir. Daha sonra parantez içerisine yazılacak şart ifadesi sorgulamak istediğimiz şartı belirler. Örneğin kapsayıcı elemanın 400px değerinden büyük olduğu durumlarda @container (width > 400px) { ... } ifadesi yazılabilir.

Bir web sayfası içerisinde bir çok farklı kapsayıcı içerik bulunabilir. Medya sorguları ile tüm sayfa genişliğini sorgulamak yerine sadece genişliği değişen kapsayıcı öğeye göre yazı, görsel boyutu gibi içeriklerin içinde bulunduğu kapsayıcıya göre otomatik olarak birçok durumda web tasarımcısına kolaylık ve esneklik sağlamaktadır.

MOBİL ve MASAÜSTÜ CİHAZLARIN GENEL ÖZELLİKLERİ

Mobil ve masaüstü cihazların kullanımı, web tasarımında iki farklı yaklaşımı beraberinde getirir. Mobil cihazlar genellikle küçük ekranlara, dokunmatik kontrollere ve mobil bağlantı seçeneklerine sahipken, masaüstü cihazlar büyük ekranlar, klavye ve fare gibi giriş araçları ve daha güçlü donanım özellikleri ile kullanıcı deneyimini farklılaştırır. Masaüstü cihazlar, özellikle iş ve eğlence için daha kapsamlı görevleri yerine getirmeye uygunken, mobil cihazlar hızlı bilgi erişimi ve taşınabilirlik için tercih edilir. Web tasarımcıları ve geliştiriciler, kullanıcıların cihazlarla etkileşim biçimini ve bağlantı türlerini anlamalı, tasarımlarını bu özelliklere göre optimize etmeli ve her iki platform türünün kullanım amacını göz önünde bulundurarak esnek, kullanışlı ve erişilebilir deneyimler yaratmalıdır. Bu dengeli yaklaşım, her türlü cihazda kullanıcı ihtiyaçlarını karşılamayı hedefler ve modern web tasarımının temelini oluşturur.

MOBİL ÖNCELİKLİ TASARIM

Mobil öncelikli tasarım, akıllı telefonların yaygın kullanımı ile giderek önem kazanan bir web tasarım yaklaşımıdır. Bu yaklaşım, kullanıcıların mobil cihazlarla etkileşim biçimlerine ve beklentilerine dayanır. Mobil tasarım, sınırlı ekran boyutlarına rağmen kullanıcı arayüzünün basit ve etkili olmasını gerektirir, çünkü kullanıcıların kararları sunulan deneyimin kalitesine bağlı olarak hızla değişebilir. Ayrıca, mobil kullanıcıların düşük toleransı ve dikkat dağınıcı çoklu görevleri, tasarımcıların hızlı yükleme süreleri ve sorunsuz bir kullanıcı deneyimi sağlamak için sıkı kodlama ve sunucu yanıtları gibi performans optimizasyonlarına odaklanmalarını gerektirir. Mobil cihaz kullanıcıları genellikle kısa oturma süreleri yaşadığından, tasarımcılar ve geliştiriciler, bu kısa oturumları destekleyecek şekilde etkileşimleri optimize etmeli, böylece kullanıcılar hızlı bir şekilde bilgi edinebilir ve gerektiğinde etkinliklere ara verebilir. Mobil öncelikli tasarım, web sitelerinin artan mobil kullanımına uyum sağlamasını, böylece uzun vadeli başarıyı desteklemesini sağlar ve sitelerin geniş bir cihaz yelpazesinde tutarlı bir deneyim sunmasına olanak tanır. Bu yaklaşım aynı zamanda SEO için de önem taşır, çünkü arama motorları hızlı yüklenen ve mobil cihazlara uyumlu siteleri tercih eder. Özetle, mobil öncelikli tasarım, çağımızın dijital gerekliliklerini karşılayan esnek ve kullanıcı dostu bir web deneyimi sunmak için hayati bir yaklaşımdır.

Mobil Öncelikli Tasarımın Avantajları ve Dezavantajları

Mobil öncelikli tasarım, artan mobil kullanım ve arama motorlarının mobil uyumluluğa verdiği önem nedeniyle popülerleşmiştir. Hızlı yükleme süreleri ve mobil uyumluluk, sitelerin SEO performansını iyileştirirken, engelli kullanıcılar için erişilebilirliği artırır. Ancak, mobil öncelikli sitelerin geliştirilmesi zaman alabilir ve eski tarayıcılarla uyumsuzluklar gösterebilir. Tasarımcıların ve geliştiricilerin, mobil ve masaüstü sürümleri güncel tutmak için ek çaba sarf etmeleri gerektiği, bu sürecin karmaşık ve zaman alıcı olabileceği unutulmamalıdır.

Adım Adım Mobil Öncelikli Tasarım

Mobil öncelikli tasarım, web sitelerinin artan mobil kullanımına uyum sağlamasını gerektirir ve etkili bir kullanıcı deneyimi sunmak için net ve basit bir arayüz, hızlı yükleme süreleri ve dokunmatik duyarlılık esastır. Tasarımlar, farklı ekran boyutlarına uyum sağlayacak şekilde duyarlı olmalı, sezgisel gezinme sağlamalı ve cihaz özelliklerini entegre etmelidir. Beyaz alan kullanımı ve okunabilir yazı tipleri önemliken, kapsamlı cihazlar arası testlerle uyumluluk sağlanmalıdır. Bu stratejiler, kullanıcı memnuniyetini artırmak ve sitenin başarısını desteklemek için kritik öneme sahiptir.

MASAÜSTÜ ÖNCELİKLİ TASARIM

Masaüstü öncelikli tasarım, büyük ekranlara ve güçlü donanım özelliklerine sahip masaüstü ve dizüstü bilgisayarlar için geliştirilmiş bir web tasarım yaklaşımıdır. Bu yaklaşım, kullanıcıların geniş ekranlarda rahat gezinmelerini, detaylı grafikler ve medya öğelerini etkili bir şekilde görüntülemelerini sağlar ve zengin multimedya içeriği ve etkileşimli özellikler sunar. İşaretçi aygıtlarının sağladığı hassasiyetle, hover efektleri ve animasyonlar gibi ileri düzey kullanıcı etkileşimleri mümkün hale gelir, bu da kullanıcı deneyimini daha sürükleyici ve etkileşimli kılar. Masaüstü tasarım, büyük görsellerin ve video gibi medya öğelerinin kullanımını desteklerken, mobil kullanımın artışıyla birlikte duyarlı tasarımın önemi artmıştır. Tasarımcılar, kullanıcı ihtiyaçlarını her türlü cihazda karşılamak ve modern web standartlarına uyum sağlamak için mobil uyumluluğu da göz önünde bulundurmalıdır. Bu, kullanıcıların cihaz türü ne olursa olsun kaliteli bir deneyim yaşamalarını

garanti altına alır ve sitelerin uzun vadeli başarısını destekler.

Masaüstü Öncelikli Tasarımın Avantajları ve Dezavantajları

Masaüstü öncelikli tasarım, büyük ekran ve güçlü donanım avantajlarından yararlanarak zengin ve detaylı kullanıcı deneyimleri sunar. Özellikle masaüstü kullanımın yaygın olduğu iş kolları için idealdir ve karmaşık grafiklerle etkileşimi güçlendirir. Ancak, mobil kullanımın artışı göz önünde bulundurulduğunda, mobil cihazlar için optimize edilmemiş olması önemli bir dezavantajdır. Masaüstü sitelerin mobil cihazlarda yavaş yüklenmesi ve SEO sıralamasında düşük performans göstermesi, bu yaklaşımın sınırlamaları arasındadır, bu da mobil kullanıcı deneyimini ve arama motoru görünürlüğünü olumsuz etkileyebilir.

MOBİL YA DA MASAÜSTÜ TASARIMI SEÇERKEN DİKKAT EDİLMESİ GEREKEN UNSURLAR

Mobil ve masaüstü tasarım kararları alınırken, hedef kitlenin davranışları ve cihaz kullanım alışkanlıkları önemlidir. Mobil öncelikli tasarım, artan mobil kullanımı nedeniyle tercih edilebilir ve genellikle daha düşük maliyetlidir, performans artışı sağlar ve farklı cihazlara uyumluluk sunar. Masaüstü tasarım ise, zengin multimedya ve interaktif içerikler için uygundur. Bütçe kısıtlamaları ve sürdürülebilirlik gibi faktörler de bu seçimi etkiler. Sonuç olarak, her iki platform için optimize edilmiş, yanıt veren bir tasarım, geniş bir kullanıcı kitlesine hitap edebilir ve uzun vadede maliyet etkinliği sağlayabilir. Tasarımcılar, en uygun tasarım stratejisini belirlerken bu faktörleri dikkate almalıdır.

Günümüzde kullanılan Bootstrap sayfa düzeni, flex ve grid özelliklerinin kombinasyonu ile özelleştirilmiş yapılar sunar. Bu yapılar, Web Tasarımın Temelleri bölüm 9'da ele alınan CSS FlexBox özelliklerini daha ileri seviyede işleyerek, kendi sayfa yerleşimlerinizi akışkan, mobil uyumlu ve duyarlı bir şekilde tasarlamaya olanak tanır.

Web sayfalarında kullanılabilen Flex ve Grid yapıları, içeriklerin dikey, yatay veya farklı çözünürlüklerdeki görünümleri için uygun düzenler oluşturmanıza olanak sağlar. Flex düzeni, bir nesnenin içindeki çocuk nesnelerin yatay veya dikey yerleşimini düzenleyen bir CSS özelliğidir. Bu özellik, seçilen bir nesne içindeki nesnelere esnek bir şekilde yerleştirmenize olanak tanır.

Flex Özellikleri

Flex düzenini kullanarak, sayfa veya kapsayıcının boyutlarıyla oynayarak iç nesnelerin düzenini değiştirebilirsiniz. Örneğin, flex-direction özelliği ile öğelerin sıralanma yönünü belirleyebilirsiniz. Flex-wrap özelliği, öğelerin bir satıra sığmadığı durumlarda nasıl davranacaklarını kontrol eder. Justify-content, öğelerin yatay yönde nasıl hizalandığını belirler. Align-content, satırlar arasındaki boşluğun nasıl dağıtılacağını kontrol eder.

Align-items özelliği, kapsayıcı içindeki öğelerin eksen boyunca nasıl hizalanacağını belirler.

Hizalama yönü (flex-direction) seçimine bağlı olarak, justify-content, align-content ve align-items özelliklerinin etkisi değişir. Bu özellikler, duyarlı tasarımlarda ve içerik hizalamada kullanışlıdır.

