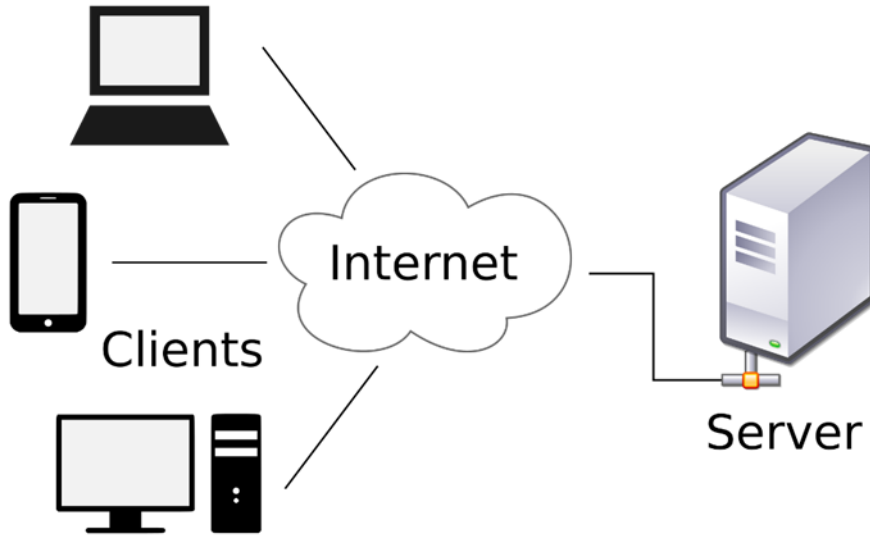


Web Teknolojileri ve Web Programlama (HTML, PHP, CSS, JavaScript)

Web Teknolojileri – Temel Kavramlar



İstemci: Sunuculara erişim isteğinde bulunan her türlü donanım veya yazılım.

İnternet: Başka bilgisayarlara veya donanımlara erişim sağlayabildiğimiz en büyük ağ yapısı.

Sunucu: Bilgilerin belirli bir alandan veya tüm dünyadan erişilebilmesini sağlamak için oluşturulan depolama ve yayınlama alanları.

Web Tarayıcıları (Web Browser): Web sitelerine erişmemizi ve gezinmemizi sağlayan araçlardır.

Bazıları: Chrome, Internet Explorer, Yandex Browser, Mozilla Firefox

Web Sayfası (Web Page): Web tarayıcıları ile erişip görüntüleyebildiğimiz her türlü dosyaya denir. Genel olarak HTML kullanılarak oluşturulmuş olur.

Alan adı (domain):

Web sitelerine erişmemiz için kullandığımız adreslerdir.

Örnek:

be.gazi.edu.tr

- be veya www→alt düzey domain alanı
- gazi→asıl domain alanı
- edu→üst düzey domain alanı, domainin amacını açıklamak amacıyla kullanılır

- tr→üst düzey domain alanı devamı veya ait olduğu ülkeyi ifade eden tanımlama

Üst düzey domain türlerinden bazıları

.com = Commerce – Ticari

.net = Network - Ağ

.gov = Government – Hükümet (Resmi)

.org = Organization – Organizasyon, Dernek

.edu = Education – Eğitim

.info = Information – Bilgi

Barındırma (Hosting):

Web sitelerinin/uygulamalarının web browserlar ile bir çok farklı konumdan erişilebilmesi için tüm istemcilerin bağlı olduğu bir bilgisayar/sunucu’da depolanması ve erişilebilmesi işlemidir.

Alan Adı Sunucusu (DNS: Domain Name Server):

Alan adları (domain) ile barındırma alanlarını (hosting) bağlayan sunuculardır. Bir danışma ofisi gibi düşünülebilir.

HTML (Hiper Metin İşaretleme Dili - Hyper Text Markup Language):

Web sayfalarının şeklini, yapısını, barındırdığı her nesnenin konumu ve özelliklerini belirlemek için kullanılan ve basit etiketlerle kontrol edilen işaretleme dilidir.

Etiket(Tag): <html>, <body>, <head> vb. yapılar etiket olarak tanımlanmaktadır.

URL (Tekdüze Kaynak Konumlandırıcı - Uniform Resource Locator):

İnternet üzerinde erişilen adreslerin genel adıdır.

Örnek:

www.gazi.edu.tr/klasör_adi/sayfa_adi.html

www: alt düzey domain alanı

gazi: asıl domain alanı

edu: üst düzey domain alanı, domainin amacını açıklamak amacıyla kullanılır

tr: üst düzey domain alanı devamı veya ait olduğu ülkeyi ifade eden tanımlama

HTTP (Hyper Text Transfer Protocol): Html sayfalarının web tarayıcısı tarafından erişilebilmesi için kullanılan protokoldür.

HTTPS: HTTP'nin güvenliği artırılmış halidir. Verilerin SSL ile şifrelenerek iletilmesinden kaynaklı daha güvenlidir.

Hyperlink, link, bağlantı (Gelişmiş bağlantılar):

Kullanıcının bir sayfadan, başka bir sayfa veya dosyaya yönlendirilmesi için kullanılan yönlendiricilerdir.

CSS (Cascading Style Sheets):

Html yapılarını esnek, görsel açıdan güzel görünen yapılara kavuşturmak için kullanılan stil dilidir. Html nesnelerinin nasıl görüneceğini detaylandıran bir dildir.

Web Betik Dilleri (Web Script Languages):

Kullanıcı ile web sitenin etkileşimli bir şekilde çalışması için kullanılabilen dillerdir. En bilinenlerden biri Javascript dilidir.

Web Programlama Dilleri

Web uygulamalarının programatik işlemlerinin yapılmasını sağlayan ve geliştirme sürecinde kullanılan dillerdir. Yaygın olarak PHP, Java (Jsf,Jsp vb.), Python (Django), Asp.Net kullanılmaktadır.

Veritabanları

Veritabanları yapılandırılmış bilgi veya verilerin depolandığı alanlardır. Veri tabanı yönetim sistemleri tarafından yönetilmektedir. Verinin depolanması ve işletilmesi amacı ile yönetim sistemleri ile efektif bilgi yönetimi sağlanmaktadır. Bunun yanında web programlama dilleri ile birlikte bilginin yönetimi web uygulamaları için kullanılmaktadır. Yaygın olarak MySQL, PostgreSQL, MongoDB, Oracle, MsSQL vb. kullanılmaktadır.

Tümleşik Geliştirme Ortamları (IDE)

IDE bilgisayar programcılarının hızlı ve rahat bir şekilde program geliştirebilmesini amaçlayan, geliştirme sürecini organize edebilen birçok araç ile birlikte geliştirme sürecinin verimli kullanılmasına katkıda bulunan araçların tamamını içerisinde barındıran bir yazılım türüdür. Bazı geliştirme ortamları JetBrains WebStorm, PyCharm, Visual Studio, Netbeans, Eclipse vb.

