

ETİKET	AMAÇ	KULLANIM
<b>Css söz dizimi</b>	Css kodlarını yazma biçimine css söz dizimi denir. Söz dizimi 3 bölümden oluşur. Selector { property : value; } 1-) Selector : Seçici olarak adlandırılır ve html sayfasındaki her hangi bir html etiketini seçmek için kullanılır. Css kodunu etiket ile ilişkilendirmek için ilk olarak etiketi seçmemiz gerekir. 2-) Property: Kullanacak olduğumuz css özellik tipidir. 3-) Value : Css özelliğinin alacağı değeri temsil eder.	p { color:red; } ** Sayfadaki tüm etiketlerinin yazı rengini kırmızı yapar.  ** p: selector, color: property, red: value h1 { font-size : 40px; color:blue; }  ** Sayfadaki tüm etiketlerinin font büyüklüğünü 40px ve yazı rengini mavi yapar.
<b>Css Yazım Türleri</b>	Css kodlarımızı farklı amaçlara göre 3 farklı şekilde tanımlayabiliriz. • Inline Css • Internal • External Css	
<b>Inline Css</b>	Sadece tanımlandığı etiket için geçerli olmasını istediğimiz css kodlarımızı inline css olarak tanımlarız. ** Inline css kodları direk etiket içine yazılır ve sadece o etikete uygulanır. ** Inline olarak yazılan css kodları aynı özellik için yazılmış diğer css kodunu ezer ve etkin olur.	<span style="color:red;"> text </span> ** <span> etiketi içindeki yazıları kırmızı yaparız. <a href="#" style="color:blue;text-decoration:none;"> tıklayınız. </a> ** <a> etiketi için 2 tane css kodunu inline olarak yazdık. * text-decoration:none; demekle link yazısı altındaki alt çizgiyi kaldırdık. * color ile link yazısını mavi yaptık.

<b>Internal Css</b>	<p>Internal css etiketleri arasına etiketleri eklenerek kullanılır.</p> <p><b>**</b> Internal olarak tanımlanan css kodları sadece bulunduğu html sayfasını ilgilendirir.</p>	<p><b>**</b> index.html içindeki h1 etiketi için yazılan css kodu sadece index.html sayfasındaki h1 etiketini ilgilendirir. about.html sayfası içindeki h1 etiketine de aynı css kodunu eklememiz gerekiyorsa aynı işlemi about.html sayfasında da tekrarlamamız gerekir.</p>
<b>External Css</b>	<p>Eğer tanımladığımız css kodları birden fazla html sayfasını ilgilendiriyor ise bu durumda .css uzantılı bir dosya oluşturup bu dosya içerisine css kodlarımızı tanımlamamız gerekir.</p> <p><b>**</b> Harici .css uzantılı dosyaya yazdığımız css kodlarını hangi html sayfasına eklemek istiyorsak o html sayfası ile css dosyasını bir birine &lt;link&gt; ile bağlamamız gerekiyor.</p> <pre>&lt;head&gt; &lt;link rel="stylesheet" href="style.css"&gt; &lt;/head&gt;</pre>	<p>style.css dosyasına aşağıdaki css kodlarını ekleyelim.</p> <pre>h1 { font-size :32px; color :blue; } p { color : red; font-size: 16px; }</pre>
<b>Tagname selector</b>	Etiket ismine göre yaptığımız seçim işlemidir	<pre>p {color:blue; }</pre> <p><b>**</b> sayfadaki tüm etiketleri seçilir.</p>
<b>id selector</b>	<p>Etiketlere verdiğimiz benzersiz isme id denir.</p> <p><b>**</b> Bir html dökümanında aynı id ismiyle sadece bir etiket olmalıdır.</p> <p><b>**</b> id ile seçim yaparken # işareti kullanılır.</p>	<pre>&lt;h1 id="header"&gt; sayfa başlığı &lt;/h1&gt; #header { font-size:40px; }</pre> <p><b>**</b> header id'sine sahip sadece bir etiket olmalıdır.</p>

