

boyutlarını orjinal halde getirmektedir. Değişiklik yapmak için width ve height özellikleri kullanılır.

Mesela `` yazılırsa 200x180 pixel ayarlanmış olur.

34) alt özelliği ile herhangi bir hatadan dolayı açılmayan bir resim için kullanıcıya mesaj verir.

`` şeklinde kullanılabilir.

35) title özelliğini linklerde göstermiştik. Mouse imleci getirildiğinde açıklama yazıyordu.

`` şeklinde kullanılabilir.

36) Resimler yatay olarak konumlandırılabilir. align komutu için sağ veya sola kaydırmak için right ve left kullanılır.

`` şeklinde yazılırsa resim sayfanın sağında gözüktür.

Resim ortalanmak istenirse align kullanılmaz. `<center></center>` etiketi ile yapılabilir.

37) Resimler dikey olarakda konumlandırılabilir. Bunun için yine align kullanılır. Align özelliği ile resmin yanında bulunan yazının resme göre dikey konumunu belirtebilirsiniz. Dikey konumlama için üç değer vardır. Bunlar top, middle ve bottom'dur. Eğer herhangi bir özellik kullanılmamışsa bottom olarak kabul edilir.

a)top = resmin yanında bir yazı varsa resmin en alt kısmına göre hizalansın.

b)middle = resmin yanında bir yazı varsa resmin ortasına göre hizalansın.

c)bottom = resmin yanında bir yazı varsa resmin en alt kısmına göre hizalansın.

38) Resimlere border özelliği ekleyebilir. Border özelliği resmin etrafına bir çerçeve çizer. border="5" ifadesi çerçevenin kalınlığı

5 olacaktır. `` şeklindedir.

39) Resimlerde hspace ve vspace özellikleride kullanabiliriz. Bunlar resim ve metin arası boşluğu sağlar. Resminizin yanında

yazı varsa daha belirgin halde kullanılması için hspace,vspace kullanabiliriz.

`` şeklindedir.


```
<tr>
<td>2.Satır , 1.Sütun</td>
<td>2.Satır , 1.Sütun</td>
</tr>
<tr>
<td>3.Satır , 1.Sütun</td>
<td>3.Satır , 2.Sütun</td>
</tr>
</table>
```

52) <th></th> etiketleri sütun başlığı belirtmek için kullanılır. (<th> belirtirken, <tr> veya <td> belirtmek gerekmez.)

Örnek:

```
<table border=2>
<th>Continuum</th>
<th>Company</th>
<tr>
<td>Doruk</td>
<td>Uğur</td>
</tr>
</table>
```

53) <thead></thead> etiketi ile tablomuz için başlık oluşturabiliriz.(<table></table> etiketi içinde kullanılır.)

54) colspan özelliği ile sütunları birleştirebiliriz. Örnek(<td colspan=2>Burası 2 kolondan oluşur</td>) şeklinde.

55) rowspan özelliği ile satırları birleştirebiliriz. Örnek(<td rowspan=2>Burası 2 satırdan oluşur</td>) şeklinde.

56) width, height özellikleride kullanırız. Tablo genişlik ve yüksekliği ayarlayabilmek için sayısal veya yüzdeler olarak ayarlama yaparız. Yüzdeler ayarlamalarda sayfanın oranına göre yüzdeleyebiliriz. Örnek(<table border="1"width="50%"height="50%">) şeklinde.

57) width, height özelliği kullanarak hücre genişlik ve yüksekliği de ayarlayabilirsiniz. Örnek(<td width="10"height="20">Doruk UĞUR</td>)

58) bgcolor özelliği ile tablonun arkaplan rengini belirleyebiliriz. Örnek(<table border="1"bgcolor="yellow">)

59) bgcolor ile her hücreye farklı renkte verilebilir. Örnek(<td

70) `<frame>`/`</frame>` etiketi ile gösterilecek olan sayfalar belirtilir. `<frameset>` etiketi içinde kullanılır ve her bir sayfa için bir tane `<frame>` etiketi kullanılır. `<frame>` etiketinin temel kullanım şekli şöyledir. `<frame name="frame1" src="sayfa1.html">`

Burada `name="frame1"` bölümünde frame ismi, `src="sayfa1.html"` bölümünde gösterilecek olan sayfa belirtilmiştir. Birde eğer sayfa içinde navigasyon kullanacaksak `name` özelliğini belirtmemiz gerekir. Çünkü frame içindeki sayfaya link vermek için sayfa ismi değil frame ismi kullanılır.

