

اصول طراحی صفحات وب

هدف: طراحی وب سایت‌های ایستا

CSS

نسخه: ۱.۳۸

kishuniv@montajab.com

<http://montajab.com/kishuniv>

The logo for CSS3, featuring the letters 'CSS' in black above a blue shield containing a white '3'.

Cascading Style Sheets

CSS زبانی برای توصیف نحوه نمایش اسناد نوشته شده توسط یک زبان نشانه گذاری مانند HTML است. با استفاده از این زبان، استایل و چگونگی نمایش عناصر بر روی صفحه نمایش (انواع دستگاه‌ها با اندازه‌های مختلف صفحه)، کاغذ (در چاپ) و دیگر رسانه‌ها را تعیین می‌نماییم. برای استفاده از CSS سه روش وجود دارد:

1. Inline style
2. Internal style sheet
3. External style sheet



1. Inline style

از این روش فقط برای استایل دادن به یک عنصر (و به ندرت) استفاده می‌شود. جهت بکارگیری این روش، صفت style را به عنصر مورد نظر اضافه کرده و ویژگی‌های CSS را درون آن تعریف می‌کنیم.

```
<p style="color:blue;">This is a paragraph.</p>
```



Cascading Style Sheets

1. Inline style

```
<html>
<body >
  <p style="color:red;">This is a paragraph.</p>
  <p style="color:green;">This is a paragraph.</p>
  <p style="color:blue;">This is a paragraph.</p>
</body>
</html>
```

This is a paragraph.

This is a paragraph.

This is a paragraph.



2. Internal style sheet

هنگامی که بخواهیم بر روی یک تک صفحه، یک استایل خاص را اعمال کنیم، از این روش استفاده می‌شود. در این روش از تگ `<style>` در بخش head استفاده کرده و کدهای CSS را درون آن تعریف می‌کنیم.



Cascading Style Sheets

2. Internal style sheet

```
<html>
  <head>
    <style>
      p { color:blue; font-family:tahoma; }
    </style>
  </head>
  <body >
    <p>Text #1</p>
    <div>Text #2</div>
    <p>Text #3</p>
  </body>
</html>
```



Text #1

Text #2

Text #3



3. External style sheet

رایج‌ترین روش استفاده از CSS، از طریق یک فایل جداگانه از نوع *.css است. این راه، ایده‌آل‌ترین روش برای استایل دادن به یک یا چندین صفحه است. با این روش می‌توان ظاهر یک وب سایت را با ویرایش یک فایل، تغییر داد.

CSS

Cascading Style Sheets

استایل‌های تعریف شده در یک فایل متنی CSS، با ذخیره پسوند CSS می‌شوند. برای استفاده از یک فایل CSS، باید آن را به صفحه مورد نظر، توسط یک لینک ارجاع داد:

<head>

Relationship: ارتباط بین سند جاری و سند لینک شده را تعیین می‌کند.

<link rel="stylesheet" type="text/css" href="cssfile.css"/>

</head>

نوع فایل

نام فایل CSS

```
cssfile.css
p {
  color:blue;
  font-family:tahoma;
}
```

CSS

Cascading Style Sheets

CSS Syntax

Selector {*CSS_property*:*CSS_value*; *CSS_property*:*CSS_value*;}

p {color:blue; font-family:tahoma; }

مثال:

Selector، عنصر HTML که می‌خواهیم به آن استایل بدهیم، را مشخص می‌کند. در بلاک روبری آن، که با آکولاد مشخص شده است، هر تعریف CSS، با علامت ; بسته می‌شوند. هر تعریف CSS، شامل یک نام ویژگی و یک مقدار برای آن ویژگی است، که با علامت : از هم جدا می‌شوند.

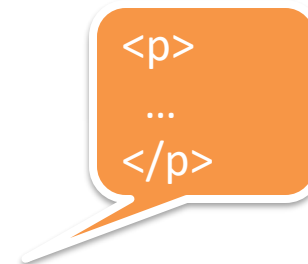
CSS

Cascading Style Sheets

```
p {color:blue; font-family:tahoma; }
```

برای خوانایی بیشتر کد، می‌توان هر تعریف را در یک سطر جداگانه قرار داد:

```
p {  
    color:blue;  
    font-family:tahoma;  
}
```



در این مثال تمامی عناصر پاراگراف در صفحه با رنگ آبی و فونت tahoma نمایش داده خواهند شد.

CSS Selectors

Selectorها امکان انتخاب و دستکاری عناصر HTML را می‌دهند.

Selectorها جهت انتخاب و یا به عبارتی یافتن عناصر HTML بر اساس id، کلاس، نوع، ویژگی و مانند آنها، مورد استفاده قرار می‌گیرند و عبارتند از:

Element Selector:

بر اساس نام عناصر HTML، آنها را مشخص می‌کند:

```
h1 {  
  color:blue;  
  font-family:tahoma;  
}
```

در این مثال تمامی عناصر <h1> در صفحه با رنگ آبی و فونت tahoma نمایش داده خواهند شد.

CSS Selectors

در مواردی که چندین نوع عنصر با یک استایل داریم، می‌توان نام Selectorها را با ویرگول از هم جدا کرده و همه را در یک بخش ادغام نمود:

```
h1 {  
    color:blue;  
    text-align:center;  
}  
h2 {  
    color:blue;  
    text-align:center;  
}
```

```
h1 ,h2 {  
    color:blue;  
    text-align:center;  
}
```

CSS Selectors

id Selector:

بر اساس نام صفت id عنصر HTML، آن را مشخص می‌کند.

یک id در یک صفحه، یکتاست. در نتیجه برای مشخص کردن یک عنصر خاص از صفت id آن استفاده می‌کنیم (به عبارت ساده، از id برای آدرس‌دهی یک عنصر استفاده می‌کنیم).