<b>class selector</b>	<p>Class ismine göre yaptığımız seçme işlemidir.</p> <p><b>**</b> bir class ismi bir html dökümanında istenildiği kadar kullanılabilir.</p> <p><b>**</b> class ile seçme işleminde “.” (nokta) işareti kullanılır.</p> <p><b>**</b> bir etikete birden fazla class verilebilir</p>	<pre> &lt;ul&gt;   &lt;li class="active"&gt; item 1 &lt;/li&gt;   &lt;li&gt; item 2 &lt;/li&gt;   &lt;li class="active"&gt; item 3 &lt;/li&gt; &lt;/ul&gt; &lt;div class="active f30"&gt;   arka plan yeşil olur. &lt;/div&gt; .active {   background-color : green; } .f30{   font-size:30px; } </pre> <p><b>**</b> active class’ ına sahip tüm elemanların zemin rengi yeşil olur.</p>
<b>Group selector</b>	<p>Css Group Selector' larını bir kapsam içinde aradığımız etiketler için kullanıyoruz. Örneğin farklı seviyelerde olan bir etiketleri için kullanabiliriz. Başlıca group selector kullanımları;</p> <p>1 : (*) =&gt; tüm etiketler</p> <p>2 : (p) =&gt; tüm p etiketleri</p> <p>3 : (div p) =&gt; div içindeki tüm p etiketleri</p> <p>4 : (div,p) =&gt; tüm div ve tüm p etiketleri</p> <p>5 : (div &gt; p) =&gt; üst etiketi div olan tüm p etiketleri</p> <p>6 : (p ~ div)=&gt; p ile aynı seviyede tüm div etiketleri</p> <p>7 : (p + div)=&gt; p etiketinden sonra gelen aynı seviyedeki div etiketi</p>	<pre> &lt;div&gt;   &lt;p id="first"&gt; &lt;/p&gt;   &lt;p class="description"&gt; &lt;/p&gt; &lt;/div&gt; &lt;p&gt; &lt;/p&gt; div p {   /* div altındaki tüm p etiketleri */ } div #first{   /* div altında id' si first olan eleman */ } div .description{   /* div altında class' ı description olan eleman */ } </pre>

<b>Pseudo Class Selector</b>	<p>html etiketlerinin durumunu kontrol ettikten sonra uygulayacağımız css kodları için kullanılır.</p> <p><b>**</b> css kodları bir etiketin tamamı için kullanılır.</p> <pre>selector:pseudo-class {   property:value; }</pre>	<pre>p::first-line { } <b>**</b> &lt;p&gt; etiketinin kapsamındaki yazı içinde ilk satırı seçmek istediğimizde kullanırız. Bir kapsamın içindeki bir parçayı seçtiğimize dikkat ediniz. p::first-letter { } <b>**</b> &lt;p&gt; etiketi içindeki yazının ilk harfini seçer</pre>
<b>font-size</b>	Font türü için kullanılır	<pre>p {   font-family: "Times New Roman", Times, serif; }</pre>
<b>font-weight</b>	<p>Yazı kalınlığını temsil eder.</p> <p>Alabileceği değerler : normal, bold, bolder</p>	<pre>p { font-weight:bold; }</pre>
<b>color</b>	<p>Yazı rengi için kullanılır.</p> <p><b>**</b> yazı rengi için önceden tanımlanmış renk isimlerini kullanabiliriz.</p> <p><b>**</b> RGB (Red,green,blue) 3 ana temel renk için 0-255 arasında değer alır.</p> <p><b>**</b> Hexadecimal renk tanımlaması yapılabilir. #ffffff ya da # 000000.</p>	<pre>p {   color:white; } p {   color:rgb(255,255,255); } p {   color: ffffff; }</pre>
<b>background-color</b>	Arka zemin rengi için kullanılır.	<pre>div {   background-color : blue; }</pre>

<b>width</b>	genişlik için kullanılır. ** block ve inline-block elemanlar için width kullanılır, inline etiketlere genişlik ve yükseklik verilmez.	#container { width: 600px; }
<b>height</b>	yükseklik için kullanılır. ** block ve inline-block elemanlar için width kullanılır, inline etiketlere genişlik ve yükseklik verilmez.	#container { height: 200px; }
<b>max-width max-height</b>	Bir nesnenin alabileceği maksimum genişlik ya da maksimum yükseklik değeri için kullanılır	div { max-width: 600px; max-height: 200px; }
<b>float</b>	Nesneleri hizalamak için kullanılır. Genelde 'leri yan yana getirmek için kullanırız. Alabileceği değerler ; Left => sola hizalanır. right => sağa hizalanır none => hizalama işlemi yapılmaz.	<div id="first" class="box"></div> <div id="second" class="box"></div> <div id="third" class="box"></div> .box { width: 150px; height: 150px; margin: 2px; } #first { background: red; float:left; } Detay #second { background: blue; float:left; } #third { background: green; float:left; }