Bu flex özellikleri kullanılarak, mobil menülerden sütunlu düzenlere kadar çeşitli tasarımlar oluşturabilirsiniz. Özelliklerin kabul ettiği değerler arasında yapılan seçimler, tasarımınızın istenen sonuca ulaşmasına yardımcı olacaktır.

Çocuk Nesnelere Flex Kullanımı

Flex özelliği, web tasarımında kapsayıcının altında tanımlanan çocuk nesnelerin esnek bir şekilde düzenlenmesini sağlayan güçlü bir CSS özelliğidir. Bu özellik, kapsayıcının içindeki çocuk nesnelerin boyutunu, sıralamasını ve hizalamasını kontrol etmek için kullanılır. Flex özelliği, "flex-grow", "flex-shrink" ve "flex-basis" alt özellikleri içerir, bu alt özellikler tek bir satırda kısa bir CSS koduyla da kullanılabilir.

Flex özelliğinin temel alt özellikleri şunlardır:

1. flex-grow: Bu özellik, bir flex öğesinin nasıl büyüyeceğini belirler. Değeri genellikle bir tam sayı veya sıfır olabilir. Varsayılan değeri 0'dır. Eğer bir flex öğesi "flex-grow" değeri 0 ise, öğe boyutu büyümeyecektir. Eğer değer 1 ise, öğe diğer flex öğelerine göre daha fazla büyümeyecektir.
2. flex-shrink: Bu özellik, bir flex öğesinin nasıl küçüleceğini belirler. Değeri genellikle bir tam sayı veya sıfır olabilir. Varsayılan değeri 1'dir. Eğer bir flex öğesi "flex-shrink" değeri 0 ise, öğe boyutu küçülmeyecektir. Eğer değer 1 ise, öğe diğer flex öğelerine göre daha fazla küçülecektir.
3. flex-basis: Bu özellik, bir flex öğesinin başlangıç boyutunu belirler. Değer genellikle bir uzunluk (örneğin, "px" veya "%") veya "auto" olabilir. Varsayılan değeri "auto" dur ve öğenin içeriğine bağlı olarak otomatik olarak boyutunu belirler.

Flex özelliğinin kısa CSS tanımlamaları şunlardır:

•"flex: none" veya "flex: 0 0 auto": Öğenin sabit bir boyuta sahip olmasını ve büyüme veya küçülmeyi kabul etmemesini sağlar.

•"flex: auto" veya "flex: 1 1 auto": Öğenin esnek bir boyuta sahip olmasını ve büyüme veya küçülmeyi kabul etmesini sağlar.

•"flex: 1" veya "flex: 1 1 0%": Öğenin esnek bir boyuta sahip olmasını, büyüme veya küçülmeyi kabul etmesini ve başlangıç boyutunun sıfır olmasını sağlar.

Kapsayıcı ve Çocuk Flexlerin Varsayılan Özellikleri

Eğer bir kapsayıcı nesne içinde çocuk nesnelere varsa ve sadece display: flex; komutu atanırsa, kapsayıcı ve çocuk nesnelere aşağıdaki varsayılan özellikler otomatik olarak atanmış kabul edilir:

Kapsayıcı için:

- flex-direction: row;
- flex-wrap: nowrap;
- justify-content: flex-start;
- align-content: stretch;
- align-items: stretch;

Bu ayarlar, kapsayıcı nesnenin soldan sağa hizalı bir satır gibi davranmasını, alt satıra düşmemesini,

yatayda sola hizalı olmasını ve çocuk nesnelere dikeyde yukarı ve aşağıya genişlemiş olarak hizalamasını sağlar.

Çocuk nesne için:

- flex: 0 1 auto;
- flex-grow: 0;
- flex-shrink: 1;
- flex-basis: auto
- order: 0;

Bu ayarlar, çocuk nesnelere genişliklerini paylaşma miktarlarının belirlenmediği, küçülme katsayısının 1 olduğu, başlangıç genişliklerinin otomatik olarak atandığı ve nesne sırasının değiştirilmediği varsayılan özellikleridir.

Yukarıdaki bilgiler ışığında, esnek bir web tasarımında flex özelliklerini kullanarak çeşitli düzenlemeler yapabilirsiniz. Flex özellikleri, özellikle responsive (duyarlı) tasarımlarda ekstra esneklik sağlar ve farklı ekran boyutlarına uyum sağlamak için ideal bir araçtır.

Mobilmenu ise içinde yazı karakterleri bulunan menü aç/kapat işini yapacak bir nesnedir. Sayfa küçüldüğünde gözükmeye için display özelliği block yapılmıştır. Pozisyonu sayfada diğer nesnelere konumlarından etkilenmemesi için position: absolute; olarak belirlenmiş ve sayfanın sağından right: 10px ile 10px uzaklaşacak şekilde ayarlanmıştır. Üzerine gelindiğinde fare simgesi işaretçi (cursor:pointer) olacak şekilde ayarlanmıştır. Arkaplan rengi ve iç boşluklar verilerek görünüm düzenlenmiştir.

```
.mobilmenu: focus + .kapsayici {
display: flex;
flex-direction: column;
align-items: flex-end;
position: absolute;
right: 10px;
top: 40px;
border: 0px;
}
.cocuk {
justify-content: right;
}
.kapsayici: hover {
display: flex;
flex-direction: column;
align-items: flex-end;
position: absolute;
right: 10px;
top: 40px;
border: 0px;
}
}
```

Mobilmenü'nün focus durumu, farenin bu nesneye tıkladığı veya TAB tuşu ile seçtiği yada odaklandığı durumları ifade eder. Dolayısı ile sayfamız küçükken, menü butonuna tıkladığımız esnadaki kapsayıcı özelliklerini tanımlayabilmek için .mobilmenu: focus + .kapsayici CSS tanımlamasını yaparak içeri dolduruyoruz.

Kapsayıcıya yine Flex özelliği atanmış fakat bu sefer dikey gözükmeye için flex-direction: column; yani yönü yukarıdan aşağıya olarak seçilmiş, align-items: flex-end; ile sağa doğru hizalanmıştır.

Aynı şekilde sayfa küçüldüğünde çocuk menüler de metinleri justify-content: right; ile sağa hizalanmıştır.

Son CSS tanımlaması da arayüzün daha kullanışlı olması için eklenmiştir. Fare ile çocuk alanları üzerinde gezerken bir linke tıkladığımız anda menü butonunun focus özelliğini kaybedeceğimiz için, focus ile yaptığımız kapsayıcı CSS tanımlamaları da yok olacaktır. Bunun önüne geçmek için kapsayıcının üzerinde gezinirken aynı CSS tanımlamalarını tekrardan tanımlıyoruz. Bunu .kapsayici: hover özelliğine aynı CSS tanımlamalarını koyarak sağlıyoruz.

Bu CSS tanımlamaları Tablo 3'te de gözüktüğü gibi geniş ekranda sayfa geniş dururken, dar ekranda menü gizlenerek menü butonu gözükmeye hale geliyor. Menü butonuna tıkladığımız anda kapsayıcıya flex özelliği atayarak dikey olarak gözükmeye sağlıyoruz.

Grid Layout

CSS Grid Layout, web sayfalarında karmaşık ve özelleştirilebilir düzenler oluşturmak için kullanılan etkili bir CSS özelliğidir. Bu özellik, sayfanın düzenini ve öğelerin yerleşimini tamamen kontrol etme imkanı sağlar. CSS Grid Layout, diğer düzen sistemlerine kıyasla daha dinamik ve esnek bir yapı sunar. Sayfanın tüm düzenini istenilen şekilde oluşturabilir, öğeleri tam kontrol edebilir ve özellikle karmaşık yapıdaki web sayfaları için anlaşılır bir düzen sağlar.

Duyarlı ve uyum sağlayabilen tasarımlar oluşturmada etkili olan CSS Grid Layout, sütun ve satırları istenildiği gibi düzenleyebilme özelliği sayesinde ekran boyutlarına göre kolayca yerleşimleri değiştirebilir. Ayrıca, grid şablonları kullanarak farklı sayfalarda hızlıca ve yeniden kullanılabilir şekilde gridler oluşturabilirsiniz. Bu özellik, tasarım sürecini hızlandırır ve kodu daha düzenli hale getirir.

CSS Grid Layout, iç içe layout'lar oluşturmanıza da izin verir. Bu özellik, özellikle karmaşık ve hiyerarşik düzenler oluşturmak isteyenler için kullanışlıdır. Ayrıca, kodunuzu daha anlaşılır ve izlenebilir hale getirir, çünkü daha az kod kullanarak grid oluşturabilir ve hata yapma olasılığını azaltabilirsiniz.

Modern web tarayıcılarında genellikle sorunsuz çalışan CSS Grid Layout, popüler tarayıcılar arasında Chrome, Firefox, Safari, Edge ve Opera gibi tarayıcılarda tam desteklenir. Ancak, bazı eski tarayıcılar ve sürümler tam uyumlu olmayabilir, bu nedenle dikkatli olunmalıdır.

Grid özelliğinin kullanabileceğimiz bazı alt komutları şunlardır: grid-template-columns, grid-template-rows ve gap (row-gap, column-gap). Bir öğeyi display: grid; CSS komutuyla grid kapsayıcısına dönüştürdüğümüzde, bu öğenin tüm doğrudan alt öğeleri grid hücrelerine dönüşür. Tüm doğrudan çocuklar artık grid öğeleridir ve display: grid komutuyla çocuk nesnelere tek bir sütun gibi listelenir. Grid yapısını elde etmek için grid-template-rows ve grid-template-columns özellikleri kullanılmalıdır.

Sütunların Düzenlenmesi

Grid sütunlarını belirleme konusunda grid-template-columns özelliği kullanılır. Bu özellik, bir grid kapsayıcının sütunlarının genişliklerini belirlemek için kullanılır. Değerler olarak piksel (px), yüzde (%) veya fraksiyon (fr) kullanılabilir. Piksel değerleriyle her sütunun genişliği belirlenebilir. Yüzde değerleriyle sütunların genişliği orantılı olarak belirlenebilir. Fraksiyon değerleri, grid kapsayıcısının kullanılabilir boş alanının fr katsayısına göre paylaştırılarak sütunların genişliklerini belirler. Ayrıca, daha karmaşık değerlerle otomatik oluşturma ve sınırlama yapma imkanı da vardır.

Satırların Düzenlenmesi

Grid satırlarını belirleme konusunda ise grid-template-rows özelliği kullanılır. Bu özellik, grid kapsayıcının satırlarının yüksekliklerini belirler. Piksel, yüzde veya fraksiyon (fr) değerleri kullanılarak her satırın yüksekliği belirlenebilir. auto değeri verilerek satırların içeriklerine göre otomatik boyutlanması da sağlanabilir.

