



W3'ÜN 3 ATLISI

HTML, CSS

VE JAVASCRIPT İLE
MODERN WEB TASARIMI

FATİH ÇELİK

Yazar Hakkında

1983 yılında Ankara'da doğdum. Gazi Üniversitesi Bilgisayar Sistemleri Öğretmenliği Bölümü'nden 2008 yılında mezun olarak aynı yıl İstanbul Başakşehir Çok Programlı Anadolu Lisesi'nde Bilişim Teknolojileri Öğretmeni olarak görev yapmaya başladım.

Web Tasarımı ve Programcılığı alanında özellikle Html,Css,JavaScript,Asp.net,Asp.net MVC ve Sql Server teknolojileri eğitimi üzerinde çalışmalarda bulundum. Web teknolojileri geliştirme konusunda eğitim öğretimi güncel dökümanlarla desteklemek amacıyla video,e-kitap ve uygulama örnekleri oluşturup bu alanda kendini geliştirmek isteyen öğrencilerimle paylaşıyorum.

Halihazırda eğitim dökümanlarımı yayınladığım www.webmucidi.com isimli sitenin ve [YouTube/WebMucidi](https://www.youtube.com/channel/UC...) kanalının yöneticiliğini yapmaktayım.

"Bir problemi çözmeye ilk başladığınızda, ilk bulduğunuz çözümler oldukça çok karmaşıktır ve birçok insan burada durur. Ancak devam ederseniz, sorunla yaşarsanız ve soğanın daha fazla katmanını soyarsanız, çoğunlukla çok şık ve basit çözümlere ulaşırsınız."

Steve Jobs, 2006

İçindekiler

1- İnternet Ortamı ve Tasarım

İnternetin tanımı ve işleyişi, tasarımın temel ilkeleri

2- Html ile Sayfa İskeleti Oluşturma

Html tanımı,kullanımı ve tarihçesi, Html komut yapısı ve etiketleri

3- Css ile Sayfa Stili Tasarlama

Css yazım kuralları,seçicileri ve özellikleri

4- JavaScript ile İstemci Taraflı Web Programlama

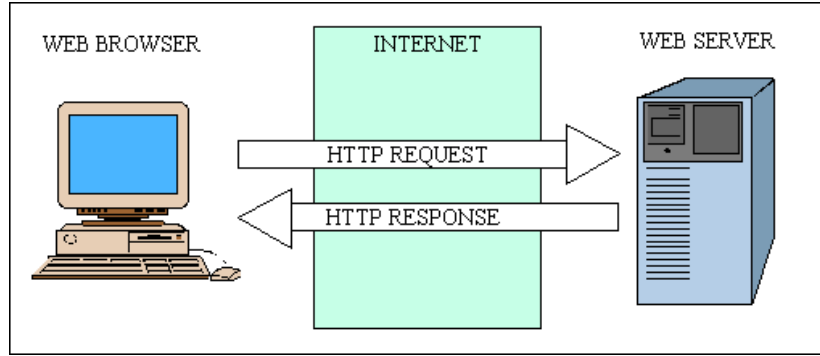
JavaScript ile web sayfasında programlama ve animasyon işlemleri

1- İnternet Ortamı ve Tasarım

1.1. İnternet ve tasarımıla ilgili temel kavramlar

İnternet dünya üzerinde milyonlarca bilgisayarın birbiriyle bağlantılı olduğu bir ağ olarak tanımlanabilir. **WWW(World Wide Web)** ifadesi de dünya çapında internet ağı olarak Türkçe'ye çevirildiğinde aslında bu tanımlı karşımıza getirmektedir. Web siteleri tasarlayabilmek için öncelikle web teknolojinin işleyişini iyi anlamak ve kavramlara hakim olmak gereklidir.

İnternet ortamında iletişim ve paylaşımlar web sayfaları üzerinden gerçekleştirilir. Temel olarak işleyiş bir web sitesine bağlanmak isteyen istemcinin(**client**) tarayıcı(**browser**) vasıtasıyla istediği adrese(**domain**) giderek siteye ait içeriklerin(resim,yazı,video vs.) sunucu bilgisayardan(**server**) belli kurallar ile(**http protokolü**) alınarak ekranda gösterilmesiyle sonuçlanır.



Bir web sitesinin geliştirilmesi aşamasında tasarım(**web ui-ux/front-end designer**),kodlama(**web coder/back-end programmer**) ve içerik yönetiminin sağlanması(**web master/administrator**) işleri **web developer/Full-stack developer(proje geliştirici)** olarak adlandırılan kişilerce yürütülür ve bu işlemler global olarak kabul edilegelmiş belirli ilkelere bağlı kalınarak yapılmalıdır.

Web Tasarımında Temel İlkeler:

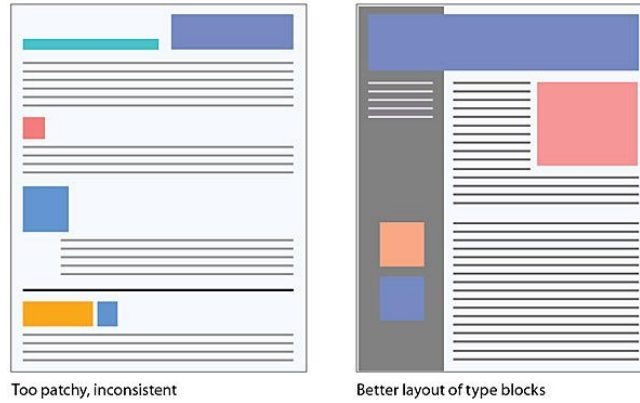
a. İçerik b. Tasarım c.İşlevsellik d.Kullanılabilirlik e. Tarayıcı ve Cihazlarla Uyumluluk

Bu bahsedilen ilkeler içerisinde en ön planda dikkate alınması gereken şey sitenin belli bir konuya özel içerik sunmak için oluşturulduğunun ve dolayısıyla diğer bütün unsurların **içeriği ön plana çıkarmak** için destekleyici yan faktörler olduğunun unutulmamasıdır. Sitenin özgün ve tutarlı bir tasarıma sahip olması; filtreleme,kategorize etme vs. gibi özellikler sayesinde işlevsel olması; sürekli kullanılabilir ve erişilebilir olarak ve farklı platformlarda(tarayıcı,mobil cihaz,işletim sistemi vs.) çalışır durumda olması aslında içeriğe ulaşma konusunda bir siteyi etkin ve iyi bir site kılar.

Biçimsel Özellikler:

a.Renk Dengesi b.Okunabilirlik c.Hizalama d. Metin ve Nesne Arası Boşluklar

Temel ilkeler doğrultusunda içeriği sunmak için yapılan tasarımda ise mutlaka yukarıdaki başlıklar dikkate alınmalı ve içeriğin kullanıcı tarafından etkin şekilde algılanabilmesi sağlanmalıdır. Örneğin site tasarımında projenin içeriğine göre kurumsal karaktere uyumlu renk seçimleri yapılmalı,sıfırdan bir tasarım yapılacaksa kurumsal kimliği belirlemek için sektör tanımı,müşteri beklentileri vs. analiz edilerek gerekirse hazır renk paletleri ve şablonları kullanılıp(Örnek: colorhunt.co) tutarlı ve dengeli renklere sahip tasarımlar oluşturulmalıdır. Örneğin bilimsel olarak kırmızı renginin acıkma hissini tetiklediği olgusundan yola çıkarak yemek sektöründe faaliyet gösteren bir firmanın web sitesi tasarlanıyorsa kırmızı ve tonları tasarım için uygun bir seçim olacaktır. Tasarımlarda genelde 4 ana renk seçilerek tasarım öğeleri bu renklerle kurgulanır.



Bu renklerin kullanımı içeriğin okunabilirliğini azaltmamalı,dolayısıyla zemin ve yazı renkleri genelde zıt renklerden seçilmelidir. İçeriklerin hizalı olmasına ve birbirleri arasındaki boşluk ve mesafelerine de yine içeriğin kolay algılanabilmesi ilkesine hizmet edecek şekilde dikkat edilmelidir. İçerik kullanımında sayfadaki beyaz alanların(white space) kullanımına önem verilmelidir. Beyaz alanların bolca kullanımı okunabilirliği kolaylaştırmaktadır. İçerik için kullanılacak font tipinin seçimi de yine okunabilirlik ve algının yönetimi için oldukça önemli bir etken olarak sayılabilir.

Çözünürlük kavramı ve site taslağı oluşturma:

Çözünürlük, bir bilgisayar ekranının desteklediği piksel genişliği ve yüksekliğine verilen addır. Dolayısıyla yapacağımız tasarımların bir bilgisayar veya mobil bir cihazın ekranında görüntüleneceğini düşünürsek çözünürlük kavramı tasarım için oldukça önemlidir. Hedef kitlenin siteyi açacağı ortalama çözünürlük değerleri genel kullanım istatistikleri dikkate alınarak belirlenmeli veya farklı çözünürlüklere uyumlu yani responsive olacak şekilde dizayn edilmelidir. Günümüzde yaygın şekilde rastlanan HD ekranlar için 1920×1080p çözünürlük değeri kullanılmakta olup, mobil cihazlar için bu ölçü ortalama olarak genişlikte 600-1000 px arasında değişmektedir.

Cihazlar için tanımlı genişlik değeri tarayıcı açısından “**viewport**” olarak ifade edilen görüntülenebilecek alan olarak algılanır. Yükseklik değeri web sayfaları için scroll kaydırma sayesinde tasarımcıya biraz daha esnek bir alan bırakırken genişlik değeri site taslağı oluştururken dikkate alınmalı ve taslak boyutları genel kullanıcı kitlesine hitap edecek şekilde oluşturulmalıdır.

Web Tarayıcılar:

Web sayfalarını görüntülemeye yarayan yazılımlardır. Örnek olarak: Internet Explorer,Edge,Mozilla Firefox,Google Chrome,Opera,Safari gibi yazılımlar verilebilir.

Domain,Host Kavramları:

Domain (alan adı) sizin, yada sahip olduğunuz şirketin İnternetteki kimliğidir. İnternette yer almak için başvurduğunuzda size bir İnternet protokol numarası verilir. Buna IP Adres (İnternet Protokol adres) denilmektedir.

İnternetteki her bilgisayarın bir IP adresi vardır. Diğer bir bilgisayarla bağlantı kurabilmek için karşı bilgisayarın IP adresinin tanınmış olması gerekir. Örneğin www.webmucidi.com' un IP adresi 213.144.111.115 gibi bir değerdir. İnternetteki diğer bilgisayarlar, sizi bu IP numarası sayesinde bulurlar. Domain ise bu numaranın sadece bir görüntüsüdür.