HTML

HTML Dosyası Nedir?

HTML, Hyper Text Markup Language'in kısaltmasıdır.

HTML çeşitli anlamlara gelen kodlamalar (tag) ve düz metinden oluşur.

Bu kodlar ancak bir Web Browser (Internet Explorer, Firefox) tarafından anlamlı hale getirilebilir.

Bu kodla yazılmış dosyaların uzantıları genellikle html ya da htm'dir.

HTML, en basit yolla bir not defteri ile yazılabilir.

Örnek kod:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Başlık</title>
  </head>
  <body>
    Site içeriği...
  </body>
</html>
```

Sayfada görünen:

Site içeriği...

DOCTYPE nedir?

DOCTYPE tanımlaması "<html>" etiketinin bir üst satırında olmalı ve DOCTYPE tanımlamasının üstünde hiçbir şey olmamalıdır. DOCTYPE tanımlaması bir HTML etiketi değildir bu yüzden "<html>" etiketi yerine kullanılamaz. İlgili DOCTYPE tanımlaması HTML5 versiyonu için kullanılan tanımlamadır. Önceki versiyonlar için ilgili tanımlama farklılık göstermektedir. Fakat HTML5 yaygın kullanıma sahip olduğu için diğer yapıları öğrenmenin bir faydası olmayacaktır.

Ana Etiketler

<html>...</html>	Bu kod browsera HTML belgesinin başladığı ve bitiği yeri bildirir.Bütün diğer kodlar bu iki kod arasına yazılır
<head>...</head>	HTML belgesinin ilk bölümüdür.Bu bölüme yazılanlar metin olarak ekranda gözükmez.
<title>...</title>	Head bölümü içine yazılan Title kodu sayfanın başlığının yazıldığı bölümdür.
<body>...</body>	HTML belgesinin bütün içeriği bu kodlar arasına yazılır.

Meta etiketleri

Meta etiketleri <head> etiketi içinde ilgili sayfanın tanımlama ve ayarlarının yapıldığı bölümdür. Bu aşamada yazan bilgilere göre, browser gösterimi ve arama motoru yönlendirmeleri gerçekleşmektedir. İlgili sayfanın yazım dilinden, içeriği oluşturan yazar bilgisine, anahtar kelimesinden, ilgili içeriğin ne zaman oluşturulduğuna arama motorlarının özenle otomatikleştirip önem verdiği bilgisellerin konumlandırıldığı yer.

Başlıklar

<h1>Başlık</h1>	Başlık
<h2>Başlık</h2>	Başlık
<h3>Başlık</h3>	Başlık
<h4>Başlık</h4>	Başlık
<h5>Başlık</h5>	Başlık
<h6>Başlık</h6>	Başlık

Metin Biçimlendirme Etiketleri

...	Yazıları koyu yapar
<i>...</i>	Yazılarınızı italik karakter yapar
<u>...</u>	Yazılarınıza alt çizgi koyar
<p>...</p>	Paragraf etiketi yapar. Paragrafınızın altına ve üstüne boşluk koyar.
 	Satırbaşı yapar.

Etiketler ve Parametreler

HTML’de bulunan etiketlerin özelliklerini ayarlamak için parametreler kullanılmaktadır. Parametreler başlangıç etiketinin içinde parametre=”değer” şeklinde tanımlanır. Eğer ilgili parametre yazılmıyorsa varsayılan olarak browserlar baz alacağı default değer işletilmektedir.

Bağlantı Etiketi

Bağlantılar ziyaretçinin bir tıklama ile, sitenizin içinde veya dışında, belirlediğiniz herhangi bir adrese gitmesini sağlar.Bağlantılar browser'a <a> etiketi ile bildirilir.

href	Bağlantının adresi bu parametreyle bildirilir
target	Sayfanın açılacağı yeri belirler.Target parametresi,açılacak sayfa yeni bir sayfa olarak açılacaksa “_blank” yazılmalıdır.

Resim Etiketi

Bir web sayfasında doğru ve yerinde kullanıldıklarında sayfaya estetik açılarından çok şeyler katarlar. Resimler etiketi aracılığıyla kullanılırlar.Aşağıda bu etiketin parametrelerini görüyorsunuz.

src	Resmin bulunduğu dizini bu parametre ile bildiririz
width	Resmin genişliğini piksel cinsinden bildirir
height	Aynı şekilde resmin yüksekliğini bildirir
border	Resmin etrafındaki çizginin kalınlığını belirtir.
align	Yatay konum belirler;left,right,center değerlerini alır.
title	Mouse resmin üzerindeyken yazacağınız açıklamayı gösterir

Örneğin:

HTML Tablolar

Tablolar, sayfaları satırlara/sütunlara bölmek ya da metin veya grafiklerin sayfada istediğimiz yerde durmasını sağlamak amaçlarıyla kullanabileceğimiz HTML'nin en önemli yapıtaşlarındandır. Sayfada gözüksün ya da gözükmesin tabloları bir iskelet gibi kullanabilir, böylece şu ana kadar öğrendiklerinizle yapamayacağınız gerçek düzenlemeyi yapabilirsiniz. Ana etiketi <table> etiketidir. Tablolar satır ve sütun sırasında oluşturulur. İlk olarak satır oluşturmak için <table> etiketi içinde <tr> etiketi kaç satır oluşturmak isteniyorsa o sayıda eklenir. Tüm etiketler bitiş etiketine sahip şekilde yazılır. Her satır içinde de ilgili satıra ait sütun etiketleri <td> olarak eklenir. <thead>, <tbody>, <th> gibi farklı kullanım etiketlerde ek olarak özelleştirme amacı ile kullanılan etiketlerdir. Tablolardaki en önemli parametreler rowspan ve colspan'dır. Bu parametreler td etiketleri üzerinde ilgili satır ve sütunları birleştirme amacı ile kullanılmaktadır.

Liste Etiketleri

HTML bize üç tip liste hazırlama imkanı veriyor.

Bunlar;

1. Sıralı listeler (ordered list) ...
2. Sırasız listeler (unordered list) ...
3. Tanımlama listeleri (definition list) <dl>...</dl>

Sıralı listeler rakam veya harf yada her ikisini içiçe kullanarak liste oluşturmamızı, sırasız listeler rakam/harf yerine madde imleri koyarak liste oluşturmamızı sağlar. Tanımlama listeleri ise bir listeden çok kalabalık metinlerde okumayı kolaylaştırmaya yardımcı olabilecek bir araçtır. Sıralı ve sırasız listelerde madde imlerini ... etiketi ile oluşturuyoruz. Tanımlama listelerinde ise tanımlanacak bilgi <dd>...</dd> tanım ise <dt>...</dt> etiketleri ile yapılmaktadır.