برای انتخاب یک عنصر بر اساس id آن، از یک علامت # قبل از نام id استفاده می‌کنیم:

```
#myElement {  
    color:blue;  
    font-family:tahoma;  
}
```

در این مثال عنصری که صفت id آن برابر myElement است، در صفحه با رنگ آبی و فونت tahoma نمایش داده خواهند شد.

CSS Selectors

id Selector:

```
<html>
<head>
  <style>
    #id3 { color:blue; font-family:tahoma; }
  </style>
</head>
<body>
  <p id="id1">Text #1</p>
  <div id="id2">Text #2</div>
  <p id="id3">Text #3</p>
</body>
</html>
```



Text #1
Text #2
Text #3

CSS Selectors

Class Selector:

بر اساس نام صفت class عناصر HTML، آن‌ها را مشخص می‌کند.

برای تعریف یک class در CSS، از یک نقطه قبل از نام آن استفاده می‌کنیم:

```
.center {  
    color:blue;  
    text-align:center;  
}
```

حال برای استفاده از آن class در یک عنصر HTML بدین گونه عمل می‌کنیم:

```
<p class="center">Center-aligned paragraph.</p>
```

```
<h2 class="center">Center-aligned heading2.</h2>
```

این کلاس باعث می‌شود تا پاراگراف و سرتیتر ۲ در وسط صفحه و با رنگ آبی نمایش داده شوند.

CSS Selectors

Class Selector:

```
<html>
<head>
  <style>
    .cBlue { color:blue; font-family:tahoma; }
  </style>
</head>
<body>
  <p>Text #1</p>
  <div class="cBlue">Text #2</div>
  <p class="cBlue">Text #3</p>
</body>
</html>
```



Text #1

Text #2

Text #3

CSS Selectors

Class Selector:

```
<html>
<head>
  <style>
    .cBlue { color:blue; font-family:tahoma; }
    .cBLUE { color:blue; }
  </style>
</head>
<body >
  <p>Text #1</p>
  <div class="cBLUE">Text #2</div>
  <p class="cBlue">Text #3</p>
</body>
</html>
```



Text #1

Text #2

Text #3

CSS Selectors

Class Selector:

همچنین می‌توان برای عناصر مختلف، کلاس‌هایی با یک نام، اما با ویژگی‌های متفاوت ایجاد نمود، یا به عبارتی یک کلاس را به گونه‌ای تعریف کرد که فقط بر روی یک نوع از عناصر اعمال شود؛ برای این کار ابتدا نام عنصر، سپس یک نقطه و در آخر نام کلاس را تعیین می‌کنیم:

```
p.center {  
    color:blue;  
    text-align:center;  
}
```

در این مثال کلاس center فقط بر روی پاراگراف‌هایی که، این کلاس در آنها تعریف شده است، تاثیر می‌گذارد:

```
<p class="center">Center-aligned paragraph.</p>
```

○ از id برای آدرس‌دهی یک عنصر و از class برای آدرس‌دهی گروهی از عناصر استفاده می‌کنیم.

CSS Selectors

Class Selector:

```
<html>
<head>
  <style>
    p.cBlue { color:blue; font-family:tahoma; }
  </style>
</head>
<body>
  <p>Text #1</p>
  <div class="cBlue">Text #2</div>
  <p class="cBlue">Text #3</p>
</body>
</html>
```



Text #1

Text #2

Text #3

CSS Selectors

اولویت آخرین استایل CSS تعریف شده بیشتر است، به عنوان مثال اگر رنگ پاراگراف در فایل CSS (در روش External style sheet) آبی تعیین شده باشد و در خود فایل HTML مورد نظر با روش Internal style sheet رنگ پاراگراف نارنجی در نظر گرفته شده باشد:

```
<head>
  <link rel="stylesheet" type="text/css" href="mystyle.css">
  <style>
    p { color: orange;}
  </style>
</head>
```



رنگ پاراگراف نارنجی
نمایش داده می شود.

```
<head>
  <style>
    p { color: orange;}
  </style>
  <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```



رنگ پاراگراف آبی
نمایش داده می شود.

اولویت روش Inline style ؟ 

CSS Selectors

به طور کلی اولویت استایل‌ها به صورت آشنایی است و بیشترین اولویت مربوط به آخرین استایل مشابه است، به عبارتی استایل آخر جایگزین استایل مشابه قبلی می‌شود. اولویت‌ها به ترتیب زیر است:

1. Inline Style

بالاترین اولویت را دارد.
در صفت **style** عنصر قرار می‌گیرد.

2. External style sheet/ Internal style sheet

در بخش **head** صفحه **HTML** قرار می‌گیرند.
هر کدام که اول تعریف شود، اولویت کمتری دارد، و بعدی جایگزین آن می‌شود.

3. Browser default

در صورت عدم وجود موارد ۱ و ۲، پیش فرض مرورگر اعمال می‌شود.

CSS Selectors

Combinators

برای انتخاب عناصر مورد نظر، امکان استفاده از چندین Selector به صورت ترکیبی نیز وجود دارد؛ که برای توصیف ارتباط بین این Selector ها از Combinator ها استفاده می کنیم و عبارتند از:

1. Child Selector (>):

انتخاب تمامی عناصر مرتبطی که بی واسطه، فرزند عنصر والد تعیین شده هستند.

2. Descendant Selector (فاصله):

انتخاب تمامی عناصر مرتبطی که از نوادگان (زیرشاخه های) عنصر والد تعیین شده هستند.

3. Adjacent Sibling Selector (+):

انتخاب اولین عنصر مرتبط و مجاوری که بلافاصله بعد از عنصر تعیین شده آمده است.

4. General Sibling Selector (~):

انتخاب تمامی عناصر مرتبطی که بعد از عنصر تعیین شده آمده باشند.

CSS Selectors

Pseudo Classes

از شبه کلاس‌ها جهت تعیین استایل مورد نظر، برای حالت خاصی از عناصر، استفاده می‌گردد. به عنوان مثال برای تعیین استایل یک عنصر در هنگامی که فوکوس می‌شود و یا موس روی آن حرکت می‌کند.

Syntax:

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

CSS Selectors

Pseudo Classes

:active	:last-child	:read-only
:checked	:last-of-type	:read-write
:disabled	:link	:required
:empty	:not(selector)	:root
:enabled	:nth-child(n)	:target
:first-child	:nth-last-child(n)	:valid
:first-of-type	:nth-last-of-type(n)	:visited
:focus	:nth-of-type(n)	
:hover	:only-of-type	
:in-range	:only-child	
:invalid	:optional	
:lang(language)	:out-of-range	

CSS Selectors

Pseudo Elements

از شبه عناصر جهت تعیین استایل مورد نظر، برای بخش خاصی از عناصر، استفاده می‌گردد. به عنوان مثال برای تعیین استایل اولین حرف و یا اولین سطر یک عنصر و یا اضافه کردن محتوا، قبل و یا بعد از یک عنصر مورد استفاده قرار می‌گیرند.

Syntax:

```
selector::pseudo-element{  
    property:value;  
}
```

CSS Pseudo Elements

- ::after
- ::before
- ::first-letter
- ::first-line
- ::selection

قبل از CSS3، برای تعیین شبه عناصر نیز مانند شبه کلاس‌ها، از تنها یک دو نقطه استفاده می‌شد.

CSS Selectors

علاوه بر Selector های یاد شده، انواع دیگری نیز وجود دارند که عبارتند از:

*	:active	:in-range	:optional
element,element	::after	:invalid	:out-of-range
element element	::before	:lang(language)	:read-only
element>element	:checked	:last-child	:read-write
element+element	:disabled	:last-of-type	:required
element1~element2	:empty	:link	:root
[attribute]	:enabled	:not(selector)	::selection
[attribute=value]	:first-child	:nth-child(n)	:target
[attribute~=value]	::first-letter	:nth-last-child(n)	:valid
[attribute =value]	::first-line	:nth-last-of-type(n)	:visited
[attribute^=value]	:first-of-type	:nth-of-type(n) p:nth-of-type(2)	
[attribute\$=value]	:focus	:only-of-type	
[attribute*=value]	:hover	:only-child	

CSS Selectors

*

تمام عناصر انتخاب شده و در نتیجه استایل مربوط بر روی تمام عناصر اعمال می‌شود. در مثال زیر ویژگی `color`، که آبی تعیین شده، بر روی تمام عناصر اعمال می‌گردد:

```
* { color:blue; }
```

:not(selector)

تمام عناصر، بجز موارد تعیین شده توسط `Selector`، انتخاب می‌شوند. در مثال زیر ویژگی `color` بر روی تمام عناصر، بجز عناصر `p`، اعمال می‌گردد:

Selector از نوع `element` است و تمام عناصر `p` را انتخاب می‌کند.

```
:not(p) { color:blue; }
```

CSS Selectors

element,element

استایل مربوط بر روی تمام عناصر تعیین شده اعمال می‌شود. در مثال زیر ویژگی `color` بر روی تمام عناصر `p`، `h1` و `h2` اعمال می‌گردد:

```
p,h1,h2 { color:blue; }
```

element element

Descendant Selector

برای انتخاب عناصر درون عناصر دیگر مورد استفاده قرار می‌گیرد. در مثال زیر ویژگی `color` بر روی تمام عناصر `span` که درون عنصر `p` (و زیر شاخه‌هایش) قرار گرفته‌اند، اعمال می‌گردد:

```
p span { color:blue; }
```

CSS Selectors

element>element

Child Selector

عنصری که به صورت مستقیم (بدون فاصله) فرزند عنصر والد تعیین شده هستند، انتخاب می‌شوند. در مثال زیر ویژگی `color` بر روی تمام عناصر `span`ی که دقیقا درون عنصر `p` قرار گرفته‌اند، اعمال می‌گردد:

```
p>span { color:blue; }
```

element+element

Adjacent Sibling Selector

عنصری که بلافاصله بعد از عنصر تعیین شده قرار می‌گیرد، انتخاب می‌شود. در مثال زیر ویژگی `color` بر روی عنصر `span`ی که دقیقا بعد از عنصر `p` قرار گرفته‌اند، اعمال می‌گردد:

```
p+span { color:blue; }
```

CSS Selectors

element~element

General Sibling Selector

عنصری مشخص شده‌ای که بعد از عنصر تعیین شده اول قرار بگیرند، انتخاب می‌شوند (در یک سطح).
در مثال زیر ویژگی `color` بر روی تمام عناصر `span` می‌گردد که بعد از عنصر `p` قرار گرفته‌اند، اعمال می‌گردد:

```
p~span { color:blue; }
```

در این مثال، تمام عناصر `span` که بعد از عناصر `div` و `p` (البته به ترتیب) می‌آیند، انتخاب می‌گردند:

```
div~p~span { color:blue; }
```

```
<div>DIV</div>  
<h1>H1</h1>  
<p>Paragraph</p>  
<h2>H2</h2>  
<span>SPAN</span>
```



DIV
H1
Paragraph
H2
SPAN

CSS Selectors

[attribute]

عنصری که دارای صفت تعیین شده هستند، انتخاب می‌شوند. در مثال زیر ویژگی `color` بر روی تمام عناصر `a` که دارای صفت `target` هستند، اعمال می‌گردد:

```
a[target] { color:blue; }
```

[attribute=value]

عنصری که دارای صفت با مقدار تعیین شده هستند، انتخاب می‌شوند. در مثال زیر ویژگی `color` بر روی تمام عناصر `a` که دارای صفت `target` با مقدار `_blank` هستند، اعمال می‌گردد:

```
a[target=_blank] { color:blue; }
```

CSS Selectors

[attribute~=value]

عنصری که دارای صفت مورد نظر بوده و دقیقا شامل مقدار تعیین شده باشند، انتخاب می‌شوند. در مثال زیر ویژگی border بر روی تمام عناصر که دارای صفت title بوده و در مقدار صفت title آن‌ها دقیقا عبارت book (دو طرف آن فاصله باشد.) وجود داشته باشد، اعمال می‌گردد:

```
[title~=book] { border: 5px solid blue; }
```

[attribute|=value]

عنصری که دارای صفت مورد نظر بوده و دقیقا با مقدار تعیین شده شروع شده باشند، انتخاب می‌شوند. در مثال زیر ویژگی border بر روی تمام عناصر که دارای صفت title بوده و در مقدار صفت title آن‌ها دقیقا عبارت book (دو طرف آن فاصله باشد.) وجود داشته باشد، اعمال می‌گردد:

```
[title|=book] { border: 5px solid blue; }
```

CSS Variables

متغیرهای CSS به DOM دسترسی دارند؛ یعنی می‌توان آن‌ها را به‌صورت سراسری یا محلی تعریف کرد، با جاوا اسکریپت مقدارشان را تغییر داد و حتی بر اساس Media Query (مثل اندازه صفحه) تغییرشان داد.

یکی از بهترین کاربردهای متغیرهای CSS، مدیریت رنگ‌هاست. مثلاً یک رنگ را یک‌بار در قالب یک متغیر تعریف می‌کنیم و همه‌جا از همان استفاده می‌کنیم؛ این کار کد را تمیزتر، قابل‌مدیریت‌تر و انعطاف‌پذیرتر می‌کند.

Media Query: در CSS ابزاری است که به شما اجازه می‌دهد استایل صفحه را بر اساس شرایط دستگاه تغییر دهید.

CSS Variables

متغیرهای CSS با -- شروع می‌شوند و Case Sensitive هستند.
متغیرها را می‌توان به صورت سراسری (Global) یا محلی (Local) تعریف کرد.

■ متغیرهای سراسری:

```
:root {  
  --main-bg-color: blue;  
}
```

- در کل سند در دسترس هستند.

- درون یک Selector با نام root: تعریف می‌شوند (root: ریشه اصلی سند).

```
.container {  
  -- container-bg-color: gray;  
}
```

■ متغیرهای محلی:

- فقط درون همان Selector (و فرزندانش) که تعریف شده، قابل استفاده است.

CSS Variables

تابع `var()`: جهت دسترسی به مقدار متغیرهای CSS

Syntax:

`var(--name, value)`

name: نام متغیر که با `--` شروع می‌شود.

value: مقدار جایگزین (Fallback)، اگر متغیر پیدا نشود. (اختیاری)

```
:root {  
  --main-bg-color: blue;  
}  
  
body {  
  background-color: var(--main-bg-color);  
}
```

CSS Variables

Override کردن متغیر سراسری با متغیر محلی

یک متغیر سراسری را می‌توان درون یک Selector خاص دوباره تعریف کرد. درون آن Selector مقدار محلی استفاده می‌شود و خارج از آن همان مقدار سراسری است. متغیر محلی، متغیر سراسری را فقط در همان محدوده override می‌کند (نه در کل سند).

```
:root {  
  --main-bg-color: blue;  
}  
body {  
  background-color: var(--main-bg-color);  
}
```