Bu iki özellikle grid yapısının sütun ve satırları istenilen genişlik ve yüksekliklere göre özelleştirilebilir. Ancak, satır ve sütun genişliklerini otomatik olarak belirleme davranışı arasında bazı farklılıklar bulunmaktadır. Özellikle, auto değeri verilen satırların dikey ekseninde tüm sayfayı dolduramaması ve bu davranışın grid-template-columns için farklı olması dikkat çekicidir. Bu farklılıklar, kapsayıcının genişlik ve yüksekliğinin belirlenmesi durumuna bağlı olarak değişebilir.

Repet ve MinMax Komutları

Grid ölçülerini belirlemede repeat ve minmax komutları da kullanılmaktadır. Bu komutlar, grid hücrelerinin ölçülerini ve sayısını belirlemede esneklik sağlar. Örneğin, grid-template-columns: auto repeat(3, 40px) auto; komutuyla kapsayıcı içinde toplam 5 sütuna sahip bir grid oluşturulur.

Grid ölçülerinin belirlenmesinde repeat komutu, özellikle sütun ve satır sayılarını tek bir komutta

belirleme avantajı sunar. Benzer şekilde, minmax komutu da kullanılarak ölçü değerleri daha esnek bir şekilde belirlenebilir. Örneğin, minmax(100px, 200px) komutuyla bir hücrenin en az 100px ve en fazla 200px genişliğe sahip olacağı belirlenebilir.

Grid Hücrelerinin Detaylı Ayarlanması

Grid özellikleri, grid-column-start, grid-column-end, grid-row-start, ve grid-row-end gibi komutlarla daha detaylı özelleştirmelere olanak tanır. Bu komutlar, öğelerin grid üzerindeki konumlarını esnek bir şekilde belirlemek için kullanılır. Ayrıca, aralıkların belirlenmesi için gap özelliği kullanılabilir.

Geliştirici araçlarında, grid yapısını görselleştirmek için DevTools'un Layout sekmesi kullanılabilir. Bu sekme, grid veya flex kapsayıcılarının tanımlanmışsa, bu öğelerle ilgili görselleştirme araçlarına erişim sağlar. Bu araçlar, tasarımcılara görsellik sunarak web sitesi tasarımında kolaylık sağlar.

Grid Layout Şablonu Oluşturma

Grid Layout ile web sitelerinin ana şablonları oluşturulabilir. Bu şablonları oluştururken iç içe grid yapıları oluşturabilirsiniz. Oluşturduğunuz gridin çocuk hücrelerine css grid tanımlaması yaparak, onun da içinde çocuk hücreler için bir şablon veya css tanımlamaları oluşturabilirsiniz.

Grid Layout'un "grid-template" özelliği, satırları ve sütunları belirlemek için kullanılır. Bu özellik, "grid-template-rows" ve "grid-template-columns" özelliklerini birleştirir. Çocuk nesnelere "grid-area" özelliği atanarak etiket isimleri yazılır. Daha sonra grid-template css özelliğinin içine bu alanları bir tablo hücresi gibi yazarak tasarım şablonu oluşturulur.

Grid-template içinde "." (nokta) simgesi kullanarak boş hücreler oluşturulabilir. Ancak, hatalı şablon tanımlamaları sayfa bozulmalarına yol açabilir. Grid-template şablonunda kopuk alanlar, aynı etiketlerin düzenli olmaması, bağlantısız alanlar ve dikdörtgensel bütünlüğün sağlanamaması gibi hatalar şablonun çalışmamasına sebebiyet verecektir.

Grid şablonları oluşturulurken, alan etiketlerinin dikdörtgensel bir bütünlük oluşturması önemlidir. Aynı etiketler arasında boşluk veya başka etiketler bulunmamalıdır. Var olmayan bir alan etiketi kullanıldığında "." (nokta) ile aynı etki elde edilebilir. Ayrıca, grid-template: a/b; tanımlamasındaki b alanındaki ölçülerin belirlenmesi önemlidir, çünkü belirlenmezse varsayılan olarak "auto" atanacaktır.

Grid-template içinde yatay ölçülerin belirlenmesi, sütun genişliklerinin otomatik belirlenmesini önler. Tanımlanan sütun genişlikleri ile istenilen şablon elde edilir. Yatay ölçü belirlenmemişse, sütunlar genişliği "auto" olarak atanır. Bu durum, sayfanın genişliğine bağlı olarak otomatik genişleme sağlar.

Dijital dünyada, web sitelerinin farklı cihazlarda ve ekran boyutlarında etkili bir şekilde görüntülenmesi büyük önem taşımaktadır. Duyarlı web tasarımı, mobil cihazlarda ve bilgisayar ekranlarında aynı kaliteyi deneyimlemeyi sağlamak amacıyla geliştirilmiştir. Bu bölümde, duyarlı metin, görüntü, tablo ve video kullanımının mobil uyumlu tasarım içindeki kilit rolü anlatılmaktadır.

DUYARLI METİN

Duyarlı metin kullanımı, mobil cihazlar ve farklı ekran boyutları için optimize edilmiş web tasarımlarının vazgeçilmez bir parçasıdır. Kullanıcılar, web sitelerinde hızlı ve kolay bir şekilde bilgiye erişmeyi beklerler. Bu nedenle, duyarlı metin tasarımı, kullanıcı deneyimini geliştirmek için kritik bir unsurdur. Duyarlı metin, farklı ekran boyutlarına ve çözünürlüklere uyumlu olan metindir. Duyarlı metin oluşturmak için kullanılan CSS teknikleri şunlar olabilir:

Yazı tipi boyutunu dinamik olarak ayarlama: Ekran boyutuna göre yazı tipi boyutunu ayarlayarak metnin okunabilirliğini artırır.

Metin hizalamasını dinamik olarak ayarlama: Ekran boyutuna göre metin hizalamasını ayarlamak, metnin görünümünü iyileştirir.

Metin aralıklarını dinamik olarak ayarlama: Ekran boyutuna göre metin aralıklarını ayarlamak, metnin okunabilirliğini artırır.

Duyarlı Görüntü

Mobil cihazların yaygın kullanımı, web tasarımında duyarlılığın önemini artırmıştır. Duyarlı görüntüler, farklı ekran boyutlarına ve çözünürlüklerine uyum sağlamak amacıyla geliştirilmiş bir tekniktir. Bu bölümde, duyarlı resim oluşturmak için kullanılan CSS teknikleri ve farklı cihazlara özgü görüntü formatlarının önemi ele alınacaktır.

Duyarlı görüntü oluşturmak için kullanılan CSS teknikleri şunlar olabilir:

Resim Boyutunu Dinamik Olarak Ayarlama: Resim boyutu, ekran boyutuna göre dinamik olarak ayarlanarak görüntünün daha iyi bir şekilde görünmesi sağlanır.

Resim Oranını Dinamik Olarak Ayarlama: Resim oranı, ekran boyutuna göre dinamik olarak ayarlanarak oranın korunması ve görüntünün doğru bir şekilde görüntülenmesi sağlanır.

Resim Konumunu Dinamik Olarak Ayarlama: Resim konumu, ekran boyutuna göre ayarlanarak görüntünün belirli bir konumda kalması sağlanır.

Duyarlı görüntü için farklı cihazlarda kullanılabilen görüntü biçimleri şunlardır:

JPEG (Joint Photographic Experts Group): Karmaşık ayrıntılara sahip fotoğraf ve görüntüler için uygun. Kayıplı sıkıştırma kullanır, düşük dosya boyutları elde edilir.

PNG (Portable Network Graphics): Orijinal görüntü kalitesini korur, kayıpsız sıkıştırma sağlar. JPEG'e göre daha büyük dosya boyutlarına sahiptir.

WebP: Google tarafından geliştirilen, üstün sıkıştırma sunar. Daha küçük dosya boyutları sağlar, modern tarayıcılar tarafından desteklenir.

Duyarlı görüntü için farklı cihazlarda farklı görüntü biçimleri kullanmanın yararları:

Optimize Edilmiş Performans: Her cihaz için en uygun görsel formatın kullanılması sayfa yükleme sürelerini azaltır.

Daha Az Veri Tüketimi: Daha küçük dosya boyutları, özellikle mobil kullanıcılar için daha düşük veri tüketimine neden olur.

Geliştirilmiş Kullanıcı Deneyimi: Daha hızlı yükleme süreleri ve tutarlı görüntü kalitesi, olumlu bir kullanıcı deneyimine katkıda bulunur.

Duyarlı Tablo

Web tasarımının karmaşıklığı arttıkça, farklı cihazlarda ve ekran boyutlarında mükemmel bir kullanıcı deneyimi sağlamak giderek daha zorlu hale gelmektedir. Özellikle veri tablolarının olduğu durumlarda, duyarlılık esastır. Duyarlı tablolar, farklı ekran boyutlarına ve çözünürlüklere uyumlu olan tablolardır.

Duyarlı tablo oluşturmak, kullanıcı deneyimini iyileştirmek ve erişilebilirliği sağlamak için önemlidir. Bu bölümde, "Duyarlı Web Tasarımında Duyarlı Tablo" konusuna odaklanarak, tabloların nasıl optimize edileceğini, HTML ve CSS kullanarak nasıl duyarlılık sağlanacağını anlayacağız.

Duyarlı tablo oluşturmak için CSS teknikleri kullanılır. Bu teknikler, tablonun boyutunu, genişliğini, sütun genişliğini ve satır yüksekliğini dinamik olarak ayarlayarak tabloyu farklı ekran boyutlarına göre uyarlamaya olanak tanır.

Tabloyu kaydırma çubuğu çıkabilen bir alana koyma

Tabloyu mobil cihazlarda daha kullanıcı dostu hale getirmek için yatay kaydırma çubuğu eklemek önemlidir. Bu, kullanıcıların tablo içeriğini daha iyi görmelerini sağlar. CSS kodunda, “overflow-x: auto;” özelliği, tablo içeriği geniş ekrana sığmadığında yatay kaydırma çubuğunu otomatik olarak ekler.

Duyarlı Video

Günümüzde, video içerikleri internetin vazgeçilmez bir parçası haline gelmiştir ve kullanıcılar farklı cihazlar aracılığıyla bu içeriklere erişim sağlamaktadır. Ancak, çeşitli ekran boyutları, çözünürlükler ve bağlantı hızları göz önüne alındığında, duyarlılık, web tasarımında video içeriklerini optimize etmek için kritik bir unsurdur. Bu bölümde, "Duyarlı Web Tasarımında Duyarlı Video" konusuna odaklanarak, HTML5 video elementini kullanarak ve CSS ile destekleyerek video içeriklerinin farklı ekranlarda nasıl etkili bir şekilde görüntülenebileceğini anlamaya çalışacağız.