İç içe listeler oluşturma için ilgili yer imi bilgisi içinde türüne göre yeni liste oluşturulmaktadır.

Web sayfalarında statik menu, açılır menü gibi sıralı bir çok yapı listeler kullanılarak tasarlanmaktadır.

Canvas ve SVG Etiketleri

<canvas> elementi bir web sayfasında grafik çizmek için kullanılır.

<svg> etiketi: SVG Skala edilebilir Vektör Grafikleri (Scalable Vector Graphics) anlamına gelir. SVG vektör temelli grafikleri ifade etmek için kullanılır. SVG grafikleri XML formatında tanımlar. SVG grafikleri zoom'lansalar da yeniden boyutlandırılırsalar da kaliteleri düşmez. SVG dosyaları içindeki her element, her özellik (attribute) hareketlendirilebilir (anime)

Video ve Audio Etiketleri

Html de video gösterimi yapmak için <video>...</video> etiketi, ses dosyası dinletimi için <audio>...</audio> etiketi kullanılmaktadır. Parametrelerine ek olarak ilgili dosyaları etikete dahil etmek için <source> etiketi kullanılabilir.

Form Etiketleri

Formları site ziyaretçileri ya da kullanıcılarından bilgi alma amaçlı kullanırız. Onlardan düz metin, şifre ve seçim kutuları gibi nesneler oluşturarak bilgiler alırız. Bir form alanı <form> ile başlar ve </form> ile biter. Öncelikle bilgi amaçlı yerleştireceğimiz form nesnelerini tanıyalım.

Etiket	Açıklama
<form>	Form
<input>	Giriş alanı
<textarea>	Çok satırlı metin giriş alanı
<label>	Etiket
<fieldset>	Alanların ayarlarının nasıl olacağını belirtir.

<legend>	Manşet
<select>	Açılabilir liste (combobox)
<optgroup>	Seçenek kutusu grubu
<option>	Açılabilir liste içindeki maddeleri belirtir.
<button>	Buton
<datalist>	<input> etiketi ile birlikte bir liste oluşturma elemanı
<keygen>	<keygen> etiketi, bir form içerisinde bir anahtar çifti (key-pair) jeneratörü belirtir.
<output>	<output> elementi bir hesaplamanın sonucunu gösterir (bir script gibi)

Bir form alanı oluşturulduktan sonra içine ilgili form elemanları eklenir. <label> etiketi ile ilgili form elemanının ne olduğu bilgisi ile diğer form elemanları ve aksiyon elemanları (button veya input button türleri) ile form tamamlanmış olur.

En çok kullanılan form elemanı <input>'dur. <input> etiketi hem girdi hemde aksiyon elemanı olarak kullanılır.

<input type="text">	Metin
<input type="password">	Metin fakat görünüm şifreli
<input type="submit">	Form bilgilerini formun gönderileceği action bilgisine gönderim işi yapan button oluşturur
<input type="button">	Herhangi bir aksiyon atanabilecek button görevi görür
<input type="reset">	Tüm formu varsayılan değerlerine dönderen button işlevi görür
<input type="checkbox">	Seçim yapılacak form elemanıdır. Çoklu seçimlerde kullanılır.
<input type="radio">	Seçim yapılacak form elemanıdır. Tekil seçimlerde genellikle kullanılır.
<input type="hidden">	Gizli form elemanı olarak kullanılır. Veri tutmak amaçlıdır.
<input type="color">	Renk seçim paleti sağlar.
<input type="date">	Tarih alanı sağlar.
<input type="time">	Zaman alanı sağlar.
<input type="datetime-local">	Tarih zaman alanı sağlar.
<input type="email">	E-posta alanı sağlar.
<input type="file">	Dosya seçim alanı sağlar.
<input type="image">	Resim alanı sağlar.
<input type="week">	Hafta seçim alanı sağlar.
<input type="month">	Ay seçim alanı sağlar.
<input type="number">	Sayı seçim alanı sağlar.
<input type="range">	Aralıklı sayısal değer seçimi sağlar.
<input type="search">	Arama alanı oluşturmak için kullanılır.
<input type="tel">	Telefon bilgisi için kullanılır.
<input type="url">	URL bilgisi için kullanılır.

CSS

Sayfadaki nesnelerin kontrolü için daha fazla seçenek sağlamak amacıyla CSS kullanılmaktadır. CSS stil şablonu olarak da tanımlanabilir. Harici CSS dosyaları kullanılabildiği gibi sayfanın Head veya Body bölümüne de yazılıp kullanılabılır.

Html de bir bölümü 3 farklı şekilde ayrıştırabilir. Ve bu ayrıştırmayı CSS ile entegre kullanabiliriz. Bunlar id, class(sınıf) ve etikettir. Bir etikete özgü bazı CSS özellikleri atayabilirken bir etikete özel bir id bilgisine de özelleştirme atayabiliriz. En yaygını ise grupta yani sınıflamadır. Eğer bir özellik grubu birden fazla yerde aynı ise bu noktada sınıf kullanımı grupta için önemlidir.

Selektörler

Css'de ilk olarak önemli olan nereye etki edeceğini bilmektir. Bunun için CSS'de selektörler/seçiciler kullanılmaktadır. Bazı seçici türleri aşağıda sunulmaktadır:

- * (Evrensel Seçici)
- #X (ID "Tekil" Seçici)
- .X (Sınıf Seçici)
- X Y (Torun Seçici)
- X (Basit Seçici)
- X:link ve X:visited (Sözde Sınıf Seçicileri)
- X + Y (Bitişik Seçici)
- X > Y (Çocuk Seçici)
- X ~ Y
- X[Y] (Nitelik Seçici)

Bu seçiciler farklı varyasyonlarda etiket seçici, sınıf seçici ve id seçici varyasyonları ile kullanılabilmektedir.

CSS Kullanım Türleri

Bir diğer husus ise CSS kullanım türleridir. 3 tür kullanım mevcuttur. Bunlar; dahili/yerel, harici/genel ve satır içi kullanımlarıdır. Satır içi kullanım sadece ilgili noktasal alana etki etmek için kullanılmaktadır. Dahili kullanım sadece ilgili sayfada o stilin kullanılmasının planlandığı durumlarda kullanılmaktadır. Son olarak harici kullanım birden fazla noktada farklı stil gruplarının kullanılacağı durumlarda tercih edilmektedir.

Genel CSS

Web sayfasının Head bölümüne yazılırlar ve sayfanın tamamını etkilerler.