Yani siz browserinizin adres çubuğuna IP numaralarını yazmak yerine o IP numaralarının kodlandığı domainleri yazarak DNS(Domain Name Server) sayesinde adresin çözümlenmesiyle istediğiniz siteye ulaşırsınız. Domain bu anlamda hafızada kalması zor olan IP numaralarının daha kolay kullanımını sağlar. Kısaca özetlemek gerekirse domain IP adreslerinin harfler şeklinde kodlanmasıdır.

Domainler sonlarındaki .com .net ve .org gibi uzantılara göre sınıflandırılır:

.com---- ticari amaç için kurulan siteler için kullanılır
.net---- network açılımıdır. bilgilendirme ve internet hakkındaki ticari amaçlı siteler içindir
.org--- ise ticari amaç gütmeyen sitelerdir.
.edu--- eğitim kurumları için kullanılır.
.gov--- devlet kurumları için kullanılır.
.gov.tr kullanımındaki .tr ülke uzantısıdır.

Host kelime anlamı olarak barındırmak demektir. Aslında internet ortamındaki anlamı da tam olarak budur. Yani internet sitenizin sanal alemde barındığı yerdir. İnternet siteniz için hazırladığınız yazıların, resimlerin, ses dosyalarının internette bir yerde saklanması gerekir.

İnterneti kendi kişisel bilgisayarınız gibi düşünürseniz Resimlerinizi resim klasöründe, müziklerinizi ise Müziklerim klasöründe saklarsınız ihtiyaç duyunca buradan çağırır ve kullanırsınız, işte internet siteleri için host (Barınak) aynıdır. Bunun tek farkı internet sitenize sadece sizin bakmanız değil birçok insanın bakma ihtiyacı olduğu için bu bilgisayarın sürekli açık kalması gerektiridir.

İnternet sitesi açmak için öncelikle bir alan adı almanız gerekmektedir (Domain) . Daha sonra bu alan adını hosting aldığınız bilgisayarın İP adresine yönlendirirsiniz ve siteniz artık yayında olur. İnternet sitelerinin host edildiği bu sürekli açık kalan bilgisayarlara Server (Sunucu) denilmektedir.

2- Html ile Sayfa İskeleti Oluřturma

2.1. Html Nedir?

HTML web sayfaları tasarlamaya yarayan bir dil olarak tanımlanabilir. Bu script dili ile HTML etiketi(tag) olarak adlandırılan işaretleme(markup) komutları kullanılarak uzantısı .html olan web sayfaları oluşturulur. İnternet üzerinde browserlar vasıtasıyla görüntülediğimiz her sayfa mutlaka HTML komutları ile kodlanmış bir dökümandır.

Web sayfası tasarlama işindeki aşamaları bir binanın inşa edilme süreci ile benzetecek olursak binayı oluşturan çimento,kum,demir,su,kereste vs. gibi temel yapı malzemeleri web sayfası tasarımında kullanılan HTML etiketlerine benzetilebilir.

Sayfa tasarımını oluşturan bu etiketler daha sonra CSS ile stil oluşturmak suretiyle biçimlendirilir ve görsellik kazandırılır. Javascript tabanlı istemci taraflı çalışan uygulamalarla kullanıcı dostu,yüksek performanslı ve göze hoş gelecek şekilde sayfa zenginleştirilebilir. Ayrıca sayfa üzerinden kullanıcı ile etkileşimli uygulamalar kullanmak ve veritabanı işlemleri yapmak Asp,Php,Asp.net vb. gibi sunucu taraflı çalışan web programlama dilleri ile mümkün kılınabilir.

Bir web sayfasında kullanılması mümkün olan bütün bu teknolojilerin olmazsa olmazı ise aynen bir bina inşaatında kullanılması zorunlu olan yapı malzemeleri gibi sayfayı oluşturan HTML kodlarıdır.

İlk olarak 1991 yılında kullanılmaya başlayan HTML dili zaman içerisinde gelişim göstererek son olarak 2014 yılında yayınlanan HTML5 sürümüyle son halini almış ve halen gelişimini sürdürmektedir. HTML5 ile birlikte gelen yenilikler sayfanın temel yapısını oluşturmakla beraber sayfa görünümünü zenginleştirme,etkin multimedia kullanımı,sürükle-bırak uygulamaları gibi modern web tasarımının ihtiyaçlarını karşılayacak işlemler konusundada web uygulama geliştiricilerinin işini oldukça kolaylaştırmıştır.

2.2. Html Editörleri

Bir HTML sayfası oluşturmak için kullanılacak kodları yazma,düzenleme ve kaydetme gibi işlemleri yapan yazılımlar genel olarak web tasarım editörleri olarak anılır. Bu editörlerden başlıcaları Visual Studio Code,Sublime Text,Notepad,Notepad++,WebMatrix ve Adobe Dreamweaver gibi programlardır. Bir windows işletim sistemi üzerinde standart olarak bulunan Notepad yazılımı sayfa tasarımında kullanılabilecek editörlerin başında gelmektedir. Daha gelişmiş işlemler ve kod yazımı için sayılan diğer editörlerin kullanılması tasarımcının işini kolaylaştırması açısından daha verimli olacaktır. Özellikle VS Code editörü gelişmiş arayüzü,eklenti desteği ve emmet* ile kod yazımını pratikleştirmesi ile diğer editörlerin birkaç adım önüne çıkmaktadır. Browser üzerinde çalışacak bir web sayfası oluşturmak için bu programlarla açılan dökümanın html formatında kaydedilmesi yeterlidir.

Bilgi: emmet kodları bazı kısayollar sayesinde kod yazımı kolaylaştırır. Örneğin kod yazım ekranında “!” işareti basıldığında bir html dökümanına ait temel omurga otomatik olarak kodlanır.

2.3. Html Komut Yapısı

Web sayfasını oluşturan temel iskeleti oluşturan HTML etiketlerini yine benzetme yöntemiyle açıklarsak insan vücudunun yapısını örnek olarak gösterebiliriz. HTML komut yapısı aynı bir insan vücudunda olduğu gibi temel bilgilerin tutulduğu bir kafa(head) ve fiziksel yapıyı oluşturan bir gövdeden(body) oluşmaktadır. Aşağıda temel olarak her HTML dökümanında bulunması gereken yapı gösterilmiştir.

```
<!DOCTYPE html>
<html>
<head>
    <!-- Sayfaya ait dil,karakter desteği,site başlığı,tasarımcı bilgileri vs. tanımlandığı bölümdür -->
</head>
<body>
    <!--Tarayıcı ile ekranda görüntülenen tüm nesnelerin tanımlandığı gövde bölümüdür -->
</body>
</html>
```

Etiket	Açıklama
<u><html></u>	Kod tanımının başlangıç komutudur.
<u><head></u>	Temel bilgilerin tanımlandığı bloktur.
<u><body></u>	Gövde içinde yer alan nesnelerin tanımlandığı bloktur.
<u><!DOCTYPE html></u>	HTML5 ile sadeleşmiş döküman tanımlama komutudur.
<u><!-- --></u>	Açıklama ekleme komutudur.

2.4. Html Etiketleri

HTML etiketleri sayfa içerisinde farklı iş ve işlevleri icra etmeye yarayan işçiler veya nesneler olarak tanımlanabilir. HTML etiketleri `<tag>...</tag>` şeklinde etiketin açılıp kapanmasıyla tanımlanır. Bazı etiketlerde kapama işlemine ihtiyaç duyulmaz,bazılarında ise açma kapama işlemi tek bir etiket içerisinde gerçekleşir. Etiketler birbirlerini kapsayacak şekilde iç içe yazılabilir. Bazı etiketlerin içerisinde attribute adı verilen o etikete has özelliklere çeşitli değerler aktarılabilir. HTML etiketlerinde büyük-küçük harf ayırımı yoktur fakat kodlamaların küçük harflerle yapılması daha çok tercih edilen bir yöntemdir. Aşağıdaki tabloda HTML yazım şekillerine ait örnekler verilmiştir.

Açılış etiketi	Etiket içeriği	Kapanış etiketi	Açıklama
<code><p></code>	Bu bir paragraftır.	<code></p></code>	Açılan etiket kapsadığı nesneden sonra kapatılmıştır.
<code></code>	Bu bir linktir.	<code></code>	Bu link href özelliği(attribute) kullanılarak yönlendirilmiştir.
<code>
</code>	Bir alt satıra geçirir.		Kapama etiketi kullanılmamıştır.
<code></code>	Resim ekler.		Kapama için sonda <code>/></code> kullanılmıştır.
<code><HEAD></code>	Sayfaya ait temel bilgiler	<code></head></code>	Büyük-küçük harf kullanımı önemsizdir.
<code><body><h1></code>	Bu bir başlıktır.	<code></h1></body></code>	Kodlar iç içe yazılabilir.

HTML Paragraf Etiketi : `<p>` etiketi ile sayfada paragraf tanımlamak için kullanılır.

Kodlama:	Ekran çıktısı:
<pre> <p> Burası birinci paragraftır. </p> <p> Burası ikinci paragraftır. </p> </pre>	<pre> Burası birinci paragraftır. Burası ikinci paragraftır. </pre>

Bilgi: `
` bir satır alta geçmek, `<hr>` sayfada düz çizgi oluşturmak için kullanılan bağımsız etiketlerdir.

HTML Başlık Etiketleri : <h1> <h6> arasında tanımlanabilen etiketlerdir.

Kodlama: <h1> Paragraf Başlığı 1 </h1> <h2> Paragraf Başlığı 2 </h2> <h3> Paragraf Başlığı 3 </h3> <h4> Paragraf Başlığı 4 </h4> <h5> Paragraf Başlığı 5 </h5> <h6> Paragraf Başlığı 6 </h6>	Ekran çıktısı: <h1>Paragraf Başlığı 1</h1> <h2>Paragraf Başlığı 2</h2> <h3>Paragraf Başlığı 3</h3> <h4>Paragraf Başlığı 4</h4> <h5>Paragraf Başlığı 5</h5> <h6>Paragraf Başlığı 6</h6>
---	--

HTML Metin Biçimlendirme Etiketleri :

Kodun kullanımı:	Ekran görüntüsü:
Yazıyı kalınlaştırma	Yazıyı kalınlaştırma
Yazıyı kalınlaştırma	Yazıyı kalınlaştırma
<i>Eğik yazı oluşturma</i>	<i>Eğik yazı oluşturma</i>
Eğik yazı oluşturma	<i>Eğik yazı oluşturma</i>
<code>Bilgisayar çıktısı formatı</code>	Bilgisayar çıktısı formatı
<p>Burası_{altyazı} Burası ^{üstyazı}</p>	Burası _{altyazı} Burası ^{üstyazı}
<p>Üstü ve altı çizili yazılar: sarı <ins>lacivert</ins></p>	Üstü ve altı çizili yazılar: sarı <u>lacivert</u> !
<p><small>Yazı boyutunu küçültme</small></p>	Yazı boyutunu küçültme
<p>İşaretli yazı oluşturma: <mark>Türkiye</mark></p>	İşaretli yazı oluşturma: Türkiye

HTML Link Oluşturma Etiketi : <a> etiketi ile sayfa içi veya sayfa dışı bağlantı oluşturmaya yarar.