Video içerikleri, kullanıcıları etkilemek, bilgiyi daha çekici hale getirmek ve marka iletişimini güçlendirmek adına güçlü bir araçtır. Ancak, web tasarımında video içeriklerini kullanırken, kullanıcıların farklı cihazlardan erişim sağlamasını ve optimal bir deneyim yaşamasını sağlamak önemlidir. Duyarlı video tasarımı, bu ihtiyacı karşılamak ve her türlü ekran boyutunda mükemmel görüntüleme sunmak adına kilit bir stratejidir. Duyarlı video oluşturmak için kullanılan CSS teknikleri video boyutunu, video oranını ve video konumunu dinamik olarak ayarlama olarak sıralanabilir.

Video boyutunu dinamik olarak ayarlama: Video boyutu, ekran boyutuna göre ayarlanarak videonun görünümünü iyileştirmek için kullanılır.

Video oranını dinamik olarak ayarlama: Video oranı, ekran boyutuna göre ayarlanarak videonun görünümünü iyileştirmek için kullanılır.

Video konumunu dinamik olarak ayarlama: Video konumu, ekran boyutuna göre ayarlanarak videonun görünümünü iyileştirmek için kullanılır.

Görünüm Penceresi, bir web tarayıcısında görülen alanı temsil eder. UI, menü çubuğu vb. hariç tutulduğunda genellikle tarayıcı penceresi ile aynıdır. Bu, görüntülediğiniz tarayıcıda görüntülediğiniz belgenin bir kısmıdır.

Web sayfalarındaki makaleler genellikle çok uzun olabilir. Görünüm penceresi, tarayıcıda o anda görünen her şeyi temsil eder. Görünüm penceresinin boyutu, ekranın boyutuna, tarayıcının tam ekran modunda olup olmadığına ve kullanıcının yakınlaştırmayı yakınlaştırmadığına bağlıdır. Görünüm penceresi dışındaki içerik, sayfa aşağıya kaydırılmadıkça genellikle görünmez.

Uygulamaların tam ekran olmadığı daha büyük monitörlerde, görünüm penceresi tarayıcı penceresi boyutundadır.

Çoğu mobil cihazda ve tarayıcı tam ekran modundayken, görünüm penceresi tüm ekrandır.

Tam ekran modunda görünüm penceresi cihaz ekranıdır, pencere tarayıcı penceresidir, bu da görünüm penceresinden büyük veya küçük olabilir ve belge web sitesidir, görünüm penceresinden çok daha uzun veya geniş olabilir. Özetle, görünüm penceresi temelde web sayfasının web tarayıcısında o anda görünen bölümüdür.

Bir sayfa daha küçük ekranlı cihazlar için duyarlı hale getirilmediğinde, kötü görünmekte ve hatta daha küçük ekranlarda bozulmaktadır. Bu sorunu çözmek için görünüm penceresini kontrol edecek duyarlı bir etiket eklenmelidir. Bu etiket ilk olarak Apple tarafından Safari için tanıtılmıştır.

Dinamik görünüm penceresi birimleri, görünüm penceresinin boyutuna göre değişen birimlerdir. Bu birimler, dinamik olarak genişleyen ve daralan arayüzler için kullanışlıdır.

Meta etiketi, HTML belgesinde head etiketi içine eklenmelidir. Bir duyarlılık etiketinin aşağıdaki özellikleri bulunmaktadır:

width: Cihazın sanal görünüm penceresinin genişliğini tanımlamak için kullanılır. Bu özelliğe "device-width" değerinin yanı sıra 1 ile 10000 arası değer tanımlanabilir. Ancak "device-width" değeri dışında elle değer tanımlanması önerilmemektedir.

height: Cihazın sanal görünüm penceresinin yüksekliğini tanımlamak için kullanılır. Bu özelliğe "device-height" değerinin yanı sıra 1 ile 10000 arası değer tanımlanabilir. Ancak "device-width" değeri dışında elle değer tanımlanması önerilmemektedir.

initial-scale: Sayfa ilk kez ziyaret edildiğindeki yakınlaştırma düzeyini tanımlamak için kullanılır. 0.1 ile 10 arası değer tanımlanabilir. Varsayılan değeri 1'dir.

minimum-scale: Bir kullanıcının sayfayı ne kadar küçültebileceği minimum yakınlaştırma düzeyini tanımlamak için kullanılır. 0.1 ile 10 arası değer tanımlanabilir. Varsayılan değeri 0.1'dir.

maximum-scale: Bir kullanıcının sayfayı ne kadar büyüttebileceği maksimum yakınlaştırma düzeyini tanımlamak için kullanılır. 0.1 ile 10 arası değer tanımlanabilir. Varsayılan değeri 10'dur.

Erişilebilirlik açısından 3 düşük değerler kullanılmamalıdır.

user-scalable: Cihazın sayfayı yakınlaştırmaya veya uzaklaştırmaya iznini tanımlamak için kullanılır. Kullanılabilen değerler "0", "1", "yes" ve "no"dur. Varsayılan değeri 1 dir ve "yes" ile aynı anlama gelmekte, ölçeklendirmeye izin vermek demektir. Değeri "0" olarak tanımlamak erişilebilirlik açısından önerilmemektedir.

Bir web sitesini ziyaret ettiğinizde, web sayfası yüklendikten sonra sayfanın tüm içeriğini görünmemektedir. Bunun yerine, tarayıcı size sayfanın bir bölümünü görebileceğiniz bir görünüm sunmaktadır. Bu görünüm penceresi Yerleşim Görünüm Penceresi olarak tanımlanmaktadır. Bir sayfanın içeriği çok büyüdüğünde sayfa içeriğine erişebilmek için tarayıcı bir kaydırma mekanizması oluşturmaktadır.

Öğeleri " position: fixed " kullanarak konumlandırırken, bu öğeler Yerleşim Görünüm Penceresine göre yerleştirilecektir. Siz bir sayfayı aşağı kaydırırken Yerleşim Görünümü yerinde kaldığı için, " position: fixed " kullanan öğeler de yerinde kalacaktır.

Yerleşim Görünüm Penceresine ek olarak, tarayıcı aynı zamanda bir Görsel Görünüm Penceresi de sunmaktadır. Görünüm penceresinin o anda görünür olan kısmını temsil eder. Yakınlaştırma düzeyi 1'de bu Görsel Görünüm Penceresi, Yerleşim Görünüm Penceresi ile aynı büyüklüktedir.

Parmakla yakınlaştırmak, Görsel Görünüm Penceresi'nin boyutu Yerleşim Görünüm Penceresinden daha küçük hale getirilir.

Web sayfalarının farklı ekran boyutlarına uyum sağlamak için kullanılan üç farklı görünüm penceresi birimi incelenmektedir: Büyük, Küçük ve Dinamik Görünüm. Geleneksel görünüm penceresi birimleri

(vh ve vw) en yaygın ve tarayıcı uyumluluđu en yüksek olanlardır, sayfa elementlerini görünüm penceresi boyutunun yüzdesi olarak tanımlarlar. Dinamik görünüm penceresi birimleri, görünüm penceresinin boyutuna göre deđişen birimlerdir ve genişleyen veya daralan arayüzler için uygundur. Bu birimler, web sayfalarında farklı ekran boyutlarına uyumlu düzenler oluşturmak için kullanılır; büyük görünüm penceresi birimleri geniş ekranlar için idealdir, küçük görünüm penceresi birimleri ise mobil cihazlar veya küçük ekranlar için küçülen elementler oluşturur.

Büyük, Küçük ve Dinamik Görünüm penceresi birimlerini kullanarak, web sayfalarınızın farklı ekran boyutlarına uyum sağlamasını ve kullanıcı deneyimini iyileştirmeyi sağlayabilirsiniz. CSS'te görünüm penceresi birimleri kullanarak, web sayfalarınızı farklı ekran boyutlarına göre optimize edebilir ve kullanıcı deneyimini artırabilirsiniz. Bu birimler dinamik tasarımlar için farklı seçenekler sunar. Hangi birimin size en uygun olduğunu belirlemek için projenizin gereksinimlerini göz önünde bulundurmalı ve hedef kitlenizin kullandığı ekran boyutlarını dikkate almalısınız. Unutmayın, tarayıcı uyumluluđunu test etmek ve web sayfanızın tüm kullanıcılar tarafından düzgün bir şekilde görüntülenmesini sağlamak önemlidir. Bu nedenle, görünüm penceresi birimlerini kullanırken dikkatli olmalı ve farklı ekran boyutlarında kapsamlı testler yapmalısınız. Görünüm penceresi birimleri, web sayfaları için esnek ve dinamik tasarımlar oluşturmanızı kolaylaştıran güçlü araçlardır. Bu birimleri kullanarak, web sitelerinizin farklı ekran boyutlarında kusursuz görünmesini ve tüm kullanıcılar için keyifli bir deneyim sunmasını sağlayabilirsiniz.

Bootstrap nedir?

Bootstrap, web geliştirme dünyasında oldukça popüler bir açık kaynaklı framework'tür. Bootstrap, kullanıcı dostu, modern ve duyarlı web siteleri ve uygulamalar oluşturmak için kullanılan bir arayüz kütüphanesidir. İlk olarak Twitter tarafından geliştirilen bu framework, HTML, CSS ve JavaScript kullanılarak web geliştirme süreçlerini büyük ölçüde kolaylaştırmaktadır.

Framework Nedir?

Framework, yazılım geliştirmenin temel yapısını ve işlevselliğini sağlayan bir yapıdır. Web geliştirme bağlamında, bir framework, belirli görevleri yerine getirmek için kullanılan kod parçalarını içeren bir koleksiyon olarak düşünülebilir. Framework'ler, geliştiricilere işlerini hızlandırma, kod tekrarını azaltma ve projeleri daha düzenli bir şekilde yönetme olanağı sunar. Bu, web geliştirme süreçlerini daha etkili ve verimli hale getirir.

Bootstrap Avantajları

Hızlı ve Kolay Başlangıç: Bootstrap, temel HTML, CSS ve JavaScript bileşenlerini içerir, bu da yeni web projelerine hızlı bir başlangıç yapmanıza yardımcı olur. Bu, aynı kodları tekrar yazma ihtiyacını azaltır.

Duyarlı Tasarım: Duyarlı tasarım ilkeleri, bootstrap ile desteklenir. Bu, web sitenizin tablet, tablet ve mobil cihazlarda daha iyi görünmesini sağlar.