```
<html>
<head>
<title>Genel CSS Örneği</title>
<style type="text/css">
<!--
h1 {font-family:calibri; color:red; font-size:12}
-->
```

```
</style>
</head>
<body>
<h1>Genel CSS li Yazı</h1>
<h1>Genel CSS li Yazı</h1>
</body>
</html>
```

Yerel CSS

```
<html>
<head>
<title>Yerel CSS Örneği</title>
</head>
<body>
<h1>CSS siz Yazı</h1>
<h1 style=" font-family:calibri; color:red; font-size:12">Yerel CSS li Yazı</h1>
<h1>CSS siz Yazı</h1>
</body>
</html>
```

Harici CSS

Aynı anda çok sayıda sayfayı ilgilendiren stil güncellemeleri kolaylıkla yapılır. Ayrıca ayrı bir dosya olduğundan ilk çağrıldığında cache (ön bellek) ye yükleneceğinden daha sonraki çağrılmalarında çok hızlı çalışacaktır.

```
h1 { font-family:calibri; color:red; font-size:12}
h2 { font-family:arial; color:blue; font-size:15}
h3 { font-family:verdana; color:#FFFF00; font-size:18}
```

Yukarıdaki satırlar style.css olarak kaydedildiğinde, aşağıdaki şekilde çağrılabilir.

```
<html>
<head>
<title>Harici CSS</title>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
```

```
<h1>h1 Harici Stilli Yazı</h1>
<h2> h2 Harici Stilli Yazı</h2>
<h3> h3 Harici Stilli Yazı</h3>
</body>
</html>
```

```
<html>
<head>
<title>Bağlantılı CSS - Import</title>
<style type="text/css">
@import url(style.css);
h4 { font-family: Cambria; color: Black; font-size: 21 }
</style>
</head>
<body>
<h1> h1 Harici Stilli Yazı </h1>
<h2> h2 Harici Stilli Yazı </h2>
<h3> h3 Harici Stilli Yazı </h3>
<h4> h4 Genel Stilli Yazı </h4>
</body>
</html>
```

CSS – Stillendirme

Text Özellikleri

HTML’deki “font” tagının taşımadığı bazı özellikleri, sayfadaki “metin” lere uygulamak için kullanılan özelliklerdir.

text-decoration : Metnin altını ve üzerini çizme

underline : Metnin altının çizili olmasını sağlar

overline : Metnin üstünün çizili olmasının sağlar

line-through : Metnin ortasından çizili olmasını sağlar

none : Metnin herhangi bir yerine çizgi çekilmemesini sağlar.

text-align: Metni yatay hizalama

left: Metnin sola dayalı olmasını sağlar

right : Metnin sağa dayalı olmasını sağlar

center : Metnin ortalanmış olmasını sağlar

justify: Metnin iki yana yaslı olmasını sağlar

text-transform : Metni büyük veya küçük harflerle görüntüleme

lowercase : Metnin tümünün küçük harf olmasını sağlar

uppercase : Metnin tümünün büyük harf olmasını sağlar

capitalize : Metnin yalnızca kelimelerinin ilk harfinin büyük olmasını sağlar

line-height : Metnin satır yüksekliğini belirler. 3px, 5cm gibi değerler alır

text-indent : Metnin soldan ne kadar girinti ile içeriden başlayacağını belirler. 3px, 5cm gibi değerler alır

Yükseklik, genişlik, uzaklık, büyüklük gibi ölçüler;

px: pixel (nokta)

pt: punto (sadece yazıtiplerinde kullanılır)

pc: picas

cm: centimeter (santimetre)

mm: milimeter (milimetre)

in: inch (inç=2,54cm)

%: yüzde

cinsinden tanımlanabilir. 10px,2in,13pt, 200%, 5cm.. gibi.

cm, mm, in gibi ölçü birimleri ekran çözünürlüğüne göre değişkenlik arz edeceğinden kullanımları tercih edilmez. İçlerinde en çok kullanılan ölçü birimleri px, pt ve %'dir.

Yazı Tipi (font) Özellikleri

Font özelliklerini değiştirmek için kullanılır.

font-size: Yazı büyüklüğünü belirler

color: Yazının rengini belirler. Yelllow, blue gibi yazılışlar olabileceği gibi #FF00AE şeklinde de yazılabilir

font-family: Yazıtipini belirler (courier, verdana, arial, ... gibi). Virgül (,) konularak birden fazla belirtilebilir. Yazıtipinin adı boşluklu ise çift tırnak (") ile yazılır. "Times New Roman" gibi.

font-style: Yazının italik (yatık) olup olmasını belirler. Bu özellik yazılmadığında normal özellik alınır.

italic: Yazıyı eğik yapar

normal: Yazının eğik olmamasını sağlar

font-weight: Yazının bold (kalın) olup olmamasını belirler. Bu özellik yazılmadığında normal özellik alınır.

bold: Yazıyı koyu yazar

normal: Yazının koyu olmamasını sağlar

Bağlantı (Link) Özellikleri

HTML BODY tagında linklere verilen özelliklerin benzerleridir.

a:link : linklerin özellikleri

a:active : aktif linklerin özellikleri

a:hover : mouse link'in üzerine gelince alacağı özellik

a:visited : ziyaretçinin daha önceden ziyaret ettiği linklerin özellikleri

HTML in standart ayarlarını tamamen değiştirmek yerine, seçiciler vasıtasıyla sadece istenilen linklerin özellikleri de değiştirilebilir.

Liste Özellikleri

list-style-type: Liste elemanlarının başında görünecek madde imi, numara veya resmi belirlemeyi sağlar:

square: Kare

disc: Daire (içi dolu)

circle: Çember (içi boş)

decimal: Sayı (1,2,3...)

lower-roman: Küçük romen rakamları (i,ii,iii,...)

upper-roman: Büyük romen rakamları (I,II,III,...)

lower-alpha: Küçük harfler (a,b,c,...)

upper-alpha: Büyük harfler (A,B,C,...)

none: Madde imi ya da numara görünmez

list-style-position: Liste elemanlarının içeri girintili olup olmayacağını ayarlamayı sağlar:

inside: Üstteki listenin içerisinden başlatır

outside: Üstteki liste ile aynı hizada başlatır

list-style-image: Madde imi olarak resim görüntülemeyi sağlar.

Arkaplan Özellikleri

Body, Table, Td, P gibi HTML nesnelerinin arkaplan özellikleri düzenlenebilir.

background-color: Arkaplan rengini, renk adını ya da rengin hexadecimal kodunu yazarak değiştirebiliriz. HTML'deki BGCOLOR gibi düşünebilirsiniz.

background-image: Arkaplan resmi, url(resim_adı.gif) şeklinde belirlenebilir. HTML'deki BACKGROUND gibi düşünebilirsiniz.

background-repeat: Resmin arkaplanda tekrarlanma türünü belirlememizi sağlar.

repeat: Resmi X ve Y eksenini boyunca çoğaltır

repeat-x: Resmi sadece X eksenini boyunca çoğaltır

repeat-y: Resmi sadece Y eksenini boyunca çoğaltır

no-repeat: Resmin çoğaltılmayacağını, tek halde görüntüleneceğini belirtir

background-position: Resmin konumunu belirlememizi sağlar.

Kenarlık (Border) Özellikleri

Border özelliğini kullanan tüm HTML taglarının kenarlık rengi, kenarlığın kalınlığı gibi özellikleri düzenlenebilir.

border-width: Kenarlığın kalınlığını belirler (thin, thick, 2px, 1mm, .. gibi).

border-color: Kenarlığın rengini belirler (blue, #0000FF, .. gibi).

border-style: Kenarlığın görünüm stilini belirler (double, groove, dashed, solid, none gibi).

border: Yukarıdaki 3 özelliği bir arada kullanabilmemizi sağlar (border: 1px dashed yellow gibi).

border-left: Sadece sol kenarlığı düzenlemek için kullanılır

border-right: Sadece sağ kenarlığı düzenlemek için kullanılır

border-top: Sadece üst kenarlığı düzenlemek için kullanılır

border-bottom: Sadece alt kenarlığı düzenlemek için kullanılır

Kenar Payı (Margin) ve Aralık (Padding) Özellikleri

Nesnenin sınır çizgisi ile hangi yönden ve ne kadar boşluk bırakılacağını belirtmek için margin özelliğini, sınırın iç tarafında bırakılacak boşluğu belirtmek için ise padding özelliğini kullanılır. Bu özellikler, nesneleri konumlandırılırken kullanışlıdır. Sadece margin belirtilerek nesnenin 4 yönünü de etkiler iken, margin-top, margin-bottom, margin-left, margin-right gibi kullanımlar da mevcuttur. Padding için de aynı yapı geçerlidir.

İmleç (cursor) Özellikleri

Farenin imleci bir nesne üzerine getirildiğindeki özellikler düzenlenebilir.

<html>

<body>