Örnek kullanım: Buraya tıklayın

<a> etiketinde href özelliği içerisine hedef adres bilgisi aktarılır. Yeni sayfanın açılma şekli ise target özelliğine verilen değerler ile belirlenir. “_blank” değeri sayfanın yeni pencerede açılmasını sağlar.

Sayfa içi link oluşturma:

Sayfa başına döndünüz. <p>Bu kısımda uzun bir sayfa içeriğinin bulunduğu kabul edelim.</p> Sayfa başına dönmek için tıklayınız.	<a> etiketinin id özelliğine aktarılan bir kimlik ile sayfa içerisindeki bir bölüm işaretlenir. Ve sayfanın başka bir yerinden bu işaretin olduğu yere yönlendirme yapılarak gidilebilir.
--	---

Resimlere link verme:

Mail adresine link verme: Mail Gönder

Bilgi: <iframe> etiketi ile başka bir sayfaya link vermek yerine harici bir sitenin içeriği sayfaya gömülebilir. Örneğin bir lokasyona ait harita bilgisi Google servisi üzerinden siteye yerleştirilebilir.

<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d24069.255265683954!2d28.759022809804225!3d41.054631374933386!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x0%3A0x79e566b3231b1d85!2zQmHF2FrXZ9laGlyIMOHB2sgUHJvZ3JhbWZEsSBBbmFkb2x1IEhpc2VzaQ!5e0!3m2!1str!2str!4v1604951245483!5m2!1str!2str" width="600" height="450" frameborder="0" style="border:0;" allowfullscreen="" aria-hidden="false" tabindex="0"></iframe>

Head Bölümü Etiketleri : Head etiketiyle tanımlanmış bölüm içerisinde daha öncede belirttiğimiz gibi sayfaya ilgili temel birtakım tanımlamalar yapılır. Head bölümü içerisinde kullanılan etiketlerden bazıları HTML5 standartlarına göre aşağıda listelenmiştir.

Etiket	Kullanım şekli	Açıklama
<title>	<title>Site Başlığı</title>	Tarayıcıda görüntülenecek site başlığını oluşturur.
<base>	<base href="/~/images/" target="_blank">	Sayfada tanımlı tüm linkler için varsayılan bir adres ve/veya linklerin varsayılan pencere açılma modunu belirlemek için kullanılır.
<link>	<link rel="stylesheet" type="text/css" href="style.css">	Html sayfası ile sayfa dışı bir döküman ya da kaynak arasında bağlantı kurmaya yarar.
<script>	<script>document.write("Merhaba")</script>	Html kodları içerisinde istemci taraflı çalışacak programlama komutlarını yazmak için kullanılır.
<style>	<style type="text/css"> body {background-color:yellow;} </style>	Html etiketlerini biçimlendirmek için stil tanımlamaları yapmayı sağlar.

<meta>	<ol style="list-style-type: none">1. <meta charset="UTF-8">2. <meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">3. <meta name="description" content="Html5 dersleri içeren web sitesi">4. <meta name="author" content="Fatih Çelik">5. <meta http-equiv="refresh" content="30">6. <meta name="viewport" content="width=device-width, initial-scale=1.0">	<ol style="list-style-type: none">1-Sayfanın desteklediği karakter grubunu tanımlar.2-Arama motorları tarafından bulunacak anahtar kelimeleri tanımlar.3-Siteyle ilgili kısa bir açıklama yapmak için kullanılır.4-Sayfanın tasarımcısına ait bilgi verilir.5- Sayfanın yenilenme aralığını belirlemeye yarar.6-Cihaz genişliğinin tamamını görüntülenebilir alan olarak belirler
--------	---	--

HTML Resim Ekleme Etiketi : etiketi ile sayfaya resim eklemek için kullanılır.

Örnek kullanım:

Src özelliği ile eklenecek resmin kaynak adresi gösterilir. Width ve Height ile resmin görüntülenecek boyutları belirlenir. Alt özelliği ise resmin gösterilememesi durumunda yerine gösterilecek bir bilgilendirme oluşturmak için kullanılır. Kullanılan resme ait hizalama işlemlerinin yapıldığı “align” özelliği HTML5 ile birlikte kullanılmamaktadır. Bu ve bunun gibi birtakım işlemler CSS kullanılarak gerçekleştirilebilmektedir.

Bilgi: img etiketi ile resim eklenirken resim için genişlik ve yükseklik değerlerinin verilmemesi durumunda tarayıcı resmin orjinal boyutlarını bilemeyeceğinden sayfanın yüklenmesi esnasında şablon boyutlarında değişiklikler olabilir.

Bilgi: Sayfada çok fazla resim kullanmak yükleme süresini artırabileceğinden resim kullanımı oldukça dikkatli biçimde yapılmalıdır. Ayrıca resim dosyalarının barındırıldığı sunucunun sürekli ulaşılabilir durumda olması resimler için sayfamızda kırık link hatası almamızı engellemesi açısından önemlidir.

Htm1 Listeleme Etiketleri : HTML listeleme etiketleri sıralı(ordered) ve sırasız(unordered) olarak ikiye ayrılmaktadır. Aşağıda her iki liste tipine ait kullanımlar gösterilmiştir.

Sıralı listeler(Ordered lists)	Sırasız listeler(Unordered lists)
<pre> Kahve Süt </pre>	<pre> Kahve Süt </pre>
<ol style="list-style-type: none">1. Kahve2. Süt	<ul style="list-style-type: none">• Kahve• Süt

Listelerin iç içe gelecek şekilde kullanımı aşağıda gösterilmiştir.

<pre> Kahve Çay Siyah çay Yeşil çay Çin çayı Afrika çayı Süt </pre>	<ul style="list-style-type: none">• Kahve• Çay<ul style="list-style-type: none">◦ Siyah çay◦ Yeşil çay<ul style="list-style-type: none">▪ Çin çayı▪ Afrika çayı• Süt
---	--

Bunların dışında aşağıda gösterilen biçimde açıklamalı listeler oluşturmak mümkündür.

<pre><dl> <dt>Kahve</dt> <dd>- Sıcak içecek</dd> <dt>Süt</dt> <dd>- Soğuk içecek</dd> </dl></pre>	<p>Kahve</p> <p>- Sıcak içecek</p> <p>Süt</p> <p>- Soğuk içecek</p>
---	---

Bilgi: Listeleme etiketlerinde “type” özelliğinin kullanımıyla sıralı listelerde numaralandırma tipinin değiştirilmesi mümkündür. Sırasız listelerde ise liste elemanlarına ait simgelerin değiştirilmesi HTML5 standartlarına göre CSS ile stil uygulanması sayesinde mümkün olabilmektedir.

Html Tablo Etiketleri : <table>, <tr>, <td> → Sık olmamakla birlikte içerik tablolarında kullanılabilir.

Html Form Etiketleri : <form>, <input> → Sayfada form üzerinden veri girişi için kullanılır.

<p>First name:</p> <input type="text" value="John"/> <p>Last name:</p> <input type="text" value="Doe"/> <input type="submit" value="Submit"/>	<pre><form action="/action_page.php"> <label for="fname">First name:</label>
 <input type="text" id="fname" name="fname" value="John">
 <label for="lname">Last name:</label>
 <input type="text" id="lname" name="lname" value="Doe">

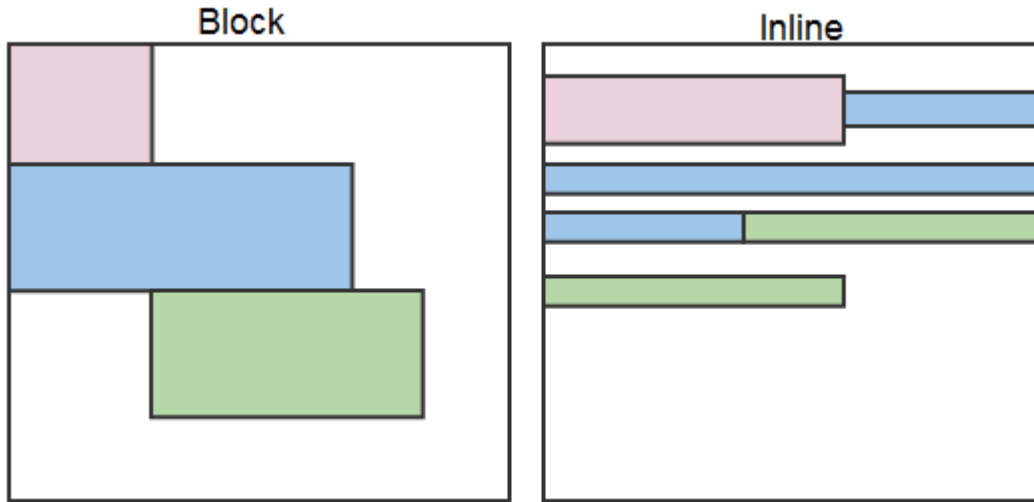
 <input type="submit" value="Submit"> </form></pre>
---	--

Html5 Multimedya Etiketleri:

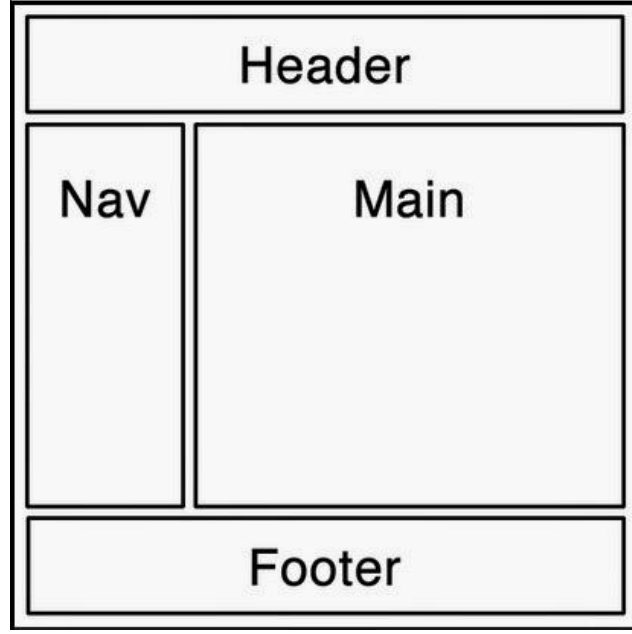
- <canvas> → Sayfada dinamik olarak grafik oluşturmak için çalışma alanı yaratır.
- <svg> → Sayfada vektörel grafikler için kapsayıcı bir alan oluşturur.
- <video> → Sayfaya dahili video ve oynatıcı eklemeyi sağlar.
- <audio> → Sayfaya dahili ses dosyası ve oynatıcı eklemeyi sağlar.
- <object> → Sayfaya harici medya eklentilerini gömmek için kullanılır.
- <embed> → Sayfaya harici medya eklentilerini gömmek için kullanılır.
- <iframe> → Sayfaya harici medya içeriklerini(youtube vs.) gömmek için kullanılır.