Hazır Bileşenler: Bootstrap, kullanıcı arabirimini oluşturmak için kullanabileceğiniz hazır bileşenlerden oluşan bir paket içerir. Bunlar modallar, etiketler, gezinme menüleri, formlar, düğmeler ve daha fazlasını içerir.

Daha iyi kullanılabilirlik: Bootstrap, kullanılabilirlik ile ilgili sorunları ele alır. Bileşenler ve stil kuralları, web sitenizin veya uygulamanızın kullanımını kolaylaştırır.

Kapsamlı Belgeleme: Bootstrap için ayrıntılı bir rapor sağlar. Bu, geliştiriciler için kullanım kılavuzları sağlar.

Topluluk Desteği: Bootstrap'ın büyük bir kullanıcı topluluğu vardır ve aktiftir. Bu topluluk, soruları yanıtlamak, sorunları çözmek ve yeni şeyler yapmak için bir araya gelir.

Esneklik ve Özelleştirme: Bootstrap ile özelleştirme yapabilirsiniz. İhtiyaçlarınıza göre bileşenleri yapılandırırken LESS veya SASS gibi önışlemcileri kullanabilirsiniz.

Çapraz Tarayıcı Uyumluluğu: Bootstrap, Chrome, Firefox ve Internet Explorer gibi çeşitli tarayıcılarda sorunsuz çalışacak şekilde test edilir.

Verimlilik: Stil kuralları ve hazır bileşenler, geliştirmeyi hızlandırır ve kodun yeniden kullanılabilir olmasını sağlar.

Ücretsiz ve Açık Kaynaklı: Bootstrap ücretsiz ve açık kaynaklıdır, bu nedenle hem kişisel hem de işletme projelerinde kullanılabilir.

Bu faydalar, Bootstrap'in web geliştirme süreçlerine nasıl yardımcı olduğunu ve neden birçok geliştirici tarafından tercih edildiği gerçeğini ortaya koymaktadır.

Bootstrap Sürümleri

Bootstrap 1 (Ağustos 2011): Temel CSS ve bileşenleri içeren ilk Bootstrap sürümüdür.

Bootstrap 2 (Şubat 2012): İkinci sürüm, IE7 desteği sağlar ve ek bileşenler ve özellikler ekler.

Bootstrap 3 (Ağustos 2013): Yeni bileşenler, mobil duyarlılık ve özelleştirilebilirlik özellikleri bu sürümde mevcuttur.

Bootstrap 4 (Aralık 2015): Bootstrap 4, Flexbox tabanlı bir ızgara sistemini benimser ve daha fazla bileşen ve özelleştirme seçeneği ile daha çağdaş bir görünüm ve his sağlar.
Bootstrap 5 (Kasım 2020): Bootstrap 5, önceki sürümlerden bazı bileşenleri ve jQuery bağımlılığını kaldırarak daha fazla CSS özelleştirme yeteneği sağlar ve çağdaş web geliştirme standartlarını karşılar.

Gerekli Araçlar ve Kaynaklar

Metin Düzenleyici veya Kod Düzenleyici: Web geliştirme projeleri için her ikisine de ihtiyacınız vardır. Bu araçlar, HTML, CSS ve JavaScript kodlarını düzenlemenize olanak tanır. Atom, Sublime Text ve Visual Studio Code gibi ücretsiz ve açık kaynaklı düzenleyicileri örnek olarak kullanabilirsiniz.

HTML, CSS ve JavaScript Bilgisi: Bootstrap, temel olarak HTML, CSS ve JavaScript kullanılarak oluşturulur. Sonuç olarak, bu temel web teknolojilerini anlamak ve kullanmak hayati önem taşır. CSS stil ve görünümü düzenlerken, HTML web sayfasının yapısal bileşenlerini belirler. JavaScript, web sayfalarına daha fazla işlevsellik ve etkileşim getirir. Bu dilleri kullanarak önceden oluşturulmuş bileşenleri uygulamak ve özelleştirmek için Bootstrap'ın tasarımını temel olarak bu dilleri kullanmayı amaçlamaktadır.

Bootstrap Kütüphanesi: Bootstrap'ü kullanabilmeniz için, çerçevenin resmî web sitesinden veya GitHub deposundan en son sürümünü indirmeniz gerekir. Kütüphane dosyalarını indirdikten sonra projenizin kök dizinine ekleyin.

Web Tarayıcısı: Projelerinizi geliştirirken ve test ederken birden fazla web tarayıcısı kullanmanız gerekir. Web sitenizi ve uygulamalarınızı farklı tarayıcılarda görüntülemek için Chrome, Firefox, Safari ve Edge gibi çeşitli tarayıcıları kullanabilirsiniz.

Dökümantasyon ve Örnekler: Bootstrap'in resmî sitesi, kullanım kılavuzları, örnekler ve dökümantasyon sağlar. Bu kaynaklar, çerçevenin tüm özelliklerini ve parçalarını açıklar. Örnekler, nasıl kullanılacağını gösterir.

Bootstrap'i hızlı bir şekilde kullanmaya başlamak istiyorsanız, başlangıç şablonlarını kullanabilirsiniz. Bu şablonlar, projenize Bootstrap'ü entegre eder. Projeyi hızlandırmak için ilk yapınızı oluşturmalısınız.

Topluluk Desteği: Bootstrap ile çalışırken karşılaştığınız sorunları çözmek ve geliştirme sürecinizi iyileştirmek için internetteki topluluklardan yararlanabilirsiniz. Stack Overflow gibi platformlarda sorularak diğer geliştiricilerin deneyimlerini öğrenebilirsiniz.

Bootstrap Kurulumu

Bootstrap İndirme:

İlk adım, Bootstrap'in resmî websitesini (<https://getbootstrap.com>) ziyaret ederek en son sürümünü indirmektir. Ana sayfanın üst kısmında "Download" düğmesini bulacaksınız. İndirilen dosya bir ZIP arşivi olacaktır.

ZIP Arşivini Çıkarma:

İndirdiğiniz ZIP arşivini bilgisayarınıza indirdikten sonra çıkarın. İçeride, Bootstrap'in CSS, JavaScript ve yazı tipleri dosyaları bulunacaktır.

Dosyaları Projeye Ekleme:

Projeyi geliştirdiğiniz dizine gidin ve Bootstrap dosyalarını bu dizine ekleyin. İdeal olarak, bu dosyalar projenizin kök dizininde olmalıdır.

Bağlantıları Ekleme:

HTML belgenizin bölümünde Bootstrap CSS ve JavaScript dosyalarını bağlantılandırın.

Tarayıcıda Test:

Bootstrap ile projenizi geliştirirken, tarayıcınızda test etmeyi unutmayın. Farklı tarayıcılarda ve cihazlarda web sitenizin nasıl görüldüğünü kontrol ederek herhangi bir uyumsuzluğu düzeltebilirsiniz.

Grid Sistem Nedir?

Bootstrap Grid Sistemi, web geliştiricilerin mobil duyarlı ve estetik olarak çekici web siteleri oluşturmalarına yardımcı olan güçlü bir ön yüz çerçevesidir. Bootstrap, açık kaynaklı bir CSS ve JavaScript çerçevesidir ve web tasarımında kullanılan en popüler araçlardan biridir. Bootstrap, Twitter tarafından başlatılan açık kaynaklı bir ön yüz çerçevesidir. Web geliştiricileri ve tasarımcıları için özellikle mobil duyarlı web siteleri oluşturmayı kolaylaştırmak amacıyla geliştirilmiştir. Bootstrap, kullanıcı arayüzü bileşenlerinin yanı sıra birçok stil ve düzen özelliği sunar, ancak en dikkat çekici özelliklerinden biri Grid Sistemi'dir. Grid sistemleri, içeriğin düzenini ve sunumunu düzenlemek için kullanılan bir yapıdır. Bu sistemler, dikey ve yatay çizgilerin birleştiği bir ızgara şeklinde tasarlanır ve tasarımın düzenini daha düzenli ve okunaklı hale getirir. Görsel tasarımda, bu gridler, içeriği düzenlemek, görsel öğeleri daha iyi yönetmek ve sunumları daha düzenli hale getirmek amacıyla tasarımcılar tarafından yaygın bir şekilde kullanılır.

Web Tasarımında, Grid Sistemleri Genellikle İki Ana Kategori Altında Sınıflandırılır

Esnek (Responsive) Grid Sistemi: Bu tür bir grid sistemi, web sayfasının farklı ekran boyutlarına uyumlu olmasını sağlamak amacıyla tasarlanmıştır. Mobil cihazlar, tabletler ve masaüstü bilgisayarlar gibi farklı ekran boyutlarına sahip cihazlar için optimize edilmiştir. Esnek grid sistemleri, içeriğin düzenini dinamik bir şekilde ayarlar, böylece kullanıcılar herhangi bir cihazda tutarlı bir deneyim yaşarlar.

Sabit Grid Sistemi: Sabit grid sistemleri, web sayfasının belirli bir sabit genişlikte olduğu ve ekran boyutlarının değişmediği tasarımlar için kullanılır. Bu tür bir grid sistemi, özellikle daha eski tarayıcılara ve tasarım gereksinimlerine sahip projelerde tercih edilir. İçerik, belirli bir genişlikte sabitlenir ve herhangi bir ekran boyutu değişikliği yapmaz.

Grid Sistemi Kullanımı

Bootstrap Grid Sistemi'nin kullanımı oldukça basittir. Her sütun bir "col-" önekini takip eder, ardından sütun genişliği belirtilir. Örneğin, "col-6" sınıfı, bir sütunu yarı genişlikte oluşturur (12 sütunun yarısı, yani 6 sütun). Bu sınıfları kullanarak sütunları kolayca oluşturabilir ve düzenleyebilirsiniz. Ayrıca, Grid Sistemi, farklı ekran boyutlarına yönelik sınıflar içerir. Örneğin, "col-sm-6" sınıfı, küçük ekranlarda (mobil cihazlar) 6 sütun genişliği sağlar. Bu, içeriğin farklı ekran boyutlarına göre uyumlu hale getirilmesini kolaylaştırır.

Neden Bootstrap Grid Sistemi Kullanmalısınız?

Mobil Duyarlılık: Bootstrap Grid Sistemi, mobil cihazlar dahil tüm ekran boyutlarına uygun web siteleri oluşturmayı kolaylaştırır. Bu, kullanıcı deneyimini artırır ve daha fazla ziyaretçiyi çeker. **Hızlı Geliştirme:** Bootstrap'in hazır bileşenleri ve stil kuralları, web geliştiricilerin daha hızlı ve verimli çalışmalarını sağlar. Daha az kod yazarak profesyonel görünümlü web siteleri oluşturabilirsiniz.