71) Herhangi bir linkin gideceği sayfayı frame içinde göstermek istiyorsak, `target` özelliğinde frame adı kullanılır.

` `

72) `frameset` özelliklerinden `border` özelliği ile frame'ler arasındaki çerçeve kalınlığı belirtilir. (`<frameset rows="100,*" border=10>`)

Çerçeveyi renklendirmek içinse `bordercolor` özelliği kullanılır. (`<frameset rows="100,*" border=10 bordercolor=red>`)

Not: çerçevesiz sabit bir görüntü için `border=0` değeri atanır.

73) `<noframes>`/`</noframes>` etiketi ile frame yapısını desteklemeyen web tarayıcıları veya herhangi bir sebepten dolayı frame'lerin görüntülenememesi gibi durumlarda frame desteklenmiyorsa kullanıcıya gösterilmek istenen mesaj yazılabilir. `<noframes>` etiketi `<frameset>` içinde tanımlanır.

74) frame'lerde `noresize` özelliği kullanılırsa frame sabit hale gelir. Çerçevesi konumlandırılmaz.

`<frame name="anasayfa" src="anasayfa.html" noresize>`

75) `scrolling` özelliği ile sayfanın kenarında bulunan kaydırma çubuğunu kaldırabilirsiniz. `yes` ve `no` değerlerini alır. `no` değeri verirseniz kaydırma çubuğu ortadan kalkar.

76) Tablolardaki gibi frame'lerde de iç içe kullanım yapılabilir. Bu yapı bize kolaylık sağlamaktadır.

77) `<iframe>`/`</iframe>` etiketi ile sayfa içinde istediğiniz herhangi bir yerde başka bir sayfayı görüntülemenizi sağlar.

`<iframe name="ic_frame" src="sayfa2.html" height=200 width=300></iframe>`

<<<<<<<<FORMLAR>>>>>>>>>>

Bütün form elemanları `<form></form>` etiketi arasında tanımlanır. `<form>` etiketine ait iki özellik bulunur. Bunlar; `method` ve `Action` özellikleridir.

Method özelliği : Bu özellik form elemanlarını nasıl gönderileceğini belirtir. İki değer alır: `Bunlar`; `Post` ve `Get` değerleridir.

- 1.) `Post` : Form elemanları gönderilirken url içinde gösterilmez.
- 2.) `Get` : Form elemanları gönderilirken url içinde gösterilir. `Get` metodunda form elemanları url içinde gösterileceği için güvenlik açısından pek kullanılmaz. Formlarda daima `Post` metodunu kullancaz.

Action özelliği : Bu özellik ile formdaki elemanlara uygulayacağımız işlemi belirleyeceğiz. `Action` özelliği de iki değer alır.

- 1.) `mailto` : `mailto` özelliği kullanıldığında formdaki elemanlar belirtilen mail adresine gönderilir.
- 2.) `URL` : url ise herhangi bir web sayfasının ismi olabilir. Bu değer ile formdaki elemanlar belirtilen web sayfasına gönderilir. Değişkenlere aktarılıp kullanılabilir.

Form Elemanları : Form elemanları `<form></form>` etiketi arasında tanımlanır. Form içinde veri girişi veya verileri göndermek için kullanılır. Form elemanı tanımlanırken `<input type="eleman_tipi">` etiketi kullanılır. `Type` özelliği ile form eklenecek olan elemanın tipi belirlenir. Bu bir `Textbox`, `Button`, `Checkbox`, `Radiobutton` vs. gibi elemanlar olabilir.