Html Şablon(Layout) Etiketleri :

(!)Önemli Bilgi: Html etiketleri sayfada dizilişlerine göre **block(altalta sıralı)** veya **inline(yatay sıralı)** olmak üzere ikiye ayrılır. Block etiketlere: <h1>, <p>, , , <div>, <table>, <tr> ; Inline etiketlere : <td>, <a>, etiketleri örnek olarak gösterilebilir.

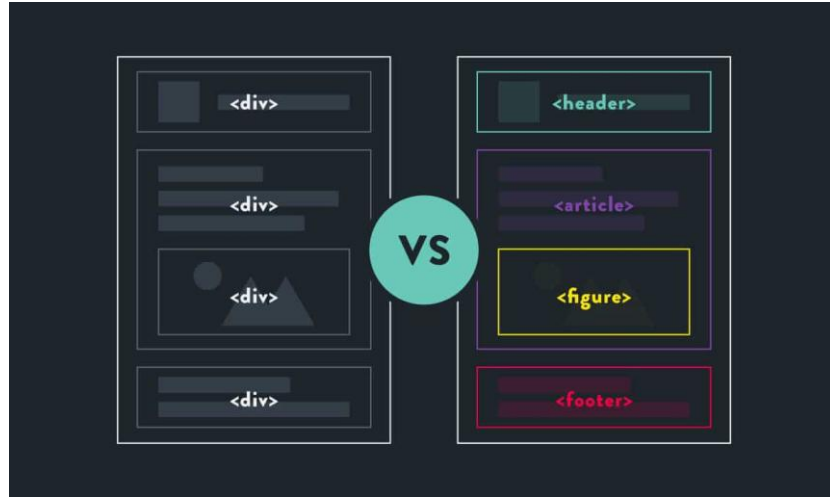


Bir web sayfası tasarlarlarken yapılması gereken öncelikle sayfaya ait bir iskelet oluşturmaktır. Sayfa iskeleti bir evin odaları gibi sayfada kullanılabilir alanları amaçlarına göre bölümlendirmek olarak düşünülebilir. Buna göre standart bir sayfa şablonuna ait iskelet (layout) aşağıdaki gibidir.



Sayfadaki ihtiyaca ve tasarıma göre bu alanlar çoğaltılabilir veya değişik ölçü ve konumlandırmalar yapılabilir. Ancak ana fikir hepsinde aynıdır,bölümlendirilen alanlar kenarlıklarla belirlenmiş olsun veya olmasın sayfadaki düzeni sağlamak maksadıyla bu işlem mutlaka yapılmalıdır.

Layout için bölümlendirilmiş alanları oluşturmada genel amaçlı “div” etiketleri veya özel görevli “Semantic(header,nav,section vs.)” nesneleri kullanılır.



Bu nesnelerle oluşturulan iskelet CSS ile **biçimlendirme,boyutlandırma ve konumlandırma** işlemlerinden geçirilerek istenilen sayfa tasarımını ortaya çıkarır.

3- Css ile Sayfa Tasarımı

3.1. CSS Kullanım Tipleri

CSS(Cascading Style Sheets) sayfa iskeletini oluşturan tüm HTML nesneleri için istenilen görsel stil özelliklerini tanımlamaya yaran bir dildir.

CSS kodları ilişkilendirileceği HTML nesneleriyle buluşmak üzere farklı yerlere yazılabilirler. Buna göre:

-**Yerel CSS** ile tanımlanacak stil özelliği direk olarak muhatabı olan HTML nesnesine style attribute tanımlamasıyla atanabilir.

Örnek: `<h1 style="color:red;">Kırmızı Başlık</h1>`

-**Genel CSS** ile head kısmında “<style>..<</style>” etiketleri arasında seçilen nesnenin stil özellikleri tanımlanır.

Örnek: `<style> h1{color:red;} </style>`

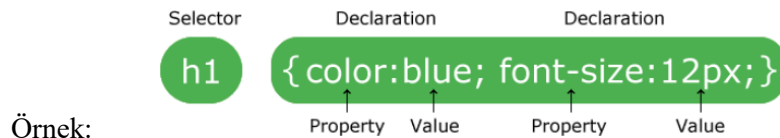
-**Harici CSS** ile head kısmında “<link>” etiketi ile harici stil sayfasına bağlantı yapılır.

Örnek: `<link rel="stylesheet" href="styles/stil.css">`

Bilgi: Aynı nesne için birden fazla CSS tanımı farklı şekillerde yapılmışsa öncelik olarak yerel stil tanımı gözönüne alınır. Sonrasında harici ve genel css komutları tarayıcının varsayılan sıralamasına göre algılanır.

3.2. CSS Seçiciler(Selectors)

CSS ile stil tanımlamada yazım kuralları(syntax) gereği öncelikle stil özellikleri tanımlanacak HTML nesnesinin seçilmesi ve ardından {..} parantezleri içine stil özelliklerinin tanımlanması gerekir.



Bu seçim işlemi için nesneyi tanımanın yani onu çağırırken “hoop filanca nesne sen bi bak hemşerim” demenin farklı yolları mevcuttur. Örneğin bir amele pazarında kalabalık bir gruptan işçi seçmek için boyacı/boyacılar gelsin diyebilir veya boyacıları usta-kalfa-çırak olarak sınıflandırıp sadece usta sınıfından boyacıları çağırabiliriz. Ya da direk nokta atışı “Ahmet usta” sen gel diyerek istediğimiz işçiye ulaşırız. Aşağıda CSS ile bu seçim işlemlerinin nasıl yapılacağı gösterilmektedir.

-ID(Kimlik) Seçici ile sayfadaki nesnelerden sadece birisi için eşsiz bir kimlik oluşturmak suretiyle sadece o nesneye özgü stil özellikleri tanımlanır.

Örnek: `#container{background-color: red;}`

-Class(Sınıf) Seçici ile sayfada ortak stil özelliklerine sahip olması istenen birden fazla nesne seçilerek ayrı ayrı stil tanımlı yapmak yerine nesneler sınıflandırılır.

Örnek: `.kutud{background-color: red;}`

-Tag(Etiket) Seçici ile stil tanımlaması yapılacak nesne kendi varolan tag adıyla çağırılarak o etiketin tüm mensupları için stil özellikleri atanır.

Örnek: `body{background-color: red;}`

Bilgi: * seçicisi ile sayfadaki tüm HTML nesneleri seçilir.

Bilgi: `h1,h2,p` şeklinde virgülle ayırarak aynı özellikleri atayacağımız nesneler için ortak bir seçici kullanılabilir.

Bilgi: `id` ve `class` isimlendirmede rakamla başlayan ifade kullanılamaz.

Gelişmiş CSS Seçicileri: Bu seçicilerle daha spesifik olarak istenilen nesne/nesnelerin elde edilmesi mümkün olabilmektedir. Daha önce verdiğimiz örnekte olduğu gibi bir inşaat için boyacı seçerken X şirket bünyesinde çalışmış olan boyacılar veya sadece Ahmet ustanın yanında çalışan çırak gibi bir seçim yapabiliriz. Aşağıdaki tabloda bazı spesifik(özel) CSS seçicileri gösterilmektedir.

Seçici	Örnek kullanımı	Açıklama
<code>element.class</code>	<code>p.kirmizi</code>	Kırmızı olarak sınıflandırılmış p nesnelerini seçer.
<code>element element</code>	<code>div p</code>	div altında tanımlı tüm p nesnelerini seçer.
<code>element>element</code>	<code>div > p</code>	div altında tanımlı yalnızca çocuk p nesnelerini seçer.
<code>element+element</code>	<code>div + p</code>	div tanımından hemen bir sonraki p nesnesini seçer.
<code>element1~element2</code>	<code>div ~ p</code>	div tanımından itibaren tüm p nesnelerini seçer.
<code>[attribute]</code>	<code>[target]</code>	target özneliğine sahip tüm nesneleri seçer.
<code>[attribute=value]</code>	<code>[target=_blank]</code>	target özneliğine _blank değeri atanmışları seçer.
<code>:hover</code>	<code>a:hover</code>	Mouse ile üzerine gelinen linkleri seçer.
<code>:focus</code>	<code>input:focus</code>	İmlecın üzerine odaklandığı form nesnelerini seçer.
<code>::after</code>	<code>p::after</code>	Her p nesnesinin ardından eklenecekleri seçer.
<code>::before</code>	<code>p::before</code>	Her p nesnesinden önce eklenecekleri seçer.
<code>:first-child</code>	<code>p:first-child</code>	Kapsayıcısı altındaki ilk çocuk olan nesneyi seçer.
<code>:last-child</code>	<code>p:last-child</code>	Kapsayıcısı altındaki son çocuk olan nesneyi seçer.
<code>:nth-child(n)</code>	<code>p:nth-child(2)</code>	Kapsayıcısı altında ikinci çocuk olan nesneyi seçer.

Bilgi: HTML nesnelerinin sayfadaki dizilişi hiyerarşik bir yapıya sahiptir ve iç içe yazılan bu nesneler arasında kapsayıcılık açısından **ebeveyn-çocuk(parent-child)** ilişkisi bulunur. **DOM(Document Object Model)** adı verilen bu yapıda HTML nesneleri bir aile soy ağacına benzer şekilde **node** adı verilen dallara ayrılır şekilde dizilmektedir.