Topluluk Desteği: Bootstrap, büyük ve aktif bir geliştirici topluluğuna sahiptir. Bu, sorunlarınızı çözmek ve yardım almak için birçok kaynak sunar.

Özelleştirme Yeteneği: Bootstrap Grid Sistemi, temalarınızı ve tasarımlarınızı özelleştirmenize olanak tanır. CSS ve SASS kullanarak istediğiniz gibi değişiklik yapabilirsiniz.

Grid sistemi kullanmanın faydaları arasında şunlar bulunur;

Netlik ve Tutarlılık: Grid sistemleri, içeriği düzenlemek için bir standart sağlar, bu da sayfa düzeninin net ve tutarlı olmasını sağlar.

Çeşitli Ekran Boyutlarına Uyum: Özellikle responsive grid sistemleri, farklı ekran boyutlarına sahip cihazlarda web sitenizin düzgün görünmesini sağlar.

Tasarımın Uzun Ömürlü Olması: Grid sistemleri, tasarımın uzun vadeli kullanımını kolaylaştırır. Gelecekteki güncellemeler için temel bir yapı sağlar.

Yeniden Kullanılabilirlik: Grid sistemleri, tasarımın farklı versiyonlarını oluşturmak için önceki çalışmalarını kolayca yeniden kullanılabilir hale getirir.

Grid sisteminin kullanıldığı web tasarımında temel işlevler şunlar;

Sütunlar Oluşturma: Grid sistemleri, sayfanın yatay düzenini oluşturmak için sütunlar oluşturmanıza yardımcı olur. Her sütun, sayfa genişliğinin belirli bir yüzdesini kaplar veya sabit bir genişliğe sahiptir. Bu, içeriği düzenlemenizi ve farklı öğeleri yan yana veya üst üste yerleştirmenizi sağlar.

Sütunlar Arası Boşluk: Grid sistemleri, sütunlar arasında boşluk eklemek için kullanılır. Bu boşluklar, içeriği daha okunaklı hale getirir ve tasarımın estetiğini geliştirir.

Responsive Tasarım: Grid sistemleri, farklı ekran boyutlarına uyum sağlamak için kullanılır. Mobil cihazlar, tabletler ve masaüstü bilgisayarlar gibi çeşitli cihazlarda sayfanın düzgün görünmesini sağlar. Bu, responsive (esnek) tasarımın temelini oluşturur.

Sütunlar Arasında Dağıtım: Grid sistemleri, sütunları farklı genişliklerde dağıtarak içeriklerin dengeli bir şekilde yerleştirilmesine yardımcı olur. Örneğin, bir sayfa üzerinde bir sütunun yarısı, diğer sütunun çeyreği kadar geniş olabilir.

İçerik Yönetimi: Grid sistemleri, içeriğin düzenini ve akışını yönetir. Bu, metin, resimler, videolar ve diğer öğeleri sayfada nasıl yerleştireceğinizi belirlemenize yardımcı olur.

Tasarım Bütünlüğü: Grid sistemleri, tasarımın bütünlüğünü korur. Her sayfa veya sayfa grubu aynı grid yapısını kullanarak tutarlılık sağlar. Bu, marka kimliği ve kullanıcı deneyimini güçlendirir.

Özelleştirme: Grid sistemleri, tasarımın belirli ihtiyaçlara ve estetik tercihlere uyarlanmasını sağlar. CSS ve stil dosyaları aracılığıyla grid özelliklerini kişiselleştirebilirsiniz.

Modülerlik: Grid sistemleri, tasarımın modüler bir şekilde oluşturulmasına olanak tanır. Bu, öğelerin yeniden kullanılabilir olmasını sağlar ve tasarımın hızlı bir şekilde güncellenmesini kolaylaştırır.

İçerik Hiyerarşisi: Grid sistemleri, içeriği hiyerarşik bir şekilde düzenlemenizi sağlar. Başlık, alt başlık, ana içerik ve yan içerik gibi öğeleri farklı sütunlarda düzenleyerek içerik akışını optimize edebilirsiniz.

Çapraz Tarayıcı Uyumluluğu: Grid sistemleri, farklı web tarayıcılarında ve platformlarda (Windows, macOS, iOS, Android vb.) düzgün çalışacak şekilde tasarımın oluşturulmasına yardımcı olur.

Sonuç

Bootstrap Grid Sistemi, web geliştiricilerin mobil duyarlı, hızlı ve estetik olarak çekici web siteleri oluşturmalarına yardımcı olan güçlü bir araçtır. Temel prensipleri basit ve anlaşılır olduğu için hem yeni başlayanlar hem de deneyimli geliştiriciler için idealdir. Mobil cihazların yaygınlaştığı bir çağda, Bootstrap Grid Sistemi, web sitelerinizin her tür cihazda mükemmel görünmesini sağlamak için vazgeçilmez bir araçtır. Bu nedenle, web geliştirme projelerinizde Bootstrap Grid Sistemi'ni kullanarak tasarım sürecinizi hızlandırabilir ve son kullanıcılara daha iyi bir deneyim sunabilirsiniz.

Bootstrap, web geliştirmede çok popüler bir araçtır ve özellikle kullanımı ve tasarımı geliştirmek isteyen geliştiriciler için güçlü bir araçtır. Bootstrap, HTML, CSS ve JavaScript'in kullanıldığı bir çerçevedir. Bu, web geliştiricilerinin hızlı ve etkili bir şekilde duyarlı, kişiselleştirilebilir web siteleri oluşturmasını sağlar. İçerik ve form araçları, kullanıcı arayüzü tasarımını basitleştirme, düzeni düzenleme ve formları stilize etme gibi konularda özellikle önemlidir. Başlık, paragraf, resim ve diğer özellikler Bootstrap içerik araçlarında bulunabilir. Ek olarak, herhangi bir cihazda güzel görünecek şekilde duyarlıdır. Bu araçlar sayesinde sayfa içeriğini hızlı bir şekilde oluşturabilirsiniz.

Kullanıcılar, form araçları aracılığıyla bilgi gönderebilir ve etkileşimde bulunabilir. Bootstrap, form elemanlarına özel bileşenler ve sınıflar sunar. Bu, hızlı bir şekilde form oluşturmanızı ve stilize etmenizi sağlar.

Bootstrap'in içerik ve form araçları, geliştiricilere tasarım becerileri gerektirmeksizin profesyonel görünümlü web siteleri yapmalarına yardımcı olur. Duyarlı tasarımı sayesinde web sitenizin tüm cihazlarda sorunsuz çalışmasını sağlar. Bu, daha fazla ziyaretçi çekmenize ve kullanıcı deneyimini geliştirmenize yardımcı olur.

İçerik Araçları

Yazı Tipi

Font family ile özel yazı tiplerini tanımlayabilirsiniz. Yazı tipi büyüklüğü için Display, h1 ve h2, vurgu için lead ve small ile küçük yazı tiplerini elde edebilirsiniz. Yazının kalın, italik ve diğer metin stilleriyle özelleştirebilirsiniz. Metin rengini ve arka plan rengini de sınıfları kullanarak değiştirebilirsiniz.

Başlıklar

Bootstrap, başlık stillerini ve başlıkları özelleştirmek için çok sayıda sınıf ve özellik sunar. Bootstrap sınıfları, başlık etiketlerini;

,
,
,
,
,
,

kullanarak belirli bir başlık seviyesini tanımlamak ve stilize etmek için kullanılabilir. Başlık metin sınıfları .text-primary, .text-secondary ve diğerleri başlığın rengini belirler. Örneğin, .text-primary başlığı mavi olarak değiştirilir. Bootstrap, başlıkların yanı sıra metin boyutu, yazı tipi ailesi ve diğer metin özelliklerini değiştirebileceğiniz sınıflar sunar. Bu sınıfları kullanarak, web sitenizin başlık ve metinlerini tasarım tercihlerinize göre değiştirebilirsiniz.

Metin Stilleri

Bootstrap, metinleri özelleştirmek için kullanabileceğiniz çok sayıda metin sınıfı sunar. Metin boyutlarından renklere kadar çeşitli metin özelliklerini kolayca değiştirebilirsiniz.

Bağlantılar

Bootstrap, bağlantı ve bağlantı sınıflarını değiştirebilirsiniz. Bu sınıflar, bağlantıları çeşitli renklerde, boyutlarda ve stilde gösterebilir.

Butonlar

Butonlar, kullanıcı etkileşimi, form gönderme ve çok daha fazlasını gerçekleştirmek için kullanılabilir. Temel buton sınıfı btn ile kullanılır. Buton renkleri şu şekilde listelenir;

.btn-primary: Mavi renkte buton.
.btn-secondary: Gri renkte buton.
.btn-success: Yeşil renkte buton.
.btn-danger: Kırmızı renkte buton.
.btn-info: Açık mavi renkte buton.
.btn-warning: Sarı renkte buton.
.btn-light: Açık renkte buton.
.btn-dark: Koyu renkte buton.

Buton boyutları lg ve sm ile ayarlanır. Buton gruplarının kullanımı ise .btn-group şeklindedir.

Listeler

Listeler üç şekildedir. Bunlar sıralı, sırasız ve tanımlı listelerdir. Sıralı listelerdeki öğeler sayı harf ya

da işaretle listelenir. Sırasız listelerdeki öğeler madde ve simgelerle listelenir. Tanımlı listeler kavram ve terim açıklamak için kullanılır. Liste gruplarını kullanmak için list-group sınıfı kullanılır. Listedeki öğeleri renklendirmek için sınıflar sağlar. Örneğin, list-group-item-primary sınıfını kullanarak listedeki öğeleri mavi renklendirmek için bir yol bulabilirsiniz.

Resimler

Sayfanıza resimleri eklemek için etiketini kullanmanız yeterlidir.

Resimlere yuvarlatılmış köşeler ve kenarlık eklemek için .img-thumbnail sınıfı kullanılır. Resimleri mobil cihazlarda daha iyi görünmesi ve responsive tasarım için .img-fluid sınıfı kullanılır. Resim galerisi oluşturmak için .thumbnail sınıfı kullanılır.

Form Araçları

Form Etiketleri

HTML form elemanlarını kullanıcılar için anlamlı bir şekilde etiketlemek ve açıklamak için form etiketleri olarak bilinen HTML öğeleri kullanılır.

Metin Girişi

Bootstrap, metin girişi için kullanılır.

Onay Kutusu

Onay kutuları, kullanıcının önceden belirlenmiş bir listeden birden fazla seçenek seçmesine olanak tanır. Kullanıcı, istediği kadar seçeneği seçebilir veya kaldırabilir.