```
<head>
<title>Cursor Özellikleri</title>
<body bgcolor=yellow>
  <p><font face="verdana" size="2"><b><u>Sonucu görmek için, imleci yazıların üzerine götürünüz.
</u></b></font>
</p>
  <font face="verdana" size=1 color="#0000AA" size="2">
<b><span style="cursor: crosshair">crosshair</span></b><p>
<b><span style="cursor: default">default</span></b><p>
<b><span style="cursor: help">help</span></b><p>
<b><span style="cursor: move">move</span></b><p>
<b><span style="cursor: pointer">pointer</span></b><p>
<b><span style="cursor: e-resize">e-resize</span></b><p>
<b><span style="cursor: n-resize">n-resize</span></b><p>
<b><span style="cursor: ne-resize">ne-resize</span></b><p>
<b><span style="cursor: nw-resize">nw-resize</span></b><p>
<b><span style="cursor: s-resize">s-resize</span></b><p>
<b><span style="cursor: se-resize">se-resize</span></b><p>
<b><span style="cursor: sw-resize">sw-resize</span></b><p>
<b><span style="cursor: w-resize">w-resize</span></b><p>
<b><span style="cursor: text">text</span></b><p>
<b><span style="cursor: wait">wait</span></b><br>
</font>
</body>
</html>
```

Katman (CSS-P) Özellikleri

Div tagı içinde, CSS-P özellikleri ile nesnelerin pozisyonları pixel pixel düzenlenebilir.

width: Katmanın genişliği

height: Katmanın yüksekliği

position: Katmanın nasıl konumlanacağını belirler.

absolute: Katmanın konumunu kesin olarak bildireceğimiz zaman kullanırız.

relative: Katmanın konumunu diğer öğelere göre değişebilen şekilde vermemizi sağlar.

top: Katmanın pencerenin üst kenarından uzaklığı

left: Katmanın pencerenin sol kenarından uzaklığı

overflow: Katmanın belirtilen yükseklik ve genişliğe sığmayan kısmına ne olacağını belirler. İki değer alır. scroll, kaydırma çubukları ekler, auto ise kaydırma çubuklarını otomatik olarak belirler.

visibility: Katmanın görünür veya gizli kalacağını bildirir. Alacağı değerler Explorer'da görünür yapmak için visible ve gizli yapmak için hidden; Netscape'de ise görünür yapmak için show ve gizli yapmak için hide'dır.

z-index: Katmanın sayfa üzerindeki önceliğini bildiren sıra sayısını belirtir. Üst üste yerleştirilmiş katmanların numarası diye düşünülebilir.

JavaScript

JavaScript popüler bir programlama dilidir. HTML dili ile kullanılır. İnternette, bilgisayarlarda, tabletlerde ve telefonlarda çalışır. Günümüzde JavaScript desteklemeyen bir tarayıcı kalmamıştır. HTML DOM, W3C uyumlu bir HTML elementleri erişim sistemidir. JavaScript HTML ile uyumlu olarak DOM'u değiştirir.

Örneğin aşağıdaki örnek id="demo" olan elementin içeriğini (innerHTML) değiştirmeye yarıyor:

```
document.getElementById("demo").innerHTML = "Merhaba JavaScript";
```

Yukarıdaki document.getElementById(), JavaScript'te kullanılan yüzlerce yöntemden biridir. JavaScript kullanarak şunları yapabilirsiniz;

- HTML nesnelerini değiştirmek
- HTML nesnelerini silmek
- Yeni HTML nesneleri eklemek
- HTML nesnelerini kopyalamak
- Ve daha fazlası...

JavaScript Kod Yapısı

JavaScript kodları HTML belgesi içerisinde kullanılır. Kod yazımına başlamadan önce **<script>** nesnesi eklenir ve kod yazma işlemi bittikten sonra **</script>** yazılarak element kapatılır.

```
<script> alert('Hoş geldiniz!'); </script>
```

Bir JavaScript kodu için oluşturacağınız **<script></script>** nesnesini **<head>** bölümünde ya da **<body>** bölümünde kullanabilirsiniz.

NOT: BODY elementi içerisinde kullanmak sayfa yüklenme hızını arttıracaktır.

İsterseniz bir javascript dosyası oluşturarak ve bunu HTML belgenizden çağırarak kullanabilirsiniz. Bu kullanım şekli kaynak kodlarınızın sade olmasını sağlayacaktır.

Bunun için **<script>** kodunuza **src=""** özelliği ekleyip dosya adını ana dizine göre belirtin. Örneğin script dosyamızın adı "script.js" ise;

```
<script src="script.js"></script>
```

JavaScript'i Test Etmek

JavaScript'in yazdırma ya da çıktı verme gibi bir özelliği yoktur. JavaScript, sadece HTML Belgesinde yer alan nesneleri değiştirmek için kullanılır.

getElementById() Fonksiyonu: HTML nesnelere erişmek için JavaScript'in **getElementById(id)** kodunu kullanırız.

Yukarıdaki kodda parantez içinde yer alan **id**, HTML nesnesini tanımlayan bir id özelliğidir. Aşağıdaki örnekte innerHTML ise tanımlanan nesnenin içeriğidir. Aşağıdaki kodu yazarak bir HTML nesnesinin içeriğini değiştirebildik:

```
<!DOCTYPE html>
<html>
<body>

<h1>Web Sayfam</h1>

<p id="para">İlk Paragraf</p>

<script>
    document.getElementById("para").innerHTML = "Paragraf değişti.";
</script>

</body>
</html>
```

Yukarıdaki JavaScript kodu (<script> içindeki) web tarayıcısı tarafından çalıştırılır.

document.getElementById("para") bir javascript kodudur ve id="para" olan HTML nesnesini seçmemize yarar.

innerHTML = "Paragraf değişti"; kodu ise HTML içeriğini değiştirmeye yarar.

Sadece test amaçlı olarak HTML belgesine yazı yazdırmak için **document.write()** kodunu kullanabilirsiniz.