3.3. CSS Özellikleri

CSS Biçimlendirme Özellikleri:

- Type(Yazı tipi) Özellikleri:
 - Font-family: Karakter grubu seçimi yapar.
 - Font-size: Yazı boyutunu belirler.
 - Font-weight:Yazı kalınlığını belirler.
 - Font-style: Yazı stilini(italic,normal,oblique) belirler.
 - Line-height:Satır yüksekliğini ayarlar.
 - Text-transform: Harflerin büyük,küçük olma durumunu belirler. (capitalize vs.)
 - Text-decoration: Yazıların altı çizili vs. olma durumunu(underline,overline,line-through,blink,none) belirler.
 - Text-shadow: Yazı gölgesi oluşturur.
 - Color: Yazı rengi belirler.
- Background(Arkaplan) Özellikleri:
 - Background-color: Arkaplan rengi belirler.
 - Background-image: Arkaplan resminin/resimlerinin* kaynağını belirler.
 - Background-repeat: Resim tekrarını belirler.(no-repeat,repeat,repeat-x,repeat-y)
 - Background-position: Arkaplan resmi için X - Y eksenlerine göre konum belirler.
 - Background-attachment: Arkaplan resminin zemine sabitlenmesi veya scroll ile sadece konumlandığı yere gelindiğinde görünmesini sağlar.
 - Background: Tüm arkaplan özelliklerinin topluca tanımlanabilmesini sağlar.
 - Background-size: Arkaplan resmini yeniden ölçeklendirir. (cover,contain,10%)
 - Background:linear-gradient(yellow,blue); arkaplanda renkler arası geçiş yapar.
- List(Liste) Özellikleri:
 - List-style-type: Liste simge tipi(none,square vs.)
- Border(Kenarlık) Özellikleri:
 - Border-style: Kenarlık çizgi stilini belirler. (solid,dotted,dashed vs.)
 - Border-width: Kenarlık kalınlığını belirler. (1px,thin,thick vs.)
 - Border-color: Kenarlık rengini belirler. (black,#000 vs.)
 - Border: Tüm kenarlık özelliklerini tek satırda tanımlar.(border:thin solid black;)
 - Border-radius: Kenarlık köşelerini yuvarlar. (border-radius: 0 5px 10px 20px;)
- Block(Hizalama) Özellikleri:
 - Display: Nesne algılanma,sıralanma biçimi(block,inline,none vs.)
 - Text-align: Yazı hizalama(left,center vs.)
- Diğer Özellikler:
 - Visibility: Nesneyi gizlemeyi sağlar.(hidden)
 - Opacity: Nesne saydamlığını belirler.(0.5 vs.)
 - Overflow: Yüksekliği tanımlanmış blok nesnelerde taşma olması durumunda içeriğin nasıl görüntüleneceğini belirler.(visible,scroll,hidden,auto)
- CSS Harici Fontları: Google CDN üzerinden siteye harici fontlar eklemek mümkündür. (`<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">`)
- CSS İkonları: fontawesome.com sitesi üzerinden siteye harici ikon setleri yüklenebilir. (`<script src="https://kit.fontawesome.com/yourcode.js"></script>`)

Bilgi: CSS ile belirlenecek renkler sözel olarak veya RGB,HEX,HSL gibi değişik formatlarda atanabilir.

CSS Boyutlandırma Özellikleri:

Width(Genişlik) ve Height(Yükseklik) değerleri ile nesne boyutlandırılır. Boyutlandırma piksel(px),punto,mm,cm gibi kesin(absolute) birimler ya da em,rem,%,vh,vmax,initial,inherit veya auto cinsinden bağıl(relative) değerlerle yapılabilir. Aşağıdaki tablo bağıl(relative) değerleri göstermektedir.

Birim	Açıklama	Endeks
em	Tanımlandığı nesnenin font değerine bağlıdır. (16px ebatında bir nesne için 2em 32px'dir.)	Yazı
ch	“0” karakterinin genişliğine bağlıdır.	
rem	“root” nesnenin yani html dökümanının font değerine bağlıdır.	
vw	Tarayıcıdaki görüntülenebilir alan(viewport) genişliğinin 1% değerine eşittir.	Ekran
vh	Tarayıcıdaki görüntülenebilir alan(viewport) yüksekliğinin 1% değerine eşittir.	
vmin	Tarayıcıdaki görüntülenebilir alan(viewport) genişliğinin 1% değerine eşittir. Fakat alan küçülürse nesnenin boyutu da otomatik olarak küçülür.	
vmax	Tarayıcıdaki görüntülenebilir alan(viewport) genişliğinin 1% değerine eşittir. Fakat alan büyürse nesnenin boyutu da otomatik olarak büyür.	
%	Nesne ölçüsü kapsayıcı nesnesinin(parent) ölçüsüne bağlıdır.	Parent
fr	CSS Grid sisteminde kapsayıcı nesneye göre boyutlandırmada alınan payı ifade eder.	

Bilgi: em ve rem birimleri genelde esnek/relative ölçekli tasarımlar oluşturmak için uygundur.

Bilgi: viewport değeri 50 cm genişlikte ise 1vw=0.5cm'ye eşit olur.

Bilgi: max-width,min-height gibi özellikler kullanılarak nesne boyutlarına alt üst sınırlar belirlenebilir.

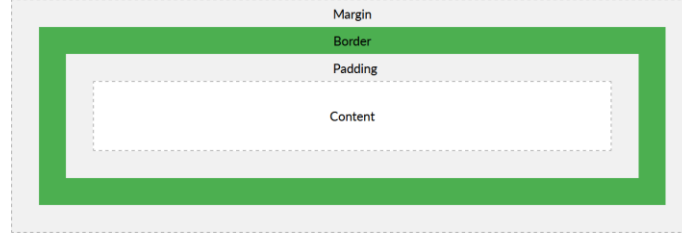
CSS Konumlandırma Özellikleri:

- Position(Konumlandırma) Değerleri:
 - Relative:Sayfada bir nevi bir referans veya merkez noktası olarak belirlenmesi istenen nesneye,katmana uygulanır. Diğer nesneler,katmanlar bu referans noktasına göre konumlandırılır. Sayfa şablonu oluştururken en dış seviyede buluna kapsayıcı katman(genel katmanı) relative olarak konumlandırılmalıdır.
 - Absolute: Bir katman veya nesne sayfada relative olarak konumlandırılmış referans noktasına göre veya bir üst seviyesinde kapsayıcı nesnesine göre mutlak konumlandırmayla yerleştirilecekse bu değer seçilir. Absolute konumlandırma yapılan nesne referans noktasına göre sağ(right),sol(left),üst(top),alt(bottom) olmak üzere iki kenara hizalama değeri verilerek yerleştirilir.
 - Static: Nesne normal sayfa akışında olması gereken yere konumlanır.
 - Fixed: Nesne viewport(tarayıcı alanı) içerisinde sürekli belirtilen yerde sabit kalır,sayfa aşağı yukarı kaysa da nesnenin yeri değişmez.
 - Sticky: Nesne scroll pozisyonuna göre belirtilen pozisyona yapışık halde kalır.
 - Z-index: Pozisyon değeri atanmış nesneler arasında normal akıştaki öncelik sırasının değiştirilebilmesini sağlar.(z-index:99; olan 98 olanın üstündedir.)

Bilgi: HTML DOM hiyerarşisinde en iç seviyede tanımlanan nesnenin görünme önceliği en fazladır.

CSS Kutu Modeli Özellikleri:

Nesnelerin birbirlerine ve çerçevelere olan mesafeleri ve konumları sağdan,soldan,alttan,üstten olmak üzere dört taraftan belirlenebilir. Float ve Clear değerleri ile nesneleri diziliş sırası altalta,soldan sağa,sağdan sola şeklinde belirlenir.



- Padding: Nesnenin çerçevelere olan mesafesini belirler.
- Margin: Nesnelerin birbirlerine olan mesafelerini belirler.
- Float: Sayfa içerisinde nesneler,örneğin liste elemanları alt alta dizilir. Bu dizilişi değiştirmek ve soldan sağa yapmak için float değeri "left",sağdan sola yapmak için float"right" olarak seçilir. Bu diziliş sıfırlanıp normal alt alta dizilime dönmek istendiği zaman Clear özelliği kullanılır.
- Box-sizing: Normalde tarayıcılar tarafından kenarlık ölçüleri toplam genişliğe eklenerek hesaplanır. İstenirse border değerleri "box-sizing:border-box;" seçilerek kenarlık değerleri width değerinden hariç tutulabilir,border değışse de toplam genişlik etkilenmez.

Bilgi: CSS kodları içerisinde açıklama yazmak için bkz: `/* Bu bir açıklama satırıdır. */`

Gelişmiş CSS Özellikleri:

CSS önceleri sayfadaki nesnelerin görünüm özelliklerini düzenlemekten ibaretken yapılan geliştirmelerle birlikte sayfada animasyonlu işlemlerin de yapılabilmesine olanak sağlayan ve Flash gibi harici hareket işleme programlarına gerek duymadan sayfayı interaktif bir yapıya kavuşturabilir hale gelmiştir.

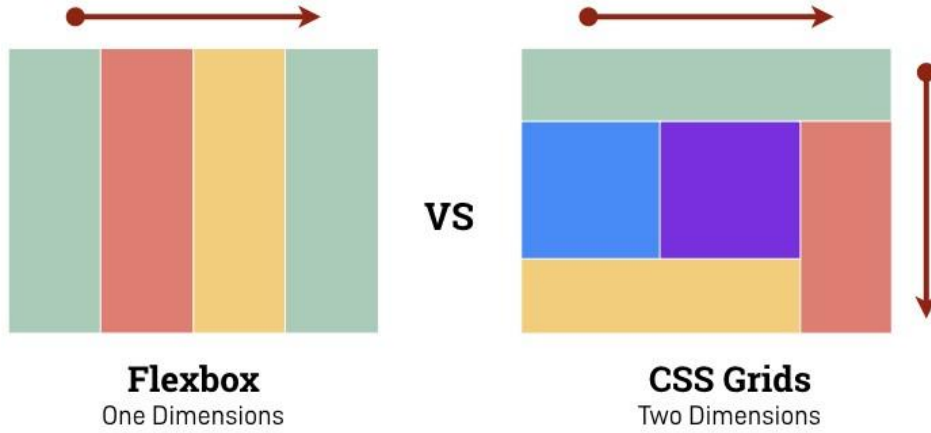
- CSS Transforms: Nesnenin 2D veya 3D olarak döndürülmesi işlemidir.
 - translate() → nesneyi sürükler.
 - rotate()→ nesneyi döndürür.
 - scale()→ nesneyi boyutlandırır.
 - skew()→ nesneyi çarpıtır.
 - matrix()→ tüm transform işlemlerini bir arada yapar.
- CSS Transitions: Nesnenin boyutlarında yapılan değişimle geçiş sağlanmasıdır.
 - transition: width 2s, height 4s; → yeniden boyutlara geçiş yapar.
- CSS Animations: Nesnenin yer değiştirmesiyle anime edilmesidir.
 - @keyframes → animasyon geçiş değerleri belirlenir.
 - animation-name → @keyframes de tanımlı animasyon adını çağırır.
 - animation-duration → animasyon süresi tanımlanır.
 - animation-delay → animasyon için gecikme süresi belirlenir.