Radyo Düğmeleri

Radyo düğmesi, kullanıcıların birden fazla seçenek arasından yalnızca birini seçmelerine izin veren form elemanı ve bir grup seçenek arasından yalnızca birini seçmelerine izin verir.

Açılır Liste

Kullanıcıya bir listeden bir şey seçme fırsatı verir. Belirli bir kategoriye veya seçeneğe yönlendirmek, sınırlı ekran alanı kullanmak veya çok fazla seçeneği görüntülemenin karmaşıklığını azaltmak için açık listeler kullanmak yararlıdır.

Metin alanları

Kullanıcılara daha fazla metin girmelerine izin verir. etiketiyle kullanılır. Belirli bir satır ve sütun sayısı ile boyutlandırılabilir. Bu, kullanıcıların metin girmesi gereken durumlarda yararlıdır, özellikle tek satır metin girişi alanları yetersiz olduğunda.

Form Doğrulama

Form doğrulama, kullanıcıların web formlarına doğru bilgileri girmelerini sağlamak için kullanılır.

Form doğrulama, yanlış veya eksik bilgilerin gönderilmesini önler. Bootstrap, kullanıcı deneyimini iyileştirmek ve form doğrulama sürecini kolaylaştırmak için özellikler sunar.

İstemci tarafı doğrulama ve sunucu tarafı doğrulama, web formlarını güvende tutmak ve doğru bilgileri toplamak için kullanılan iki farklı doğrulama yöntemidir. Her biri kendine özgü amaçlar ve avantajlar taşır. Genel olarak hem istemci tarafı doğrulama hem de sunucu tarafı doğrulama en iyi şekilde çalışır. İstemci tarafı doğrulama kullanıcı deneyimini iyileştirirken, sunucu tarafı doğrulama veri bütünlüğünü ve güvenliğini korur. Bu, web uygulamalarını daha güvenli ve kullanışlıdır.

Araç İpuçları

Araç ipuçları olarak da bilinen araç ipuçları, kullanıcılara bir öğenin anlamını veya ek bilgiyi göstermek için kullanılan faydalı bir özelliktir. Bootstrap, özel sınıflar ve JavaScript işlevleri ile araç ipuçları eklemeyi kolaylaştırır.

Sonuç olarak, web geliştiricileri, bootstrap içerik ve form araçlarıyla hızlı ve etkili bir şekilde içerik oluşturabilir ve formlar tasarlar. Bu, web sitelerini geliştirmek isteyen herkes için önemli bir araçtır. Bootstrap, kullanışlı ve görsel olarak çekici web siteleri oluşturmayı kolaylaştırır.

Bootstrap Nedir ?

Bootstrap, web geliştirme için bir CSS ve JavaScript çerçevesi olan açık kaynaklı bir bileşen araçları kütüphanesidir. Bootstrap'in çeşitli özellikleri, basit kullanımı ve geniş topluluk desteği sayesinde popülerliği hızla arttı.

Bootstrap Bileşenleri

Bootstrap ile gelen bileşen araçları, web sitesi veya uygulamasının görünümünü ve kullanıcı deneyimini iyileştirmeye yardımcı olabilir. Bu araçlar, formlar, düğmeler, gezinme çubukları, yükleme çubukları, menüler ve kartlar gibi çok çeşitli öğeleri içerir. Bu bileşenler, kullanıcıların tasarımlarını hızlı bir şekilde yapmalarına yardımcı olur ve bu da zaman kazanmalarını sağlar. Ek olarak, bootstrap'e özgü renk ve tipografi kullanımı, web sayfalarının ve uygulamalarının daha tarz sahibi ve profesyonel görünmesini sağlar. Bootstrap bileşen araçları, çağdaş web tasarımının ihtiyacı duyduğu en önemli özellikleri takip eder. Örneğin, erişilebilirlik standartlarına uyduğu için engelli kullanıcıların deneyimini geliştirir.

Glyphicons

Glyphicons, web sitelerinde kullanılmak üzere önceden tasarlanmış bir ikon setidir. İsviçreli grafik tasarım firması Jan Kovarik tarafından oluşturulan bu simge seti, Bootstrap için tasarlanmış olup toplamda 260 simge içermektedir. Bu ikonlar, web geliştiricilere kullanımı kolay ve estetik ikonlar ekleyerek sayfalarını daha çekici ve anlamlı hale getirme imkânı sunar.

Kullanıcak ikon etiketi içerisinde tanımlanır.

Navs

Navs bileşen, web sitelerinde menü oluşturmak amacıyla kullanılır ve genellikle bir sitenin gezinme çubuğunu veya sekme arayüzlerini temsil eder. Navs, web geliştiricilere kullanımı kolay ve şık navigasyon çözümleri sağlayarak kullanıcıların web siteleri arasında rahatça gezinmelerini sağlar.

Yatay bir menü oluşturmak için

- bileşeni içinde nav sınıfı ve
 - bileşeni içerisinde her bir sekme için nav-item kullanılır. Dikey menü için de flex-column sınıfı kullanılır.

Nav Tabs

Navtabs bileşeni, bir web sitesinin sayfaları, bölümlerini ve içeriğini kolayca dolaşmanızı sağlar. Navtabs, genel olarak üst navigasyon çubuğunda veya belirli bir bölmenin içinde bulunan sekmeler aracılığıyla çeşitli içerik alanlarına erişim sağlar. Bu, web geliştiricilere kullanıcıların web sitesindeki çeşitli konular arasında hızlı bir şekilde geçiş yapmalarını sağlar.

Her bir sekme başlığı içine tab-content kullanılarak içerik yazılır.

Nav Pills

Nav Pills, sekme tabanlı bir arayüz oluşturmak veya sayfa içindeki farklı bölümler arasında hızlı gezinmek için kullanılan bir bileşendir. Bu özel bileşen genellikle yatay bir gezinme menüsü olarak tasarlanmıştır ve kullanıcılara sekmeler aracılığıyla hızlı bir şekilde farklı bölümlere ve içeriklere erişim imkânı sağlar. Nav Pills, web geliştiricilere sayfalarını daha kullanıcı dostu hale getirme ve içeriği daha organize bir şekilde sunma konusunda etkili bir araç sunar.

Oluşturmak için

- içerisinde nav nav-pills tanımlanır ve her sekme için nav-item kullanılır.

Dropdown

Dropdowns, Bootstrap çerçevesinde açılır menüler oluşturmak için kullanılan bir araçtır. Açılır menüler genellikle kullanıcılara daha fazla seçenek veya detaylı bir menü sunmak amacıyla kullanılır ve geniş bir içerik yelpazesine erişim sağlar. İçeriği gizleyebilen ve ihtiyaç duyulduğunda açılabilen menülere sahiptir.

Oluşturmak için içerisinde .dropdown tanımlanmalıdır. Açılır menüyü tetiklemesi için dropdown-toggle ve data-toggle="dropdown" özellikleri tanımlanır.

Alt elemanları tanımlamak için dropdown-item kullanılır.

Navbar

Navigasyon Çubuğu olarak da bilinen Navbar sayfanın üst

kısımındadır. Marka

logosu ve menü öğeleri dahil olmak üzere çeşitli içerikler olabilir. Oluşturmak için içerisinde .navbar tanımlanır.

Breadcrumbs

Breadcrumbs, kullanıcıların bir web sitesinde dolaşırken, buldukları sayfanın konumunu ve geçtikleri yolları anlamalarına yardımcı olan bir yönlendirme aracıdır.

Her sayfa bağlantısı

- etiketi içinde verilir. Her bir kırıntı breadcrumbs-item ile tanımlanır.

Jumbotron

Bootstrap çerçevesinin bir parçası olan Jumbotron, genellikle dikkat çekici bir şekilde öne çıkarılmak istenen ana içeriği vurgulamak için kullanılır. Bu bileşen tipik olarak, başlık, alt başlık, açıklama metni, düğmeler ve diğer içerik öğelerini içeren büyük bir kutu olarak tasarlanır.

Oluşturmak için içinde .jumbotron tanımlanmalıdır. İçerik içinde yazılır. Tam genişlikte olması için jumbotron-fluid sınıfı kullanılmalıdır.

Alerts

Bootstrap, kullanıcılara belirli durumları bildirmek veya önemli bilgileri iletmek için "Uyarılar"ı içerir. Web sayfalarında kullanıcıya mesaj göndermek, başarı durumlarını, hata durumlarını veya bilgi sağlamak için uygundur. Oluşturmak için içinde belirtilecek uyarı sınıfı tanımlanır ve içerik yazılır. Dört temel yapıdan oluşur. Succes, Warning, Info ve Dangerdir. Kapatma düğmesi eklenerek uyarı kapatılabilir.

Badge

Bootstrap'ta "Badge", genellikle durumları veya sayısal değerleri belirtmek için kullanılan küçük bir rozettir.

Label

Bootstrap'ta genellikle bir durumu veya kategoriye temsil eden metin etiketi olarak tanımlanır. Oluşturmak için label sınıfı kullanılır. Renklendirmek için Bootstrap bağlamsal renk sınıfları kullanılır.

Progress Bar

Bir görevin ya da sürecin tamamlama ve ilerleme durumunu ilerleme çubuğu ile takip edilir. Progress bar'lar genellikle içerisine bir yüzdeler değeri gösteren bir etiket veya içerik içerir. Oluşturmak için etiketi içerisinde .progress-bar tanımlanır.

Media Object

Media Object metin, resim ve diğer medya içeriğini düzenlemek ve bir araya getirmek için kullanılan bileşendir. Bu, özellikle liste öğeleri, blog gönderileri ve haber öğeleri gibi içeriği düzenlemek için kullanılır. .media ve media-body bu bileşeni oluşturur. Başlık yazıldıktan sonra içerik metni yazılır. Sağa, sola ve ortaya hizalaması yapılır. Liste haline getirmek için media-list sınıfı kullanılır.

Wells

Belirli bir bölgeyi vurgulamak veya içeriği bir kutu içinde çerçevelemek için wells bileşeni kullanılıyordu fakat Bootstrap 4'te bu özellik yerini Cards bileşenine bırakmıştır.

Cards

Bu bileşen, içerik bloklarını görsel olarak çekici bir şekilde sunmak için kullanılır. Card bileşeni tipik olarak resim, başlık, metin ve bağlantılar gibi öğeleri içeren bir kutu içerir. Bu, web sitelerinde ürünler, makaleler ve kullanıcı profilleri gibi çeşitli içerikleri düzenlemek için sıklıkla kullanılır. içerisinde .card sınıfı tanımlanır. Resim eklemek için kullanılır. Card-title ile başlık tanımlandıktan sonra metin eklemek için etiketi içinde card-text tanımlanır.