```
<!DOCTYPE html>
<html>
<body>
<head>
  <h1>Web Sayfam</h1>
</head>
<body>
  <p>İlk paragraf</p>

  <script>
    document.write(Date());
  </script>
</body>
</html>
```

NOT: document.write sadece test amaçlıdır. Yüklemesi tamamlanmış bir HTML belgesinde kullanırsanız tüm elementler sıfırlanır.

Eğer tarayıcınız hata ayıklama özelliğine sahipse, console.log() kodunu kullanarak JavaScript değerlerini tarayıcınızın konsol bölümünde görebilirsiniz.

Konsol bölümüne erişmek ve hata ayıklamayı başlatmak için tarayıcınızda F12'ye basmalısınız.

Değişkenler

JavaScript'te değişkenler verilerin saklandığı kelimelerdir. Bu kelimeler "var" ifadesi kullanılarak tanımlanır. eni bir değişken tanımlarken "**var**" komutundan faydalanırız. Örneğin sayısal bir ifade olarak 5 (beş) verisi içeren x adında bir değişken oluşturmak için;

```
var x = 5;
```

Yukarıda tanımladığımız x bir değişkendir ve değeri 5'dir.

Tek bir "var" kullanarak birden fazla değişken oluşturabiliriz, bunun için tanımlanan değişkenler arasında virgül (,) konulur:

```
var x = 1, y = 2, c = 3;
```

Değişkenleri İsimlendirirken Dikkat Edilecekler:

- Değişken adları mutlaka A-Z ya da a-z aralığında bir harf ile başlar, sayı ile başlayamaz.
- Değişkenler bazı durumlarda \$ ya da _ ile de başlayabilir.
- Değişken adları büyük - küçük harf duyarlıdır. Yani var x = 5 ile var X = 5 farklı değişkenlerdir.

Değişkenler birçok veri türü saklayabilmektedir. Bunların başlıcaları; metin, sayısal, mantıksal, dize ve objelerdir.

Yazı Verisi (String): String veriler söz dizelerini depolar. Bu tarz verilerin değerlerini girerken tırnak (") ya da kesme işareti (') kullanılır.

Sayısal Veri (Numbers): Sayısal bir değeri olan değişkenler oluştururken sadece sayılardan faydalanırız. Ayrıca nokta (.) kullanarak ondalıklı sayı girişi yapabiliriz.

Mantıksal Veri (Boolean)

Bazı değişkenler sadece iki veri içerebilir. İfade ya doğrudur (true) ya da yanlış (false).

Mantıksal veriler genellikle koşul gerektiren ifadeler için kullanılır.

Dize Verisi (Array)

Dize, bir veri türünde birden fazla veri içeren değişken topluluğudur. Dize oluştururken köşeli parantez kullanır ve değerleri virgül ile ayırırız.

```
var arabalar = [ "Toros", "Broadway", "Megan" ]; // Yazı dizesi
var fiyatlar = [ 5000, 10000, 15000 ]; // Sayısal dize
```

Nesneler - Veri İkili (Object)

Nesneler, alt değişkenler ve bunların değerlerini barındıran veri türüdür. Bu alt değişkenler sıralanırken kıvrım parantezler kullanılır; isim:değer şeklinde ve virgüllerle ayrılarak yazılır.

Boş ve Tanımsız Veriler (Undefined & Null)

Bazı değişkenlerin herhangi bir değeri yoktur, veri içermezler. Bir veri içermeyen değişken oluştururken **var degisken;** şeklinde yazılır. Yani eşittir (=) operatörü ile bir veri girişi yapılmaz. Bazen bir değişkenin sahip olduğu veriyi silmek gerekebilir. Bunun içinse **null** kullanılır.

```
var isim; // isim değişkeni veri içermez.
kullanici = null; // kullanici değişkeninin verisi silinir.
```

Değişkenin Veri Türünü Değiştirmek

JavaScript'in değişkenleri dinamiktir. Yani veri türleri yeni değerler atanarak kolaylıkla değiştirilebilir. Aşağıdaki örnekte x değişkeni sırasıyla tanımlanmış sonra x değişkenine sayısal veri girilmiş ve metin verisi girilmiştir;

```
var x; // x'in veri türü tanımsız
var x = 100; // x, sayısal bir ifade oldu
var x = "Murat"; // şimdi x, metin ifade ediyor
```

JavaScript Koşul İfadeleri

Farklı durumlarda farklı işlemler yapmak gerekebilir. Bu durumların tespit edilmesi için koşul ifadeleri bize yardımcı olur. Şartların sağlanıp sağlanmadığını tespit ederken standart üç ifade kullanılır. Bunlar if, else if ve else'dir.

- if (koşul sağlanıyorsa) { şunları yap }

- else if (koşul sağlanıyorsa) { şunları yap }
- else { şunları yap }

Koşul oluştururken mutlaka IF ile başlanır. Parantez içinde koşul belirtilir ve yapılacaklar kıvrımlı parantez içinde yazılır.

ELSE IF kullanmak için öncelikle IF kullanmış olmak gereklidir. Yani IF kullanmadan ELSE IF ile bir kontrol yapılamaz. "O olmadı şimdi bunu dene" anlamı taşır.

ELSE kullanırken parantez açılmaz ve koşul yazılmaz. ELSE, diğer tüm if ve else ifler çalışmadıysa, yani hiçbir koşul sağlanılamadıysa yapılacak şeyi belirtir.

Koşul Operatörleri

Koşul ifadeleri mantıksal değerlerdir. Doğru (true) ya da yanlış (false) olabilirler. İki eşittir (==) bir koşullu ifadedir ve iki değişken ya da değerin denliğini doğrulamayı sağlar.

Karşılaştırmalar

İşaret	Anlamı	Koşul İfadesi	Sonuç
==	Birbirine eşitse	x == 8	false
		x == 5	true
!=	Eşit değilse	x != 8	true
===	Türleri eşitse	x === "5"	false
		x === 5	true
!==	Türleri eşit değilse	x !== "5"	true
>	Büyükse	x > 8	false
<	Küçükse	x < 8	true
>=	Büyük veya eşitse	x >= 8	false

İşaret	Anlamı	Koşul İfadesi	Sonuç
<=	Küçük veya eşitse	x <= 8	true

Mantıksal İfadeler

İşaret	Anlamı	Koşul İfadesi	Sonuç
&&	Ve	x < 10 && y > 1	true
	Veya	x == 5 y == 5	false
!	Değilse	!(x == y)	true

Mantıksal ifadelerden **ve (&&)** bağlacında her iki durum da doğru ise koşul sağlanır. Eğer **veya (||)** bağlacı kullanılmışsa belirtilen koşullardan sadece biri doğru olsa da önerme doğrudur.

Switch Case

JavaScript'teki Switch kodu bir ifadeden çıkan sonuca göre işlem yapmamızı sağlar.

Bu kodun temel yapısı aşağıdaki şekildedir;