3.4. CSS Şablon(Layout) Oluşturma Yöntemleri :

HTML ile hazırlanmış sayfa iskeleti CSS ile 4 farklı yöntemle layout haline getirilebilir.

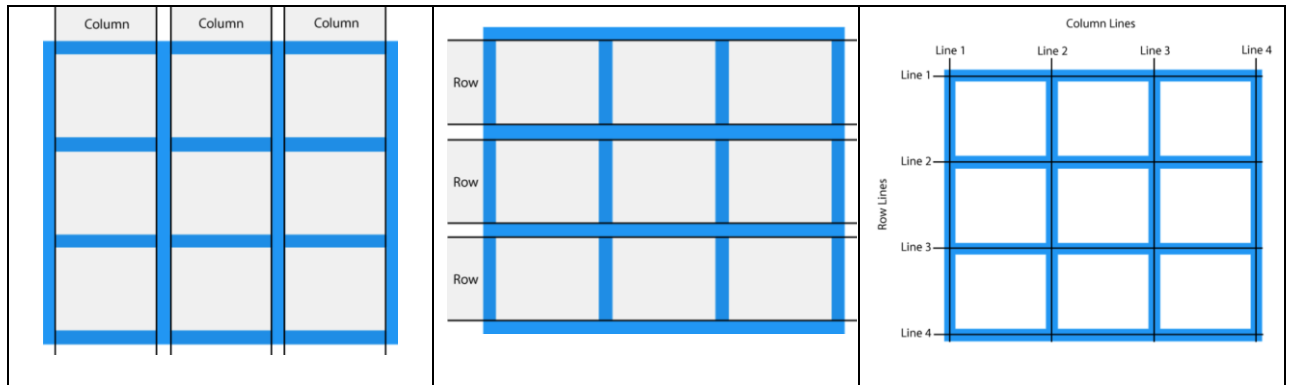
- Hazır CSS framework ile layout oluşturma(Bootstrap vs.) (Mobil uyumlu/Responsive)
- CSS float(left/right) özelliği ile nesne akışını belirleme
- CSS position(absolute/relative) özelliği ile nesne konumunu belirleme
- CSS Flexbox ile esnek boyutta ve dizilişte kutular oluşturmak(Mobil uyumlu/Responsive)
- CSS Grid ile ızgara yapısında sayfayı bölümlendirmek(Mobil uyumlu/Responsive)

Bilgi: Bu yöntemler arasında mobil uyumluluk ve verim açısından günümüz teknolojisine en uygun olanları Flexbox ve Grid yöntemleridir.



CSS Grid Yapısı:

CSS Grid yöntemi sayfayı iki boyutlu parselleyerek enine ve boyuna grid çizgileriyle ızgara şeklinde bölümlendirmeyi sağlar. Grid sisteminde ızgara hücreleri sütün çizgileri(column-line) ile oluşan sütunlar(grid-column) ve satır çizgileri(row-line) ile oluşan satırların(grid-row) kesişimiyle ortaya çıkar.



Bu sayede oluşan bölümlerle sayfada responsive yapıda esnek sayfa şablonları(layout) oluşturabilmek mümkün olmaktadır. CSS Grid sistemi sadece layout oluşturmada değil sayfa içinde resim galerisi, ona giriş formları gibi birçok modülün kullanımında büyük rahatlık sağlar.

CSS Grid sisteminin temel işleyişinde bir kapsayıcı parent nesne içerisine ızgaraları oluşturan çocuk nesneler yerleştirilir. Genelde bu nesneler için “div” etiketi kullanılmakla birlikte layout tasarımında “Semantic” etiketlerini de kullanmak mümkündür.

1	2	3
4	5	6
7	8	9

Yukarıdaki gibi bir grid yapısını oluşturabilmek için toplamda 1 adet kapsayıcı, 9 adet de çocuk olmak üzere toplam 10 nesne kullanılmıştır. 9 çocuk nesne 3 sütun ve 3 satıra yerleşecek şekilde dağıtılmıştır.

```
<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>
```

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
</style>
```

Grid sisteminde önce kapsayıcı nesne seçilerek grid yapısının temel özellikleri tanımlanır , daha sonra çocuk nesnelere ait özellikler belirtilir.

CSS Grid Kapsayıcı Özellikleri:

- display: grid; → Sayfada nesnelerin görünüm/yerleşim modunu grid olarak ayarlar.
- grid-template-columns: 100px 1fr 2fr 4fr → Grid içerisinde 4 adet sütun tanımlar.
- grid-template-rows: repeat(4,auto); → Grid içerisinde 4 adet satır tanımlar.
- grid-template-areas: “header header header”; → Grid içerisinde sözel olarak alanlar haritalar.
- grid-gap: 10px; → Grid hücreleri arasında boşluk oluşturmak için kullanılır.

Bilgi: CSS Grid sisteminde sütun ve satır tanımlamak için sayısal olarak (px,%) gibi birçok ölçü birimi kullanılabilir. ”fr” ölçü birimi ise grid sistemine özel orantısal olarak sütun ve satır boyutlarının kapsayıcıya endeksli olarak dağıtılmasını sağlar.

Bilgi: CSS Grid sisteminde aynı ölçüde birden fazla sütun veya satır oluşturulmak istendiğinde “repeat” fonksiyonu kullanılabilir.

Bilgi: CSS Grid sisteminde genelde layout tasarımları yapılırken grid alanlarının yerleşimi için “grid-template-areas” özelliği ile sözel haritalama/yerleşim planı yapılır.

CSS Grid Çocuk Nesne Özellikleri:

- grid-column: 1/5; → Sütunun başlangıç ve bitiş çizgisini belirler.
- grid-row: 1/ span 2; → Satırın başlangıç çizgisini ve satır birleştirme miktarını belirler.
- grid-area: ustAlan; → Sayfa haritalamasındaki yerleşimi için çocuk nesneyi etiketler.

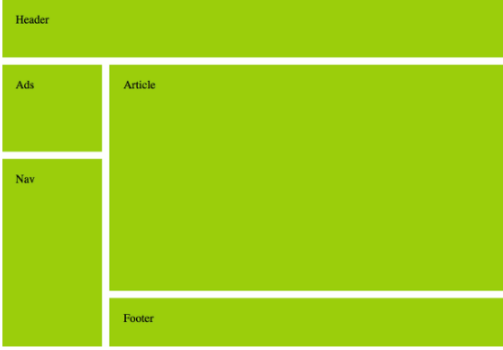
Bilgi: CSS Grid sisteminde olağan DOM akış sırasında ilk sırada bulunan bir nesne ilk sırada görünmek zorunda değildir. grid-column,grid-row ve grid-area özelliklerine atanan değerler sıralamayı belirler.

CSS Grid Hizalama Özellikleri:

Grid içerisinde bulunan nesneleri yatay ve dikey ekseninde hizalamak için kapsayıcıya şu özellikler atanır:

- justify-content: space-evenly/space-around/space-between/center/start/end; → Yatayda hizalama
- align-content: space-evenly/space-around/space-between/center/start/end; → Dikeyde hizalama

CSS Grid ile Layout Uygulaması:

	<p>Açıklama: "container" id li kapsayıcı div içerisindeki çocuk nesneleri "grid-area" ile etiketlenerek sayfa yerleşimindeki yerleri "grid-template-areas" özelliği altında tanımlanır.Sütun ve satır ölçüleri "grid-template-columns" ve "grid-template-rows" özellikleri altında belirlenir.</p>
---	---

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Grid Örnek 8</title>
  <style>
    #container{
      display: grid;
      grid-template-areas:
        "header header"
        "ads article"
        "nav article"
        "nav footer"
      ;
      grid-template-columns: 1fr 4fr;
      height: 1000px;
      grid-template-rows: 100px 1fr 3fr 100px;
      grid-gap: 10px;
    }
    .kutu1{
      grid-area: header;
    }
    .kutu2{
      grid-area: ads;
    }
    .kutu3{
      grid-area: article;
    }
    .kutu4{
      grid-area: nav;
    }
    .kutu5{
      grid-area: footer;
    }
    #container > div{
      background-color: indigo;
    }
  </style>
</head>
<body>
  <div id="container">
    <div class="kutu1">Header Alanı</div>
    <div class="kutu2">Reklam Alanı</div>
    <div class="kutu3">İçerik Alanı</div>
    <div class="kutu4">Menü Alanı</div>
    <div class="kutu5">Footer Alanı</div>
  </div>
</body>
</html>
```


CSS Flexbox Yapısı:

CSS Flexbox yöntemi sayfadaki nesnelerin akışını yatay(row) yönünde ya da dikey(column) yönünde tek yönlü olarak sağlayabilir. Esnek yapısı sayesinde özellikle paragraf,resim,link vs. gibi içeriklerin sayfa içi yerleşiminde kullanılması sağlıklı ve değişik ölçülerle uyumlu sonuçlar doğurur.

CSS Grid yönteminde olduğu gibi flex akış ve diziliş özellikleri de bir kapsayıcı nesne üzerinden sağlanır. CSS Grid'den farklı olaraksa çocuk nesnelerin kapsayıcı içerisindeki genişlik/yükseklik/genleşme gibi değerleri kapsayıcı üzerinden değil çocukların kendi seçicileri üzerinden belirlenir.



CSS Flexbox Kapsayıcı Özellikleri:

- display: flex; → Sayfada nesnelerin görünüm/yerleşim/akış modunu flex olarak ayarlar.
- flex-direction: row/column/row-reverse/column-reverse; → Nesnelerin akışını yatay ya da dikey olarak ayarlar.
- flex-wrap: wrap/nowrap; →Nesnelerin akış yönündeki alana sığmaması halinde sonraki satır/sütundan devam ederek sıkışmamasını(wrap) ya da orayla sınırlı kalmasını(nowrap) sağlar.
- flex-flow: row wrap; → Yön ve sıkışma özelliklerinin ikisini birarada kullanmayı sağlar.
- justify-content: flex-start/flex-end/center/space-between/space-around; → Kapsayıcı içindeki çocuk nesnelerin akış yönüne göre ortaya çıkan yatay eksende hizalanmasını sağlar.
- align-items: flex-start/flex-end/center /baseline/stretch; → Kapsayıcı içindeki çocuk nesnelerin akış yönüne göre ortaya çıkan dikey eksende tek satır/sütun kaplaması halinde hizalar.
- align-content: flex-start/flex-end/center/space-between/space-around/stretch; → Kapsayıcı içindeki çocuk nesnelerin akış yönüne göre ortaya çıkan dikey eksende birden fazla satır/sütuna yayılması halinde hizalamalarını yapar.