		** Örneğimizde 3 div etiketini yan yana getiriyoruz.
<b>display:inline;</b>	<p>inline etikete çevirmek için kullanırız.</p> <p>Örneğin div etiketi block etiketi iken display: inline; kullanımı ile div etiketini inline etiketine çevirebiliriz.</p> <p>** Inline etiketlere genişlik ve yükseklik veremeyiz. (width,height).</p> <p>** Inline etiketlere genişlik ve yükseklik veremediğimizden dolayı background-color sadece yazının arkasını çevreler.</p> <p>** Inline etiketlere sadece sağ ve soldan margin ya da padding değerleri verebiliriz.</p>	<pre>li {   display:inline; }</pre>
<b>display : block</b>	<p>Inline etiketleri block etiketlerine çevirebiliriz.</p> <p>** block etiketlere genişlik vermediğimizde satırı olduğu gibi kaplar ve bir sonraki nesne bir sonraki satırdan devam eder.</p> <p>** block etiketler için yükseklik ve genişlik değeri verebiliriz. Dolayısıyla aynı satırda bir den fazla block elemanı yan yana getirebilmek için bir genişlik kullanmalıyız.</p> <p>** block elemanları örneğin &lt;div&gt; etiketlerini yan yana getirmek için float ve width kullanmalıyız.</p>	<pre>a {   display:block; }</pre>
<b>display : inlineblock;</b>	<p>display:inline-block olan bir nesne hem inline gibi satırda yan yana gelir hem de bu nesnelere block nesneler gibi yükseklik ve genişlik değerleri verebiliriz.</p>	<pre>li {   display:inline-block;   width: 200px; }</pre> <p>** li etiketi için sabit bir genişlik vererek her elemanın yan yana gelmesini sağlayabiliriz. Float kullanmadığımıza dikkat ediniz.</p>

<b>display:none;</b>	display:none; uygulamadığımız nesne ekranda görünmez. Html dökümanı içinde yer kaplamaz. Dolayısıyla bir sonraki nesne görünmez olan nesnenin yerini alır. visibility: hidden; ise gene görünmez hale gelir ancak nesnenin yeri diğer nesne ile doldurulmaz. Varmış gibi yer kaplamaya devam eder. visibility: none   visible   hidden   collapse;	div { display:none; } div { visibility:hidden; }
<b>Position</b>	Web sayfalarında nesnelerin konumlarını ve davranış biçimlerini float ve display özellikleriyle değiştirebiliriz. Bu amaç için bir diğer özelliğimiz ise position özelliğidir. Alabileceği değerler : Position : static   relative   fixed   absolute   sticky	
<b>Position:static;</b>	Html etiketleri başlangıç ayarı yani varsayılan olarak static biçimde davranış gösterirler. Yani normal akış içinde birbirlerinden etkilenerek davranış gösterirler. ** Static nesneler için top, right, bottom ve left konumlandırma özellikleri kullanılamaz.	
<b>Position:relative;</b>	Bir nesneye position:relative; dediğimizde nesne gene normal akış içerisinde olmaya devam eder ancak top, right, bottom ve left özelliklerini kullanma fırsatı verir. ** relative olan nesne o anki konumu neredeyse oradan itibaren konumlandırma yapar.	.box{ width: 150px; height: 150px; background: blue; position:relative; top:50px; } ** 150x150 boyutlarındaki mavi kutu sayfanın sol üst köşesinden 50px aşağıya iner.
<b>Position:absolute;</b>	Bir nesnenin position özelliğini absolute yaptığımızda o nesne normal akış içinden çıkar ayrıca 4 köşeden konumlandırma yapmamızı sağlar. Relative ile Absolute nesnelerin farkı normal akış içinde olup olmadığı farkıdır.	

	** absolute olan nesne kendinden önce gelen nesnelerin kapladığı yer ile ilgilenmeden sayfanın sol üst köşesinden itibaren konumlandırma yapar.	
<b>Position:fixed;</b>	Bir nesneyi yatay ve dikey kaydırma çubuklarından etkilenmeden sabit bir yerde konumlandırmak için kullanırız. Örneğin sayfanın sağ alt köşesinde bulunan canlı sohbet menüsü gibi.	div { position: fixed; bottom: 0; right: 0; width: 250px; border: 4px solid red; }
<b>Position:sticky;</b>	Position: sticky kaydırma çubuklarını takip eder ve istenilen pozisyon yakalandığında nesneyi o noktada sabitler. Örneğin bir nesne için yukarıdan 0 konumuna geldiğimiz anda nesnenin sabitlenmesini isteyebiliriz. Genelde menüler için sticky position kullanılır.	div { position: sticky; top: 0; width: 250px; border: 4px solid red; }
<b>Box Model</b>	Css Box Model yani css kutu model, html 'de bulunan her bir nesnenin etrafında olan bir kutuyu temsil eder. Bu kutu modeli nesnelerin birbiri ile arasındaki mesafe (margin), kutu ile içerik arasındaki mesafe (padding) ve kutunun etrafındaki kenarlık (border) ile ilgili çalışmaları temsil eder.	
<b>Margin</b>	Css Margin, html nesnelerinin birbiri ile olan mesafelerini belirtir. ** margin : top right bottom left; => şeklinde 4 köşeden margin değerlerini tek tek belirleyebiliriz. ** margin : top-bottom right-left; => iki değer şeklinde bir kullanım yapabiliriz. ** margin : top-right-bottom-left; => şeklinde 4 köşe için tek bir değer kullanımı da yapabiliriz	div { margin:10px 20px; } ** üst-alt için 10px ve sağ-sol için ise 20px' lik bir boşluk bırakmış oluruz. div { margin:10px; } ** 4 köşe için 10px margin değeri vermiş oluruz