```
var x = 1;
var mesaj = "";

switch( x )
{
  case 2:
    mesaj = "Sayı: 2";
    break;
  case 1:
    mesaj = "Sayı: 1";
    break;
  default:
    mesaj = "Sayım 1 ya da 2 değil!";
}
```

switch(ifade) kodunda ifadede elde edilen sonuç **case deger:** ile eşit ise, o kodlar çalıştırılır. **break;** kodların sonuna eklenir ve switch'ten çık anlamına gelir.

Eğer elde edilen sonuç case ile belirtilenlerin hiçbirine uymuyorsa **default:** kullanılan kısmın çalıştırılmasını sağlayabiliriz.

JavaScript Döngüleri

Döngüler aynı kodu farklı değişkenlerle döndürmemizi sağlar. FOR ve WHILE, JavaScript'te sıklıkla kullanılan iki döngü komutudur.

5 elemanlı meyveler adında bir dizemiz olduğunda, döngü kullanmadan bunları yazdırmak için uzunca aşağıdaki kodu yazmamız gerekirdi;

```
kelime += meyveler[0] + "<br/>";  
kelime += meyveler[1] + "<br/>";  
kelime += meyveler[2] + "<br/>";  
kelime += meyveler[3] + "<br/>";  
kelime += meyveler[4] + "<br/>";
```

Fakat FOR döngüsü kullansaydık işte bu kadar kısa olacaktı:

```
for( i = 0; i < meyveler.length; i++ )  
{  
    kelime += meyveler[i] + "<br/>";  
}
```

FOR Döngüsü: Kullanırken noktalı virgülle ayrılmış üç bilgi girmemiz gerekir;

for (ifade 1; ifade 2; ifade 3)

- ifade 1: Döngü öncesi uygulanacak kodlar
- ifade 2: Döngünün hangi şartlarda devam edeceği
- ifade 3: Döngü içindeki kodların uygulanması her tamamlandığında çalıştırılacak kodlar

Örneğin şu ifadeyi yorumlayalım:

```
for (i = 0; i < 5; i++ )
```

Burada **i = 0**, i adlı bir değişken yaratıp 0 değerini verdiğimizizi belirtiyor. **i < 5**, i, 5'ten küçükse döngü devam eder. Ne zaman ki eşit ya da büyük olursa döngü sona erer. Sondaki **i++** ise döngü içindeki kodlar her çalıştırdıktan sonra i değişkeninin değerini 1 arttırmasını ister. Sonuçta döngü i = 0 ile başlayıp her seferde artarak i = 4 olduğunda son kez çalıştırılıp bitecektir. Toplamda 5 kez kodlar çalıştırılır.

FOR/IN Döngüsü: Nesnelerin barındırdığı verileri yazdırmayı sağlar.

WHILE Döngüsü: **while(koşul) { kodlar }** şeklinde kullanımı olan bu döngü koşul sağlandığı sürece tekrar edecektir. Örneğe bakalım;

```
var mesaj = "";

var i = 0;

while( i < 10 )
{
    mesaj = "Benim sayım " + i + "<br />";

    i++;
}
```

DO/WHILE Döngüsü: WHILE koduna benzer şekilde çalışır. Fakat bu kez koşulun kontrol edildiği yer döngünün başında değil sonundadır.

```
do {
    mesaj += "Benim sayım " + i + "<br />";
    i++;
}
while (i < 10);
```

Döngüyü Sonlandırmak (break): JavaScript'te çalışmakta olan bir döngü **break** komutuyla durdurulabilir. Aşağıdaki örnekte 10'dan küçükse çalıştır anlamı içerse de i = 3 olduğunda döngü sonlandırılacaktır. Çünkü break kodu kullanıldı.

```
for (i = 0; i < 10; i++)
{
    if (i == 3)
    {
        break;
    }

    text += "Benim sayım " + i + "<br />";
}
```

Sonraki Tekrara Geçmek (continue): Döngü tekrarlardan oluşur. Bazı tekrarları hızlı geçmek ve sonraki tekrardan devam etmek isteyebilirsiniz. Bu durum için kullanacağınız kod **continue**.


```
var iletı = "";

for (i = 0; i <= 10; i++)
{
    if (i == 3)
        continue;

    iletı += "Benim sayım " + i + "<br />";
}
```

Fonksiyonlar

Fonksiyonlar, çağırıldıklarında belli işlemleri yapan kod gruplarıdır. Bunlar script içerisinde yer alır ancak kullanılmadıkları sürece çalıştırılmaz.

```
function fonksiyonAdı ( varsa parametreler )
{
    çalıştırılacak kodlar
}
```

- **fonksiyonAdı** - Fonksiyonuna bir isim verin. Çalıştırırken bunu kullanacaksınız.
- **parametre** - Fonksiyonun çalışırken ihtiyaç duyacağı ya da işleyeceği değişkenlerdir. Hiç kullanmayabilirsiniz. Birden fazla ise aralarına virgül koymalısınız.

Bir fonksiyon;

- Bir olay gerçekleştiğinde (örneğin butona basılması vs.)
- JavaScript kodları yardımıyla çağırılsa
- Kendiliğinden (kendi içerisinde çağırır)

durumlarında çalışır. Fonksiyonu çalıştırırken fonksiyon adı yazılır, parantez açılıp parametre varsa yazılır ve parantez kapatılır. Parametreler, fonksiyonda belirtildiği sırada olmalıdır.

```
// Fonksiyonu oluştururken;

function fonksiyonum(parametre1, parametre2)
{
    // kodlar buraya gelecek
}

// Fonksiyonu çağırırken;

fonksiyonum(parametre1, parametre2);
```

Fonksiyonlar sadece kod parçalarını çalıştırmaz. Ayrıca çıkardıkları sonuçları da iletirler. Bunun için **return** kodu kullanılır. Örneğin bir fonksiyon yazalım, kendisine verilen iki sayıyı çarpıp sonucu yine bize iletсин.

```
var x = Carp(4, 3);

function Carp(a, b)
{
    return a * b;
}
```

PHP

PHP: Hypertext PreProcessor kısaltmasıdır. Sunucu tabanlı olarak çalışır. Yani kodları derleyen, çalıştıran sunucudur. Birçok Veritabanına uyumludur: MySQL, Oracle, ODBC vb. Ücretsiz ve açık kaynak kodlu bir yazılımdır. Bir PHP kodu, uzantısı *.php olan bir ASCII (düz metin) belgesinde yer alır. Sunucu belirttiğimiz yerlerde ya da tamamında ilgili php kodlarını derler ve bu şekilde kullanıcıya iletir. PHP en çok MySQL veritabanı ile birlikte çalıştırılır. MySQL'de tıpkı PHP gibi açık kaynak kodludur, gelişmeye imkan tanır. Küçük ve büyük sitelerde tercih edilir. PHP'yi seçmemizin temel nedeni, Windows, Linux gibi birçok platformda çalışması, tüm sunuculara uyumlu olması, ücretsiz olması ve kolay öğrenilmesidir.

PHP Kod Yapısı

PHP Scriptleri <?php ile başlar ve ?> ile biter. PHP Kodlarımız bu aralığa yazılır. Ayrıca çoğu sunucu <? ile başlamasına da destek verir. Buna karşın <?php olarak başlatmaya kendinizi alıştırmamız faydanıza olur.