CSS Flexbox Kapsayıcı Özellikleri:

- flex-grow: 1; → Çocuk nesnenin kapsayıcının alanına göre genişleme katsayısını belirler.
- flex-shrink: 0; → Çocuk nesnenin kapsayıcının alanına göre küçülme katsayısını belirler.
- flex-basis: 300px; → Çocuk nesnenin akış yönüne göre genişlik/yükseklik başlangıç değeridir.
- flex: 1 0 300px; → Üç değer birarada kullanımıdır. 0 değeri kullanılırsa büyüme/küçülme olmaz. 1 değeri kullanılırsa büyüme/küçülme otomatik gerçekleşir.
- order: -1; → Çocuk nesnenin olağan DOM dizilişindeki yerini değiştirir. -1 değeri en başa taşır.

- align-self: flex-start/flex-end/center; → Çocuk nesnenin bağımsız olarak kendini hizalamasıdır.

```
/* Sözel değerler: initial başlangıç değerlerini taşır. */
flex: auto;
flex: initial;
flex: none;

/* Tek değer rakam olarak kullanılırsa işaret edilen özellik: flex-grow */
flex: 2;

/* Tek değer genişlik/yükseklik ölçü birimleri kullanılırsa: flex-basis */
flex: 10em;
flex: 30%;
flex: min-content;

/* Bu şekilde iki değer kullanılırsa: flex-grow | flex-basis */
flex: 1 30px;

/* Bu şekilde iki değer kullanılırsa: flex-grow | flex-shrink */
flex: 2 2;

/* Bu şekilde üç değer kullanılırsa: flex-grow | flex-shrink | flex-basis */
flex: 2 2 10%;

/* Global değerler: inherit nesneye ait kalıtsal değerleri taşır */
flex: inherit;
flex: initial;
flex: unset;
```

Bilgi: Flexbox yapısında bir nesne hem kapsayıcı hem de kapsanan özelliklere aynı anda sahip olabilir.

CSS Flexbox ile Galeri Uygulaması:



Açıklama: "kapsayıcı" classlı div içerisindeki çocuk "img" resim nesneleri flex yapısında "flex-direction" ile satır yönünde dizilerek satıra sığmaması halinde alt satırdan devam etmesi "flex-wrap" ile sağlandı. Kapsayıcı alanın büyümesi ve küçülmesine bağlı olarak çocuk nesnelerin durumu ".kapsayıcı > img" seçicisi altında "flex" özelliği altında tanımlandı. "max-width" ve "min-width" değerleri ile resim ebatlarına alt üst sınırlar atandı.

```
<div class="kapsayici">
  
  
  
  
  
  
  
  
  
  
  
  
</div>
```

```
.kapsayici{
  /*Flexbox kapsayıcı özellikleri */
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  justify-content: center;

  background-color: rgb(226, 216, 216);
  width: 90%;
  height: auto;
  margin: auto;
}
.kapsayici > img{
  /*Flexbox çocuk özellikleri */
  flex: 1 1 300px;

  max-width: 400px;
  min-width: 200px;
  margin: 25px;
}
```

CSS Flexbox ile Tanıtım Kartı Uygulaması:



Açıklama: "kapsayici" classlı div içerisindeki çocuk "kart" div nesneleri flex yapısında "flex-direction" ile satır(row) yönünde dizildi. Kapsayıcı alanın büyümesi ve küçülmesine bağlı olarak çocuk nesnelerin durumu "kart" seçicisi altında "flex" özelliği altında tanımlandı. "max-width" ve "min-width" değerleri ile kart ebatlarına alt üst sınırlar atandı.

Ayrıca "kart" nesneleri de aynı zamanda kapsayıcı olarak ayarlandı ve içerisindeki "img,h3,p,a" nesneleri "flex-direction" ile sütun(column) yönünde dizildi.Çocuk nesnelerin hizalanması "align-items:center" ile ortalananacak şekilde ayarlandı.

* Örneğe ait kaynak kodlara erişmek için gidiniz. → <https://github.com/webmucidi/CssFlexExamples.git>

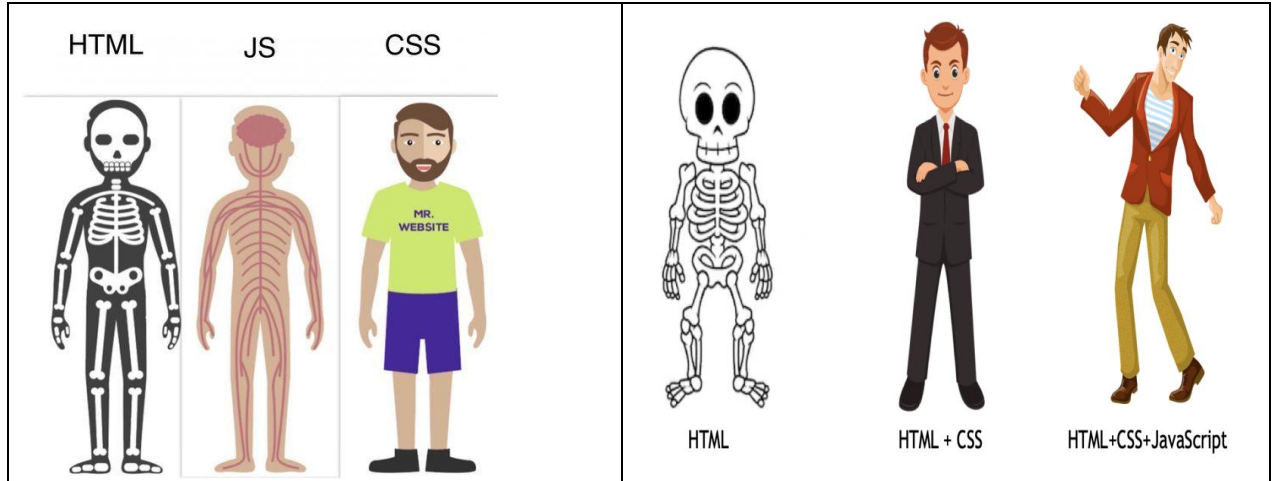
** Araştırma: GitHub nedir,ne işe yarar,nasıl kullanılır? Araştırınız ve projenizi GitHub'a yükleyiniz.

4- JavaScript İstemci Taraflı Web Programlama

4.1. JavaScript'in Tanımı ve İşleyişi

HTML ve CSS ile oluşturulan web sayfasında etkileşimli işlemleri yapabilmek için mevcut HTML nesnelerini ve CSS stil özelliklerini manipüle etmeye(değiştirmeye/düzenlemeye) yarayan istemci taraflı dil JavaScript programlama dilidir. Bir web sitesinin yapısını bir evin yapım/inşaat sürecine benzetirsek binanın kaba inşaatını(demir,çimento,tuğla vs.) yaptığımız gibi HTML nesneleri(div,p,a vs.) ile sayfanın temel yapısını oluştururuz. Evin daha güzel görünmesi için yaptığımız badana boya işlemlerinde olduğu gibi sayfa için CSS(color,background vs.) ile görsel özellikleri düzenleriz. Son olarak evin kullanılmaya hazır durumda olabilmesi elektrik,ısıtma vs. aksamının düzenlenmesi gerekir. Bir web sayfasında işlevselliği ve etkileşimi sağlayacak işlemler ise JavaScript gibi web programlama dilleri ile mümkün kılınmaktadır.

Bir başka örnekle açıklamaya çalışalım:



Yukarıdaki şekillerden de anlaşılacağı gibi insan vücudunun temel yapısı olan iskeleti HTML olarak düşünersek, bu iskeletin ete kemiğe bürünmüş ve görsel özelliklere kavuşmuş hali CSS ile mümkün olur. Fakat tüm bunların yanında esas vücudumuzu işlevsel kılan sinir sistemimizdir, bir sitenin işlevsel ve etkileşimli hale gelmesini sağlayan şey de JavaScript gibi bir programlama dilidir.

JavaScript programlama dilinde işleyiş sayfada bir olayın tetiklenmesi ile başlayarak, nesnenin seçilmesi ve o nesne üzerinde gerçekleştirilecek işlemlerin tanımlanması şeklinde olur.

Olay tanımı (butona tıklama-onclick)	Nesne seçimi (ID üzerinden seç-getElementById)	İş (function)
Karnım acıktı!	Ekmeği al!	Ye!

Bilgi: JavaScript kodları .js uzantılı harici bir sayfaya yazılıp `<script src="kodlar.js">` ile bağlanarak veya HTML sayfası içinde tercihen body kapanışından sonra `<script>..</script>` etiketleri içerisine yazılarak kullanılır.

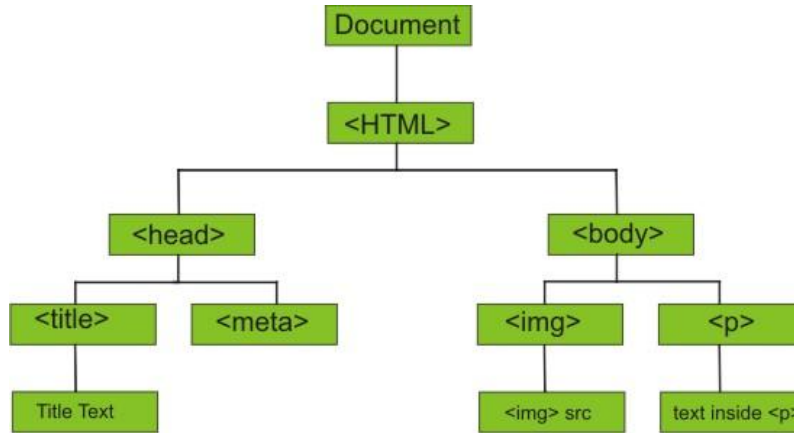
4.2. JavaScript Olayları(Events)

Window(Pencere),Form,Mouse(Fare),Keyboard(Klavye),Drag(Sürüklenme),Media,Pano ve diğer kategori dışı seçeneklerle gruplandırılır. Bunlardan en sık kullanılan bazıları aşağıdaki gibidir.

onclick	Mouse olayı	Fare sol tuşuna bir kez tıklanma
onload	Window olayı	Sayfanın başarıyla yüklenmesi
onchange	Form olayı	Seçimin değişmesi
onsubmit	Form olayı	Formun sunucuya aktarılması
onfocus	Form olayı	Input içine imlecin odaklanması
onkeydown	Keyboard olayı	Klavyeden bir tuşa basılması
onscroll	Drag olayı	Scroll ile kaydırma
oncopy	Pano olayı	Kopyalama yapılması
onplay	Media olayı	Media içeriğinin başlaması

4.3. JavaScript Nesne Seçimi

Mevcut HTML nesneleri veya onların CSS özellikleri üzerinde işlem yapabilmek için bu nesneleri elde etmek HTML DOM yapısı üzerinden mümkün olmaktadır. DOM(Document Object Model) yapısı bir ağaç tipindedir ve en tepede document nesnesi yani sayfanın kendisi bulunur. Diğer nesnelere iç içe HTML kod yapısında aşağıdaki şemada görüldüğü gibi ulaşılır.