<b>Padding</b>	Css padding ile nesnelerin etrafında bulunan kenarlık ile arasındaki mesafeyi ayarlayabiliyoruz. ** margin' de olduğu gibi 4 köşeden padding kullanımı yapabiliriz.	.box{ width:250px; background-color:red; padding:20px; }
<b>Border</b>	Css border html nesnelerinin etrafına kenarlık eklemek için kullanılır. ** Border' a verebileceğimiz 3 özellik vardır. Bunlar border-style, border-width ve border-color özellikleridir. ** border-top, border-right, border-bottom, border-left şeklinde 4 köşeden ayrı ayrı kenarlık ekleyebiliriz.	div { border:1px solid red; } ** 1px çizgi kalınlığı, solid çizgi tipi ve kırmızı çizgi rengi kutunun etrafına eklenir.
<b>Box Sizing</b>	Bir nesnenin kapladığı alan nesnenin genişliği + padding + border değerlerinin toplamıdır. Ancak box-sizing:box-box dediğimizde bu hesaba padding ve border katılmayacaktır. Nesnenin kapladığı toplam alan width ya da height ile belirlenecektir.	* { box-sizing: border-box; } ** tüm etiketler için border-box özelliğini aktif ediyoruz.
<b>text-decoration:none;</b>	Alt çizgiyi kaldırır. ** Alabileceği diğer değerler. text-decoration:underline; => altını çizerek text-decoration:overline; => üstünü çizerek text-decoration:line-through; => ortasını çizerek	a { text-decoration:none; } ** <a> etiketinin varsayılan özelliği olan alt çizgiyi kaldırır.
<b>Text Transform</b>	text-transform: uppercase; => büyük harfe çevirir text-transform: lowercase; => küçük harfe çevirir text-transform: capitalize; => her kelimenin baş harfini büyük harfe çevirir.	.uppercase { text-transform: uppercase; } .lowercase { text-transform: lowercase; } .capitalize {

		text-transform: capitalize; }
<b>Text Indentation</b>	Paragrafın ilk cümlesinin ne kadar içeriden başlayacağını belirtir.	p { text-indent: 50px; } ** paragraf 50px içeriden başlat.
<b>Letter Spacing</b>	Harf aralıklarını belirler.	p { letter-spacing: 1.2px; } ** her harf arasındaki mesafe.
<b>Line Height</b>	Satır yüksekliğini belirtir.	.small { line-height: 0.7; } .big { line-height: 1.8; }
<b>Word Spacing</b>	Kelimeler arasındaki boşluk.	h1 { word-spacing: 5px; } h2 { word-spacing: -5px; } ** negatif değer ile kelimeler birbirine çok yaklaşır.
<b>Text Align</b>	text-align => bir block eleman içindeki yazıların ya da içeriklerin hizalanmasını sağlar. text-align : left   center   right   justify; ** hizalama işlemi block etiketlere uygulanır. Eğer inline bir etiketleri için text-align özelliği işlevsel değildir. Eğer ki inline etiket grubunu hizalamak istersek bu durumda inline etiket grubunu bir block etiketi (<div>,<p>) içine alıp hizalama işlemini yapabiliriz.	<p> Html ve Css3 Eğitimi </p> p { text-align: center; }

<b>icon</b>	<p>İcon kullanımı için font-awesome sitesini ziyaret edelim.</p> <p>** How-to-use linkinden icon kütüphanesini sayfamızın kısmına kopyalayalım.</p> <p>** Icons menüsünden ise kullanmak istediğimiz iconun <i>etiketini alalım</i></p>	<pre>&lt;i class="fas fa-allergies"&gt;&lt;/i&gt;</pre> <p>Örnek bir icon kullanımı.</p>
-------------	---	--