Alt ve üst bilgi eklenir. Arka plan renk ayarlaması Bootstrap bağlamsal renk sınıflarıyla değiştirilir.

List Group

Genellikle liste grupları, bir dizi bağlantı, düğme veya diğer içerik öğesi içeren düzenli bir liste oluşturmak için kullanılır. Oluşturmak için

- içerisinde list-group tanımlanır. Her bir liste elemanı list-group-item ile gösterilir. Etkin listeyi göstermek için .active kullanılır.

Thumbnails

Bootstrap'ın "Thumbnails" bileşeni, bu tür küçük resim ve önizlemeleri düzenlemek ve göstermek için kullanılır. Oluşturmak için içerisinde thumbnails tanımlaması yapılır.

Panels

Paneller içerik gruplarını çerçeveleyen ve düzenleyen bir bileşendir. Üç temel yapısı bulunur. Panel başlığı, içeriği ve alt bilgisi. içerisinde panels tanımlaması yapılır.

ScrollSpy

Bootstrap Scrollspy, sayfanın belirli bölümleri arasında gezinmeyi kolaylaştıran bir bileşendir. Body etiketi içerisinde data-spy="scroll" ve data-target sınıfları tanımlanır.

Modals

Kullanıcıyla etkileşim kurmak için ekranın ortasında beliren küçük bir pencere veya diyalog kutusu modals olarak bilinir.

Bootstrap bileşen araçları, web geliştiricilere profesyonel ve kullanıcı dostu web siteleri ve uygulamalar oluşturmada yardımcı olan güçlü araçlardır. Kullanımı kolay, zengin özelliklere sahip ve geniş bir topluluğa sahip olması sayesinde, Bootstrap web tasarım sürecini hızlandırır ve olağanüstü sonuçlar elde etmeyi mümkün kılar.

Carousel Nedir?

Bootstrap Carousel, web sayfalarında döngüsel olarak içerik sunan bir bileşendir. Genellikle görsel içerikleri (resimler, sliderlar, reklamlar vb.) döngüsel olarak göstermek için kullanılır. Kullanıcılar, otomatik olarak veya manuel olarak kontrolleri kullanarak slaytlar arasında geçiş yapabilirler.

Web sitelerinde carousel'lar, ana sayfa görselleri, ürün galerileri veya hikâye anlatımları gibi çeşitli amaçlarla sıklıkla kullanılır. İki boyutlu bir döngüsel gezinti aracı olarak, kullanıcıların içeriği etkileşimli bir şekilde keşfetmelerini sağlayarak sitelerin veya uygulamaların kullanıcı deneyimini geliştirebilirler.

Carousel Yapısı:

Bootstrap Carousel, temel olarak şu ana bileşenleri içerir:

Carousel Container (div.carousel): Carousel'ın temel container'ıdır. Carousel'ın genelini sarar.

Carousel Inner (div.carousel-inner): Carousel içerisindeki slaytların bulunduğu bölümdür.

Carousel Item (div.carousel-item): Her bir slaytı temsil eder. Carousel Inner içinde bulunur ve görsel içeriği içerir.

Carousel Control (div.carousel-control): Carousel'ın manuel kontrol düğmelerini içerir. Önceki ve sonraki slaytlara geçiş yapmak için kullanılır.

Carousel Indicator (ol.carousel-indicators): Carousel'daki slayt sayısını ve hangi slaytın görüldüğünü gösteren noktaları içerir.

Carousel Caption (div.carousel-caption): Slaytların üzerine eklenen opsiyonel açıklama veya başlık alanıdır.

Carousel Interval (data-bs-interval): Slaytlar arasındaki geçiş süresini ayarlamak için kullanılır.

Slayt içeriğini eklemek için "img" etiketini içerisine eklenir. Blok düzenini sağlamak ve resimlerin genişliğini ayarlamak için class="d-block w-100" sınıfları kullanılır.

Slayt geçişleri prev ve next sınıflarıyla gerçekleştirilir. Prev önceki slayta geçmek için next ise sonraki slayta geçmek için kullanılır.

Otomatik Oynatma:

Carousel, sayfa yüklendiğinde otomatik olarak başlatılabilir. Bunun için data-bs-ride="carousel" özelliği kullanılır. Ayrıca, data-bs-interval özelliği ile otomatik geçiş süresi belirlenebilir.

Geçiş Efektleri:

Carousel, slaytlar arasında geçiş yaparken efektler ekleyebilir. Varsayılan olarak, Bootstrap Carousel fade (solma) efektini kullanır, ancak sağa kaydırma, sola kaydırma gibi efektler de seçilebilir.

Özelleştirme ve Etkileşim:

Carousel'ı daha fazla özelleştirmek için JavaScript kullanabilir ve olay dinleyicileri ekleyebilirsiniz. Kullanıcılar bir slayta tıkladığında veya geçiş yapılacak bir önceki/sonraki düğmeye tıkladığında özel etkileşimler ekleyebilirsiniz.

Kullanmak için gerekli olan kütüphaneleri indirilmelidir. CSS, jQuery, Popper.js ve JS gerekli kütüphanelerdir.

Dokunmatik ekranlara sahip cihazlarda slaytları manuel olarak sağa ve sola kaydırmak mümkündür. Bu özelliği data-bs-touch="false" olarak ayarlanırsa pasif hale getirilir.

HTML yapısı, geçiş kontrolleri, resim boyutları, görsel içerik uyumu, geçiş süresi ve sürüm uyumluluğu Carousel kullanırken dikkat edilmesi gereken önemli noktalardır.

Sonuç olarak Carousel'ın başlıca özellikleri şunlardır;

Döngüsel Gösterim: Carousel, içeriği döngüsel olarak gösterir, yani son öğeden sonra ilk öğeye geri döner.

Otomatik ve Manuel Geçiş: Otomatik bir modda, carousel otomatik olarak içerikleri belirli bir süre aralığında geçirir. Manuel modda ise kullanıcılar ileri ve geri düğmelerini kullanarak veya belirli bir öğeye tıklayarak geçiş yapabilirler.

Duyarlı Tasarım: Modern carousel'lar genellikle farklı ekran boyutlarına uyum sağlamak için duyarlı tasarım prensiplerini benimser. Bu, mobil cihazlarda veya tabletlerde optimal görüntüleme sağlar.

Çeşitli İçerik Türleri: Carousel, resimlerden metinlere, videolara ve diğer medya türlerine kadar birçok farklı içeriği destekleyebilir.

İlerleme Göstergesi: Carousel genellikle kullanıcıya hangi öğenin görüntülendiğini gösteren bir ilerleme çubuğu veya nokta şeklindeki göstergeler içerir.

İleri ve Geri Kontrolleri: Kullanıcılar genellikle bir sonraki veya önceki öğeye geçmek için ileri ve geri kontrol düğmelerini kullanabilirler.

Özelleştirilebilir Stiller: Carousel'ın tasarımı genellikle kullanıcı ihtiyaçlarına uyacak şekilde renk, boyut ve stil gibi özelliklerle özelleştirilebilir.

Akordeon

Uzun içerikli sayfaları düzenlemek, menülerin alt seviyelerini göstermek veya formların bölümlerini gizlemek gibi çeşitli kullanım durumlarında akordeonlar genellikle tercih edilir. Web siteleri ve uygulamalar için önemli bir kullanıcı arayüzü öğesidir; kullanıcı dostu bir gezinme ve içerik organizasyonu sağlar. Akordeon oluşturmak için .accordion sınıfı kullanılır.

Akordeonun temel özellikleri şunlardır:

Başlık ve İçerik: Akordeon, genellikle başlık ve bu başlıkla ilişkilendirilmiş bir içeriği içerir.

Başlıklar genellikle kullanıcıya içeriğin ne hakkında olduğunu anlatan kısa metinlerdir.

Genişletme ve Daraltma: Kullanıcı, başlıklara tıklayarak veya dokunarak içeriği genişletebilir veya daraltabilir. Bu, kullanıcıya sayfanın veya uygulamanın daha fazla bilgiye sahip olmasını sağlar.

Duyarlı Tasarım: Modern akordeonlar genellikle farklı ekran boyutlarına uyum sağlamak için duyarlı tasarım prensiplerini benimser. Bu, mobil cihazlarda veya tabletlerde optimize edilmiş bir görüntüleme sunar.

CSS ve JavaScript Kullanımı: Akordeonlar genellikle CSS (Cascading Style Sheets) ve JavaScript veya jQuery gibi teknolojileri kullanarak etkileşimli özellikler kazanır.

SEO Dostu: Doğru bir şekilde yapılandırıldığında, akordeonlar SEO dostu olabilir, çünkü içerik ve başlıklar düzgün bir şekilde indekslenebilir.

Çeşitli Tasarımlar: Akordeonlar, farklı tasarım stilleri ve renk seçenekleri ile özelleştirilebilir, böylece kullanıcı deneyimi ve uyumluluk artırılabilir.

Buton

Butonlar, kullanıcıların web siteleri ve uygulamalarda belirli eylemleri gerçekleştirmelerine ve etkileşim kurmalarına izin veren temel kullanıcı arayüzü öğeleridir. Buton oluşturmak için .btn sınıfı kullanılır. Butonlara renk vermek için Bootstrap'ın bağlamsal renk sınıfları kullanılır.

Primary

Secondary

Success

Danger

Warning

Info

Light

Dark

Buton boyutlarını ayarlamak için sm ve lg sınıfları kullanılır. Butonları işlemlerde devre dışı bırakmak için .disable sınıfı kullanılır.

Close Buton

Close butonları, genellikle modal pencerelerde veya açılır pencerelerde kullanılan küçük bir düğmedir.

Oluşturmak için .btn-close sınıfı kullanılır.

Buton Group

Butonlar, web siteleri ve uygulamalarda kullanıcılara etkileşimde bulunmaları ve belirli eylemleri gerçekleştirmeleri için bir araç sağlayan temel kullanıcı arayüzü öğeleridir. Oluşturmak için .btn-group sınıfı kullanılır. Butonlarda kullanılan bağlamsal renk sınıfları kullanılır. Düğme boyutlandırma için btn-group-lg ve btn-group-sm sınıfları kullanılır. Butonları dikey olarak yerleştirmek için .btn-group-vertical sınıfı kullanılır.

Ekranın genişliğinde ve tüm ekranı kaplayan butonlar oluşturmak için .btn-group-justified sınıfı kullanılır. Dropdown kullanılarak açılır menüler oluşturmak mümkündür.