Örnek: “document.html.body.img.src “ dizilişi ile resmin kaynağına ulaşılabilir.

Yukarıdaki mantıksal diziliş JavaScript syntaxı(yazım kuralları) altında yazılırken bazı hazır fonksiyonlar ile gerçekleştirilir. Nesneleri elde edebilmek için kullanılan bazı fonksiyonlar şunlardır:

- document.getElementById(“baslik”) → ID üzerinden nesne yakalama

- `document.getElementsByClassName("icerik")[i]`→ Class adı üzerinden nesneler yakalama
- `document.getElementsByTagName("p")[i]`→ Tag adı üzerinden nesne yakalama
- `document.getElementsByName("grup")[i]`→ Gruplanan nesneleri yakalama
- `document.querySelector(".link")`→ Seçici ile eşleşen ilk nesneyi yakalama
- `document.querySelectorAll("#resim")`→ Seçici ile eşleşen tüm nesneleri yakalama

4.4. JavaScript İş ve İşlemleri(Functions)

Tanımlanan olay üzerinden nesne yakalandıktan sonra yapılacak iş ve işlemler fonksiyon blokları altında tanımlanır. Örnek bir fonksiyon tanımı `function islem(){var sonuc=x+y;}` şeklindedir. Fonksiyon içinde istenen işlemleri gerçekleştirebilmek için aşağıdaki gibi bir algoritmanın adımları izlenerek sonuca ulaşılır.

1. Başla.
2. Değişkenleri tanımla.
3. Değişkenlere değer aktar.
4. Gerekliyse şart kontrolü yap.
5. Varsa döngüyü çalıştır.
6. Hesaplamaları yap.
7. Sonucu yazdır.
8. Bitir.

Yukarıdaki gibi algoritma adımlarını izlerken problemin çözümünde kullanılacak sayfa üzerinden aktarılan veriler ve elde edilen sonuçlar bellek üzerinde değişken adı verilen yapılarda saklanır. Değişken tanımlanırken diğer programlama dillerinde olduğu tip bildirimi önceden yapılmak zorunda değildir. Değişken tanımlama ifadesi **var,let,const** olmak üzere üç şekilde bildirilir. Değişken bildirimi yapıldıktan sonra içeriğinde Türkçe ve özel karakter barındırmayacak,boşluk olmayacak,rakamla başlamayacak ve javascript kütüphanesine ait anahtar ifadelerle aynı olmayacak biçimde bir değişken adı tanımlanır.(Örnek: `var sayi1=6;`) Değişken isimleri büyük-küçük harf duyarlıdır. Değişken içinde saklanan verilere önceden bir tip bildirimi yapılmamasına rağmen bu durum verilerin bir tipi olmadığı anlamına gelmez. Temelde sayısal ve sözel olmak üzere iki tip veri vardır. (`var sayisal=10;var sozel="Ali";`) Bunlar arasında daha sonra tip dönüşümü yapmak mümkündür.

Birden fazla verinin aynı anda tek bir değişkende tutulabilmesi için dizi(array) tipinde değişkenler oluşturulabilir.(Örnek: `var isimler=["Ahmet","Ali","Ayşe","Fatma"];`)

Değişkenlere aktarılan veriler üzerinde işlemler yapabilmeyi operatörler sağlar. En sık kullanılanları `→`(= aktarma, `==` eşitlik kontrolü, `<` küçüktür, `>` büyüktür, `+` `-` `*` `/` dört işlem, `++` bir artırma, `--` bir azaltma, `**` kare alma, `%` bölmeden kalanı bulma, `&&` ve `,` `||` veya `!` değil)

Bunun dışında çoğu temel işlem için JavaScript kütüphanesi birçok hazır fonksiyon barındırır. Aşağıda amaçlarına göre belirtilmiş bazı hazır fonksiyonlar ve özellikler belirtilmiştir.

- `Alert("Merhaba Dünya!");` → Açılır pencerede yazdırma fonksiyonu

- `Document.getElementById("baslik").innerHTML="Merhaba Dünya";` → HTML nesne içeriğine yazdırma
- `Document.write("Merhaba Dünya!");` → Sayfada uygun yere yazdırma
- `Console.log("Merhaba Dünya!");` → Tarayıcı console ekranına yazdırma
- `Length` özelliği → Seçilen liste/dizinin uzunluğunu/kapasitesini bulur.
- `String(veri)/veri.toString()` → Seçili nesneye ait değişken tipini sayıdan metine dönüştürür.
- `toFixed(2)` → Değişken içerisindeki değerin virgülden sonraki basamak sayısını belirler.
- `Number()` → Değişken içerisindeki değeri sayıya dönüştürür.
- `parseInt()` → Değişken içerisindeki değeri tam sayıya dönüştürür.
- `parseFloat()` → Değişken içerisindeki değeri kesirli sayıya dönüştürür.
- `Var tarih=new Date();` → Mevcut sistem saat ve tarih bilgisini elde etmeyi sağlar.
- `getFullYear()` → Elde edilen tarih/saat bilgisinden sadece yılı ayıklar.
- `getDay()` → Elde edilen tarih/saat bilgisine göre haftanın kaçınıcı günü olduğunu bulur.
- `getMonth()` → Elde edilen tarih/saat bilgisine göre ayın kaçınıcı günü olduğunu bulur.
- `var isimler=["Ahmet","Ali","Ayşe","Fatma"];` `isimler.push("Mehmet");` → Diziye eleman ekler.
- `createElement("h3")` → Yeni bir HTML nesnesi oluşturur.
- `document.body.appendChild(nesne);` → Oluşturulan nesneyi çocuk olarak ekleme
- `document.getElementById("baslik").classList.add("kirmizi");` → Varolan class stil özelliklerini seçili nesneye atamayı sağlar.
- `document.getElementById("baslik").style.color=red;` → Direk stil özelliği ve değeri atar.
- `var rastgeleSayi=Math.floor(Math.random() * 4);` → rastgele 0 dan 4 e kadar sayı oluşturur.
- `document.getElementById("myBtn").addEventListener("click", displayDate);` → Seçili "myBtn" nesnesi üzerinde "click" olayı gerçekleşirse "displayDate" isimli fonksiyonu çalıştırır.
- `document.querySelectorAll('#urunListesi option').forEach(option => option.remove());` → Tüm seçenekleri elde edip her bir seçenek için seçeneği silme işlemi yap.

Bu hazır fonksiyonlar dışında bir problemin çözüm aşamasında şarta bağlı özel durumları kontrol etmek veya aynı işi birden fazla kez tekrar etmek gerekebilir. Bu tip durumlarda tanımladığımız fonksiyon bloğu içerisinde genelde şart kontrolü için `if..else`, döngü yapıları içinse `for` döngüleri kullanılır. Bunların dışında alternatif olarak şart kontrolünde `switch..case`, döngü yapılarında `do..while` veya `foreach` kullanılabilir.

If..Else ile Şart Kontrolü

```
function yasKontrol()  
{  
  var yas=20;  
  if(yas>18)  
  {  
    alert("Ehliyet alabilirsiniz.");  
  }  
  else  
  {  
    alert("Ehliyet alamazsınız!");  
  }  
}
```

Yukarıda örnek olarak bir if..else bloğu verilmiştir. Bu örnekte olduğu gibi if..else yapılarında if() içerisinde şart kontrolü belirtilir,eğer başka bir şart kontrolü daha yapılacaksa else if() ile yeni şart kontrol ifadesi tanımlanır,en son ihtimal kaldığında ise sadece else bloğu içerisine bu şart altında yapılacak işlemlerin tanımlanması yeterlidir.

for ile Döngü Yapıları

```
function masallah()  
{  
  var mesaj="Tü tü tü...Maşallah.";  
  
  for(var i=0;i<41;i++)  
  {  
    document.write(mesaj);  
  }  
}
```

Yukarıda örnek bir for döngü bloğu tanımlanmıştır. Döngü yapıları sayfada aynı işlemin birden fazla kez tekrar edileceği durumlarda işleri kolaylaştırmak için kullanılan bir yöntemdir. Bu örnekte 41 kere document.write(mesaj); komutunu tekrar etmek yerine başlangıç değeri tanımlanmış,bitiş şartı bildirilmiş ve artış miktarı gösterilmiş bir for bloğu içinde tek bir satır yazdırma komutu vererek aynı işlem daha kısa yoldan yapılabilmiştir. Döngü içerisinde başlangıç,bitiş ve artış işlemlerini yapan bir sayaç tanımlanmalıdır. Bu sayaç genelde "i" adı verilen bir değişken içerisinde tutulur.

for(sayacın başlangıç değeri;sayacın bitiş/devam şartı;sayacın artış/azalış miktarı)

Özet

Web Tasarımı ve Programlama konusunda daha birçok farklı yapı ve teknoloji kullanılmakla birlikte web uygulamaları geliştirmede temel olabilecek 3 ana esası aşağıdaki gibi özetleyebiliriz.

Süreci bir binanın inşaat aşamalarına benzeterek anlatacak olursak:

- **HTML** ile oluşturduğumuz sayfa iskeleti bir binanın temel yapı malzemeleriyle(demir,kum,çimento,tuğla vs.) oluşan kaba inşaatı gibidir.
- **CSS** ise bu inşaatın görsel olarak düzenlenerek estetik ve kullanılabilirlik açısından dizayn edilmesi gibi sayfamızı güzelleştirir.
- **JavaScript** veya başka bir web programlama diliyle son olarak kullanıma hazır hale getirdiğimiz bu şık binayı daha yaşanabilir kılmak için gerekli olan elektrik,su,ısıtma,asansör tesisatı gibi işlevsellik katan öğeleri sayfamıza ekleyerek sayfamızı etkileşimli bir hale getirmiş oluruz.

Kaynakça ve Faydalı Linkler

İçerik ve örnekler için aşağıdaki kaynaklardan faydalanılmıştır.

- <http://www.w3schools.com>
- <http://www.css-tricks.com>
- <http://developer.mozilla.org>
- <https://www.quackit.com>

Daha fazla bilgi sahibi olmak için aşağıdaki linklerden hesaplarımı takip edebilirsiniz.

- <http://www.webmucidi.com>
- webmucidicom@gmail.com
- <http://www.twitter.com/webmucidi>
- <http://www.youtube.com/c/WebMucidi>

.