```
.container {  
  background-color: var(--main-bg-color);  
}  
.container h2 {  
  --main-bg-color: gray;  
  border: 1px solid var(--main-bg-color);  
}
```

فقط در این Selector رنگ پس زمینه خاکستری است.

CSS Units

واحدهای مطلق (Absolute):

این واحدها زمانی کارآمد هستند که اندازه رسانه خروجی مشخص باشد (مثلا چاپ محتوا بر روی کاغذ). چون ابعاد صفحه‌های نمایش متفاوت است، استفاده از این واحدها برای محتوایی که قرار است بر روی صفحه نمایش نشان داده شود، توصیه نمی‌گردد.

واحدهای نسبی (Relative):

اندازه این واحدها، نسبتی از یک واحد تعیین شده‌ی دیگر است.

CSS Units

Absolute	
cm	centimeters
mm	millimeters
in	inches (1in = 96px)
px	pixels (1px = 1/96 of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12pt)

px در صفحه نمایش، یک واحد نسبی است.

Relative
em
ex
ch
rem
vw
vh
vmin
vmax
%

CSS Units

توصیه نمی‌شود	بعضی موارد	توصیه می‌شود	
pt, cm, mm, in, pc	ex	em, px, %	صفحه نمایش
	px, ex	em, cm, mm, in, pt, pc, %	چاپ

Typeface مجموعه‌ای از طراحی الفبا و شکل حروف است که یک طرح نوشتاری را می‌سازد.
Font یک فایل دیجیتالی، حاوی یک نسخه از Typeface در وزن، اندازه و یا طرح خاص است.
Typeface Family گروهی از Typeface ها هستند، که برای استفاده در کنار یکدیگر طراحی شده‌اند.
مثال:

Arial یک Typeface است.

Arial نرمال با سایز ۱۴ یک فونت و Arial ایتالیک با سایز ۱۲ نیز یک فونت دیگر است.
Arial Typeface Family شامل نسخه Narrow, Bold Italic, Bold, Italic و ... است.

CSS Units

Relative

em:

em یک واحد تایپوگرافی است؛ یک em برابر font-size تعیین شده برای عنصر جاری است (مبنای اندازه در یک Typeface اندازه حرف M بزرگ انگلیسی است). به عنوان مثال اگر Selector ی دارای اندازه فونت 10px باشد، آن گاه:

1em = 10px

2em = 20px

برای تشخیص اندازه em هر Selector، به ویژگی font-size آن مراجعه می کنیم؛ حال اگر این ویژگی برای آن تعیین نشده باشد به Parent (والد) آن مراجعه می کنیم، اگر در والد نیز تعیین نشده بود به والد های بالاتر رجوع می کنیم، در نهایت اگر وجود نداشت به عنصر HTML می رسیم، که به صورت پیش فرض مقدار font-size آن 16px است (اندازه پیش فرض از طریق CSS و یا مرورگر قابل تغییر دادن است).

CSS Units Relative

rem:

rem یا Root EM برابر با اندازه فونت ریشه است؛ یعنی معادل اندازه فونتی که برای HTML تعیین گردیده، که به صورت پیش فرض برابر ۱۶ پیکسل است.

ex:

معادل ارتفاع حرف x کوچک، فونت جاری است (کمتر استفاده می شود).

ch:

معادل عرض صفر (0)، فونت جاری است.

CSS Units

Relative

VW:

Viewport Width

معادل یک درصد از اندازه عرض Viewport است. بنابراین مقدار آن با تغییر اندازه پنجره مرورگر تغییر خواهد کرد.

معادل اندازه پنجره مرورگر است.

vh:

Viewport Height

معادل یک درصد از اندازه ارتفاع Viewport است. بنابراین مقدار آن با تغییر اندازه پنجره مرورگر تغییر خواهد کرد.

CSS Units

Relative

vmin:

معادل مقدار کوچکتر بین vw و vh است.

vmax:

معادل مقدار بزرگتر بین vw و vh است.

%:

بر مبنای اندازه عنصر والد تعیین می گردد.

CSS Comments

اگر بخواهیم توضیحی برای بخش خاصی در کد قرار دهیم و یا جلوی اعمال بخشی از کد را بدون پاک کردن آن بگیریم، از امکان توضیحات (Comments) استفاده می‌کنیم، که با `/*` و با `*/` پایان می‌پذیرد. در واقع مرورگر این بخش از کد را نادیده می‌گیرد:

```
p {  
  color: blue;  
  /* Comment */  
  font-family: tahoma;  
  /* text-align: center;  
    font-size: small; */  
}
```

یک یا چند سطر

CSS Background

از ویژگی‌های background برای افکت‌های پس‌زمینه استفاده می‌گردد:

background: *color image position/size repeat origin clip attachment* initial | inherit;

1. background-color:

برای تعیین رنگ پس‌زمینه عنصر مورد نظر استفاده می‌شود:

background-color: *transparent*;

شفاف (شفافیت ۱۰۰ درصد)

background-color: green;

با استفاده از نام رنگ

background-color: #00FF00;

با استفاده از مقدار هگز روش RGB

background-color: rgb(0, 255, 0);

با استفاده از مقدار دسیمال روش RGB

background-color: rgba(0, 255, 0, 0.3);

با مقدار شفافیت ۳۰ درصد

Alpha: 0.0 – 1.0

background-color: hsl(120, 100%, 50%);

با استفاده از روش HSL

background-color: hsla(120, 100%, 50%, 0.6);

با مقدار شفافیت ۶۰ درصد

HSL Colors

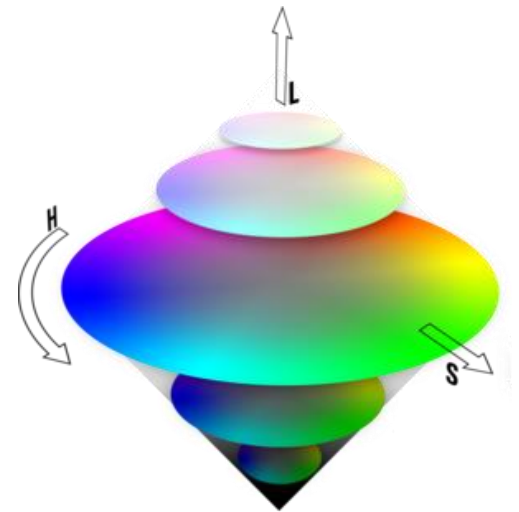
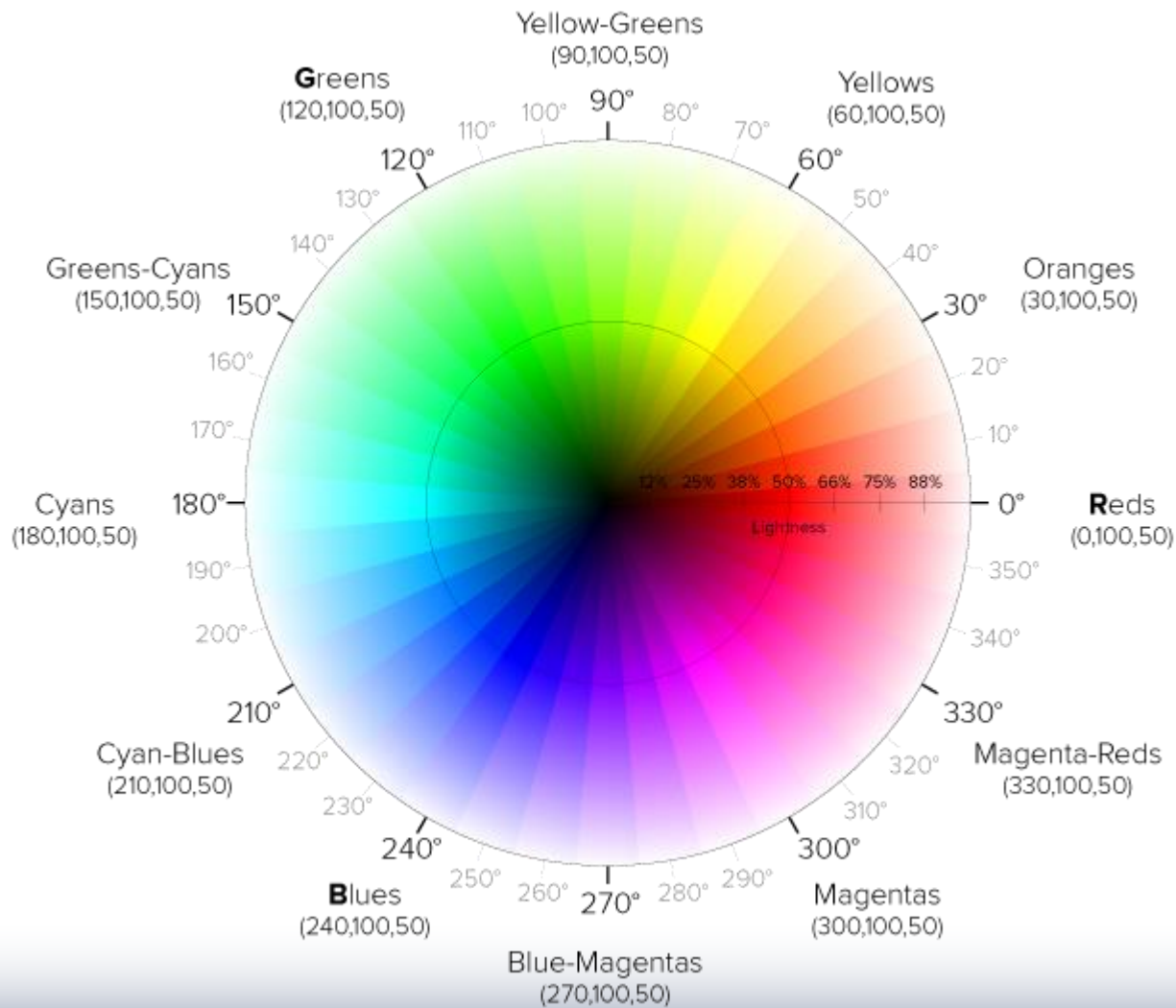
hsl(hue, saturation, lightness)

درجه رنگ در چرخه
بین ۰ تا ۳۶۰ درجه
۰: قرمز
۱۲۰: سبز
۲۴۰: آبی

درصد اشباع رنگ
بین ۰ تا ۱۰۰ درصد
صفر: سایه خاکستری
صد: پر رنگ

درصد روشنی رنگ
بین ۰ تا ۱۰۰ درصد
صفر: مشکی
صد: سفید

HSL Colors



CSS Background

از ویژگی‌های background برای افکت‌های پس‌زمینه استفاده می‌گردد و عبارتند از:

2. `background-image: url|none|initial|inherit;`

برای تعیین عکس پس‌زمینه، عنصر مورد نظر استفاده می‌شود. به صورت پیش‌فرض، عکس در محدوده عنصر تکرار می‌شود، تا کل ناحیه را پوشش دهد.

url: این مقدار، فایل عکس را مشخص می‌کند:

`background-image: url("myImage.jpg");`

none: در صورت استفاده از مقدار none، هیچ پس‌زمینه‌ای نمایش داده نمی‌شود (مقدار پیش‌فرض).

Initial: مقدار این ویژگی را برابر مقدار پیش‌فرض قرار می‌دهد.

Inherit: مقدار این ویژگی، از مقدار والد این عنصر ارث برده می‌شود.

CSS Background

از ویژگی‌های background برای افکت‌های پس‌زمینه استفاده می‌گردد و عبارتند از:

3. background-repeat: repeat | repeat-x | repeat-y | no-repeat | initial | inherit;

نحوه تکرار عکس را مشخص می‌کند:

repeat: عکس در دو محور عمودی و افقی تکرار می‌شود (مقدار پیش فرض).

repeat-x: عکس در محور افقی تکرار می‌شود.

repeat-y: عکس در محور عمودی تکرار می‌شود.

no-repeat: عکس تکرار نمی‌شود.

Initial: مقدار این ویژگی را برابر مقدار پیش‌فرض قرار می‌دهد.

Inherit: مقدار این ویژگی، از مقدار والد این عنصر ارث برده می‌شود.

CSS Background

از ویژگی‌های background برای افکت‌های پس‌زمینه استفاده می‌گردد و عبارتند از:

4. background-attachment: scroll | fixed | local | initial | inherit;

scroll: پس‌زمینه همراه با عناصر صفحه اسکرول می‌شود (مقدار پیش فرض).

fixed: پس‌زمینه ثابت می‌شود.

local: پس‌زمینه همراه با محتوای عنصر اسکرول می‌شود.

Initial: مقدار این ویژگی را برابر مقدار پیش‌فرض قرار می‌دهد.

Inherit: مقدار این ویژگی، از مقدار والد این عنصر ارث برده می‌شود.

CSS Background

از ویژگی‌های background برای افکتهای پس‌زمینه استفاده می‌گردد و عبارتند از:

5. background-position

مکان شروع پس‌زمینه را مشخص می‌کند، به صورت پیش فرض از گوشه بالا، سمت چپ شروع می‌شود.

مقادیر:

left top
left center
left bottom
right top
right center
right bottom
center top
center center
center bottom

اگر یک مقدار
تعیین شود، مقدار
Center بعدی
در نظر گرفته
می‌شود.

$x\% y\%$

$xpos ypos$

Initial

inherit

اگر یک مقدار تعیین شود، مقدار
بعدی 50% در نظر گرفته می‌شود.

CSS Text

Text Alignment:

از این ویژگی برای تنظیم چیدمان افقی متن استفاده می‌شود.

`text-align: left | right | center | justify | initial | inherit;`

مقدار پیش‌فرض، در صورتی که ویژگی `direction` برابر `ltr` باشد: `left`

مقدار پیش‌فرض، در صورتی که ویژگی `direction` برابر `rtl` باشد: `right`

Text Decoration:

از این ویژگی برای تنظیم Decoration متن استفاده می‌شود.

`text-decoration: none | underline | overline | line-through | initial | inherit;`

مقدار پیش‌فرض

CSS Text

Text Color:

از این ویژگی برای تنظیم رنگ متن استفاده می‌شود.

رنگ متن نیز مانند رنگ پس‌زمینه با یکی از روش‌های ذکر شده تعیین می‌شود، مانند:

```
color: blue;
```

```
color: #0000FF;
```

برای سازگاری با W3C، اگر ویژگی `color` تعیین شود، ویژگی `background-color` نیز باید تعریف شود.

Text Direction:

از این ویژگی برای تعیین جهت متن استفاده می‌گردد:

```
direction: ltr | rtl | initial | inherit;
```

ltr: Left to Right (مقدار پیش‌فرض)
rtl: Right to Left

CSS Text

Text Transformation:

از این ویژگی برای بزرگ و کوچک کردن حروف متن استفاده می‌شود.

text-transform: none | capitalize | uppercase | lowercase | initial | inherit;

حرف اول هر کلمه: بزرگ

حروف بزرگ

حروف کوچک

Text Indentation:

از این ویژگی برای ایجاد فرورفتگی در اولین سطر متن استفاده می‌شود.

text-indent: *length* | initial | inherit;

مقدار پیش‌فرض صفر است.

این مقدار می‌تواند منفی باشد. همچنین می‌تواند درصدی از عرض عنصر والد باشد، مانند:

text-indent: 10%;

CSS Text

Letter Spacing:

از این ویژگی برای تعیین فاصله بین حروف متن استفاده می شود.

letter-spacing: normal | *length* | initial | inherit;

مقدار پیش فرض

Line Height:

از این ویژگی برای تعیین فاصله بین سطرها استفاده می شود.

line-height: normal | *number* | *length* | initial | inherit;

مقدار پیش فرض

این مقدار نمی تواند منفی باشد.

Word Spacing:

از این ویژگی برای تعیین فاصله بین کلمات استفاده می شود.

word-spacing: normal | *length* | initial | inherit;

مقدار پیش فرض

این مقدار می تواند منفی باشد.



Cascading Style Sheets

Accent Color:

از این ویژگی برای تعیین رنگ کنترل‌هایی مانند checkbox استفاده می‌شود.

accent-color: auto | *color* | initial | inherit;

```
<style>
  input[type=checkbox] { accent-color: orange; }
</style>
```

```
<input type="checkbox" name="webskills" value="HTML" checked>HTML
<input type="checkbox" name="webskills" value="CSS" checked>CSS
<input type="checkbox" name="webskills" value="JavaScript">JavaScript
```

☒ HTML ☒ CSS ☐ JavaScript

CSS Link

لینک‌ها می‌توانند در حالت‌های مختلف، استایل‌های متفاوتی داشته باشند.
حالت‌های مختلف لینک‌ها عبارتند از:

a:link	حالت نرمال (لینک بازدید نشده)
a:visited	لینک بازدید شده
a:hover	هنگام آمدن اشاره‌گر موس بر روی لینک
a:active	هنگام کلیک بر روی لینک

نکته:

- a:hover باید بعد از a:link و a:visited بیاید.
- a:active باید بعد از a:hover بیاید.

سوال: کد `a.myclass:hover {text-decoration:underline;}` بر روی چه لینکی تاثیر می‌گذارد؟

`<a class="myclass" href="index.html"> TEST</a>`

CSS Link

حالت‌های مختلف لینک، به صورت پیش فرض:

حالت نرمال

```
a:link {color:blue;
        background-color:transparent;
        text-decoration:underline}
```

بعد از کلیک لینک

```
a:visited {color:Purple;
            background-color:transparent;
            text-decoration:underline}
```

هنگام رفتن موس
بر روی لینک

```
a:hover {color:blue;
          background-color:transparent;
          text-decoration:underline}
```

هنگام کلیک موس
بر روی لینک