1) **Textbox Elemanı :** Formlarda kullanıcıdan veri girişi almak için kullanılır. `<input type="text">` etiketiyle tanımlanır. `<input type="text" name="textbox1"/>` şeklinde. `size` özelliği ile `Textbox`'ın genişliği belirlenebilir. `maxlength` özelliği ile `Textbox`'a girilecek maksimum karakter sayısı belirlenebilir. `value` özelliği ile `Textbox`'a varsayılan bir değer atanabilir. Bu değer sayfa ilk yüklendiğinde `Textbox` içinde görülen değerdir ve kullanıcı değiştirmedikçe `Textbox` içinde bu değer kalır.

2) **Password Elemanı :** Bu etiket ile şifreli girişler için maskelenmiş `Textbox` kullanabilirsiniz. `<input type="password" name="sifre"/>`

3) Textarea Elemanı : Textarea elemanı ile daha geniş çağlı yazıları sayfamızda gösterebiliriz. Bu eleman normal Textbox elemanının genişletilmiş halidir. Bu eleman `<textarea></textarea>` etiketleri arasında tanımlanır. `cols` ve `rows` özellikleri ile Textarea elemanının genişliği ve yüksekliği satır ve karakter sayısı cinsinden belirlenir. Yani elemanın kaç satır ve her bir satırın kaç karakter genişliğinde olduğunu belirtebilirsiniz. `<textarea name="text1" rows=10 cols=30>` şeklinde. `wrap` özelliği ile Textarea içindeki yazının otomatik alt satıra geçip geçmeyeceğini belirtebilirsiniz. `wrap` özelliği 3 değer alır.

a) `soft` : Varsayılan değer budur. Yazılar sadece Textarea içinde kaydırılır. Fakat form bazında bu işlem yapılmaz.

b) `hard` : Yazılar hem Textarea hemde form bazında kaydırılır.

c) `off` : `wrap` işlemi kapatılır. Bu özellik kullanıldığında alt tarafta kaydırma çubuğu oluşur. Çünkü yazının yatay olarak tümünü göremeyebiliriz.

`readonly` özelliği ile normalde Textarea içinde yazdığınız yazıları kullanıcı silebilmektedir. Hatta bu yazılara yeni yazılar ekleyebilmektedir. Bazı durumlarda Textarea içinde yazan yazının sadece okunabilir olmasını istiyorsanız; `readonly` özelliği kullanılır. Bu özellik sayesinde kullanıcı Textarea içindeki yazılar üzerinde ekleme, silme gibi işlemler yapamaz.

```
<textarea name="text1" rows=10 cols=40 readonly> </textarea>
```

4) Checkbox Elemanı : Belli seçenekler arasından bir veya birkaçını seçmek için kullanılan yapıdır. Bir form doldururken kullanıcıdan çeşitli seçenekleri işaretlemesini istiyorsanız. Checkbox kullanmalısınız. Örneğin; kullanıcının hangi programlama dilleri bildiğini öğrenmek için bunları Checkbox ile gösterip, kullanıcının seçim yapmasını sağlayabiliriz. Temel anlamda Checkbox oluşturmak için (`<input type="checkbox"/>` Gösterilecek yazı) kodu kullanılır. `value` özelliği ile numara değeri veririz. Web programlama dillerinde bu numaralara bakarak işlem gerçekleşir. Birde aynı grup altında değerlendirdiğimiz Checkbox'lar için aynı ismi belirtmemiz gerekir. Bu ismi `name` özelliği ile belirtiriz. (`<input type="checkbox" name="p_dil" value="1"/>`Java) gibi. `checked` özelliği ile form yüklenirken varsayılan olarak işaretli olmasını istediğimiz seçenekleri belirtebiliriz. (`<input type="checkbox" name="p_dil" value="1" checked/>`Java) gibi.