```
<html>
<body>
<?php
    echo "Merhaba dünya!";
?>
</body>
</html>
```

Yukarıdaki örnekte görüldüğü üzere her php komut satırı noktalı virgül (;) ile biter. PHP'de yazı yazdırmanın iki basit kodu vardır: **echo** ve **print**, yukarıda echo ile yazı yazdırdık. PHP Kodlarımız arasına satır ya da uzun bölümler halinde açıklama eklememiz mümkün. Bunlar php tarafından görülmez ve bize kolaylık sağlar:

```
<html>
<body>
<?php
    // Bu bir açıklama

    /* Bu da
       bir
       açıklama */
?>
</body>
</html>
```

Değişkenler

PHP'de kullanacağımız sayısal değer, kelime, dize gibi değişkenler dolar işareti (\$) ile başlar. A-Z, a-z, 0-9 ve _ karakterlerinden oluşabilir.

PHP'de bir değişken önceden tanımlanmadan kullanılır. Örneğin bir değişken belirlediğimizde, o bu değeri ilk kez alıyordur. Buna bağlı türü, yapısı vb değişkenlik gösterebilir.

Bu durumun istisnası DEFINE(tanım, değer) komutudur. Böylelikle \$ işareti ile başlamayan değişmez bir değer oluşturmuş oluruz:

```
<?php
    define('PI', 3.14);

    echo PI; // 3.14 yazdırır.
?>
```

PHP'de alfasayısal yani içerisinde harf, sayı vb. barındıran değişkenlerin kullanımından ve bunlara özel işaret ve fonksiyonlardan bahsedelim.

1. Bir alfasayısal değeri belirtirken tırnak işareti kullanabildiğimiz gibi (') üst ayraç da kullanabiliriz.
- 2.. Tırnak işareti kullandığımız string değerlerde değişkenleri bunların içerisinde kullanabilmekteyiz.
- 3.. Alfasayısal değeri belirtirken tırnak ve üst ayraç kullanımına dikkat etmek gerekir.

Sadece string değişkenlerde geçerli olarak değişkenler arası nokta işareti kullanarak kelimeler birleştirilebilir.

strlen() Fonksiyonu: Değişkenin karakter uzunluğunu (harf sayısı) verir.

strpos() Fonksiyonu: Aranılan bir kelime ya da harfin değişkenin 1. harfine ne kadar uzaklıkta olduğunu gösterir (1. harf 0 sayılır).

PHP Operatörleri (İşleçleri)

Aritmetik İşlemler

Operator	Açıklama
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	Mod (bölümünden kalan)
++	Arttırma (1 ekleme)
--	Azaltma (1 eksiltme)

Görev İşlemleri

Operator	Örnek	Açıklama
=	x = y	Eşitlik
+=	x += y	x, y ile toplamına eşit
-=	x -= y	x, y den farkına eşit
*=	x *= y	x, y ile çarpımıdır
/=	x /= y	x, y ile bölümüdür

Karşılaştırma İşlemleri

Operator	Açıklama	Örnek
==	Eşittir	5 == 8 YANLIŞTIR

Operator	Açıklama	Örnek
!=	Eşit değildir	5 != 8 DOĞRUDUR
<>	Eşit değildir	5 <> 8 DOĞRUDUR
>	Büyüktür	5 > 8 YANLIŞTIR
<	Küçüktür	5 < 8 DOĞRUDUR
>=	Büyük eşittir	5 >= 8 YANLIŞTIR
<=	Küçük eşittir	5 <= 8 DOĞRUDUR

Mantıksal İşlemler

Operator	Açıklama	Örnek
&&	VE	x = 6, y = 3 ise (x < 10 && y > 1) DOĞRUDUR
	VEYA	x = 6, y = 3 ise (x == 5 y == 5) YANLIŞTIR
!	DEĞİLSE	x = 6, y = 3 ise (!(x == y)) DOĞRUDUR

PHP If Else Deyimi

Koşullara Dayalı İşlemleri Gerçekleştirme

Kod yazarken sıklıkla kullanılan koşul belirteçleri, duruma bağlı olarak işlemler yapmamıza olanak sağlar. Genel yapıları 3 şekildedir:

1. if (koşul) - koşul gerçekleşirse komut uygulanır.
2. if (koşul) else ... - koşul gerçekleşmez ise ELSE'de belirtilenler uygulanır.
3. if (koşul) else if (koşul) else ... - İlk koşul gerçekleşmezse alternatif koşullar else if ile belirtilebilir.

PHP Switch Kodu

Bir değişkene bağlı koşullu işlem yaparken IF, ELSEIF, ELSEIF ... şeklinde uzunca bir komut kullanmak hem zamanımızı alır, hem php derleyiciyi yorar. Bu gibi durumlarda Switch kodu işimizi kolaylaştırır.

PHP Array (Dize)

Dize, birden fazla değişken barındıran veri grubudur. Bu değişkenler aynı türdedir ve tek bir ana değişken içerisinde bulunurlar.

Dize İçinde Dize (Çoklu Dizeler): Bir arraylist içerisinde dizeler kullanabiliriz. Her bir dize için [] eklenir.

PHP While Döngüsü

Birden fazla işlemi aynı anda yapmak, dizelerle çalışmak için döngülerden faydalanırız. Bunlardan biri de While() döngüsüdür.

While(koşul) { ... } şeklinde kullanımı olan bu komutta koşul doğru (true) olana kadar iç kısımdaki kod tekrarlanmaktadır.

Do ... while Döngüsü: Eğer kodu önce yaptırtacak sonra koşulu kontrol edeceksek bu komut bize yardımcı olur.

PHP For Döngüsü

Bir kodu, belirttiğimiz şekillerde ve sayılarda yaptırtmak için for komutu bize yardım eder.

for(değişken, koşul, artış miktarı) { ... } şeklinde kullanılır.

Foreach Döngüsü: Bir dize (array) içerisindeki tüm değerleri listelemek, onlar üzerinde işlem yapmak için bu komuttan yararlanabiliriz.

PHP Fonksiyonları

PHP'de hazırlayacağımız özel fonksiyonlarla işlerimizi kolaylaştırabilir, türlü işlemler için zaman kazanabiliriz.

function isim(değişkenler) { ... } şeklinde kullanılır. Fonksiyonlar bazı değişkenlerle birlikte kullanılabilir.