5) Radio Elemanı : Form içinde kullanıcıdan tek bir seçim yapılması isteniyorsa Radio elemanı kullanılır. Checkbox gibi `value` özelliği ile işaretlenen seçeneği öğrenebilirsiniz. Checkbox'tan farkı ise, aynı anda sadece 1 tane seçebilmektir.

(`<input type="radio" name="grup_ismi" value="deger"/>`) gibi. Radi elemanları arasında tek seçim yapılması isteniyorsa, bu elemanlar bir grup altında toplanmalıdır. Radio elemanları arasında grup oluşturmak için de name özelliği kullanılır. Gruplanmak istenen Radio elemanlarının name özelliklerine aynı değer verilir. Radio elemanları için mutlaka bir name değeri belirtilemelidir. Aksi takdirde hiç bir seçim yapılamayacaktır. checked özelliği checkbox'da olduğu gibi, bu özellik ile form yüklenirken varsayılan olarak işaretli olmasını istediğiniz seçenekleri belirtebilirsiniz. (`<input type="radio" name="grup_ismi" value="deger" checked/>`) gibi.

6) Select Elemanı : Form içinde açılan bir liste kullanmak için bu eleman kullanılır. Açılan listeden tek bir tane değer seçilebilir.

Özellikle çok fazla seçeneğe sahip bölümler için kullanılır. Çünkü bu seçenekler form içinde göstermek aşırı yer kaybına sebep olur.

Bunu engellemek için tüm seçenekler bir liste içinde saklanır ve tıklandığı zaman liste açılır. Listeler `<select></select>` etiketi ile

tanımlanır. Listeye bir isim vermek için name özelliği kullanılır. Bu özellik ile daha sonra listemizdeki değerleri programsal olarak

elde edebiliriz. Liste elemanları ise `<option></option>` etiketi ile tanımlanır ve her bir Option elemanı için value değeri tanımlanır.

Bu değer liste elemanlarının değerini belirtir.

Örnek :

```
<select name="liste">
<option value="1">Birinci Eleman</option>
<option value="2">İkinci Eleman</option>
<option value="3">Üçüncü Eleman</option>
</select>
```

selected özelliği ile form yüklendiğinde ekrana herhangi birinin seçili gelmesini istiyorsak bu özelliği kullanırız.

(`<option value="3" selected>Üçüncü Eleman</option>`) şeklindedir. size özelliği ile liste elemanlarının kaç tanesinin görüntüleneceğini

belirleyebiliriz. Varsayılan olarak 1 değerini alır. Bu değeri değiştirerek gösterilecek eleman sayısını değiştirebiliriz. Tabi bu durumda

açılır liste özelliğini kaybedecektir. Sabir bir Listbox haline gelecektir. `<optgroup>` etiketi ile liste içinde gruplar oluşturabiliriz.

Gruplar `<optgroup></optgroup>` etiketleri arasında tanımlanır. Etikete ait Label özelliği ise grup için başlık tanımlar.

(`<optgroup label="programlama"> <option value="1"> C# </option> </optgroup>`) şeklindedir.

HTML içinde tanımladığımız canvas'ın Havascript ile kullanılabilmesi için canvas'a bir id değeri veriyoruz. Bu id değeri ile javascript içinde canvas elementini görüp üzerinde işlem yapabiliyoruz.

JavaScript :

```
var canvas=document.getElementById("bizim_canvas");
```

HTML5 :

```
<canvas id="bizim_canvas" width="300" height="200"></canvas>
```

Canvas elementine ait içeriği belirtmek için getContext(contextId) metodu kullanılır.

```
var ctx=canvas.getContext("2d");
```

Canvas için 2D contx belirtiyoruz. HTML5 canvas için henüz 3D contex desteklememektedir.

2.) <video> Elementi : Bu element ile herhangi external bir object kullanmadan direk HTML içinde video oynatabiliriz.

<video src="video.ogg"width="300"height="150"> gibi. controls özelliği ile video butonlarını görüntüleyebiliriz.

<video src="video.ogg"width="300"height="150" controls> şeklinde.