```
a:active {color:red;
           background-color:transparent;
           text-decoration:underline}
```

CSS Font

با استفاده از ویژگی‌های **font**، می‌توان خانواده، اندازه و استایل فونت متن را تعیین نمود:

Font Family: با استفاده از این ویژگی می‌توان چندین فونت تعیین نمود، تا به ترتیب در صورت عدم پشتیبانی مرورگر از هر یک، از فونت بعدی استفاده شود:

font-family: "Times New Roman", Times, serif;

Font Style: با استفاده از این ویژگی می‌توان یکی از حالت‌های **italic**، **normal** و یا **oblique** را به

متن داد:

شبه **italic** است، اما کمتر پشتیبانی می‌شود.

font-style: italic;

CSS Font

Font Size: این ویژگی اندازه فونت را تعیین می کند:

`font-size: 16px;`

`font-size: 1em;`

1em، معادل اندازه فونت جاری است؛ در مرورگرها اندازه پیش فرض فونت ۱۶ پیکسل است، در نتیجه اندازه پیش فرض 1em نیز برابر ۱۶ پیکسل است. W3C استفاده از واحد em را توصیه می کند. برای تبدیل واحد px به em از فرمول ساده زیر استفاده می کنیم:

$$em = pixels/16$$

همچنین می توان اندازه فونت را با توجه به مقدار پیش فرض به صورت درصد تعیین نمود:

`font-size: 100%;`

CSS Font

Font Weight: این ویژگی وزن فونت را تعیین می کند:

font-weight: normal | bold | initial | inherit;

Font Variant: از این ویژگی برای تعیین وزن حالت small-caps فونت استفاده می شود، در این حالت حروف کوچک به صورت بزرگ، اما با اندازه ای کوچکتر از حروف بزرگ موجود در متن نمایش داده می شوند:

font-variant: normal | small-caps | initial | inherit;

– تمام ویژگی های ذکر شده را می توان در یک سطر بیان نمود:

font: font-style font-variant font-weight font-size font-family;

font: font-style font-variant font-weight font-size/line-height font-family | caption | icon | menu | message-box | small-caption | status-bar | initial | inherit;

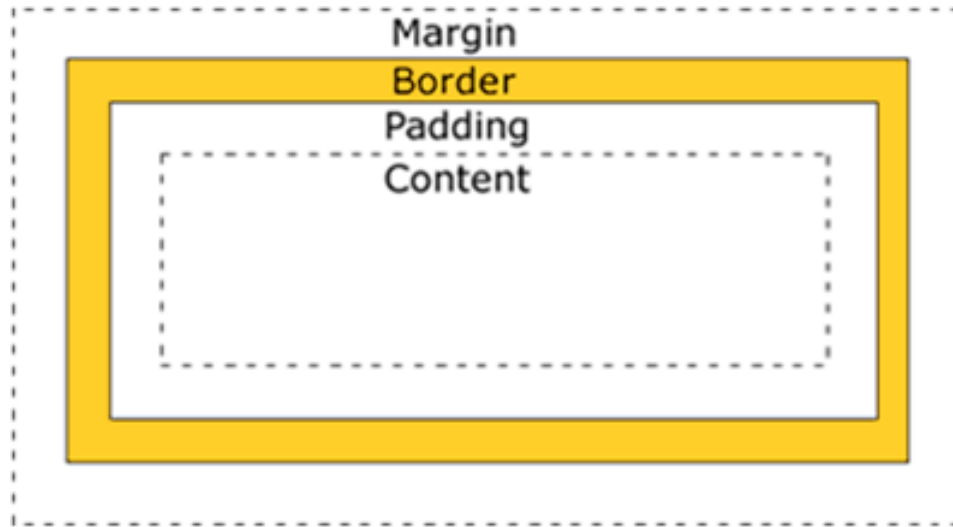
CSS Web Font

قبل از CSS3 طراحان وب فقط می‌توانستند از فونت‌هایی که بر روی سیستم کاربر نصب شده بود استفاده کنند (این فونت‌های برای زبان فارسی (مانند Arial و Tahoma) بسیار محدود هستند).

اما در CSS3، برای استفاده از فونت‌های مورد نظری که بر روی سیستم کاربر نصب نشده‌اند، می‌توان از **فونت‌های وب** استفاده نمود. برای این کار کافی است فایل فونت مورد نظر را، در وب سرور قرار دهیم؛ در صورت نیاز، فایل فونت به‌صورت خودکار دانلود خواهد شد. این فونت‌ها با استفاده از قاعده **@font-face** تعیین می‌شوند:

```
@font-face {  
    font-family: myFontName;  
    src: url(fontName.woff);  
}  
  
div {  
    font-family: myFontName;  
}
```

CSS Box Model



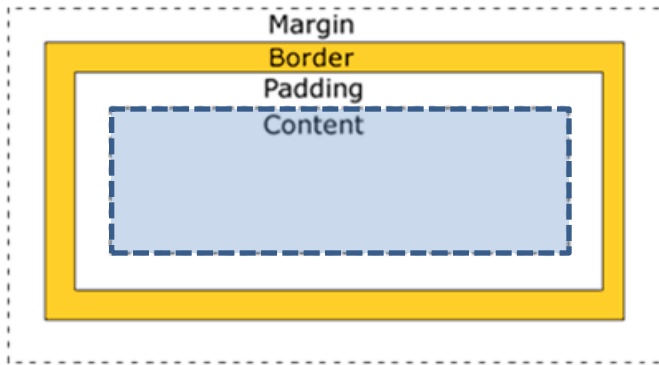
همه عناصر HTML می‌توانند به عنوان یک Box در نظر گرفته شوند.

در شکل بالا، مدل Box نشان داده شده است:

- **Content**: محتوای Box، جایی است که متن و تصویر در آن قرار می‌گیرند.
- **Padding**: حاشیه‌ی دور Content بوده و شفاف است.
- **Border**: در حاشیه‌ی Padding و Content قرار دارد.
- **Margin**: حاشیه‌ی دور Border بوده و شفاف است.

CSS Dimensions

از ویژگی‌های ابعاد، برای کنترل عرض و ارتفاع محدوده Content عناصر استفاده می‌شود:



• height: ارتفاع عنصر را تعیین می‌کند.

height: auto | length | initial | inherit;

مقدار آن توسط مرورگر محاسبه می‌گردد.

• width: عرض عنصر را تعیین می‌کند.

width: auto | length | initial | inherit;

مقدار آن یا با یکی از واحدهای px، cm و مانند آن‌ها و یا با درصد (درصدی از بلوکی که عنصر در آن قرار دارد). تعیین می‌گردد.

CSS Dimensions

• max-height: حداکثر ارتفاع عنصر را تعیین می کند.

max-height: none | length | initial | inherit;

ویژگی height را بی تاثیر می کند.

مقدار پیش فرض

• min-height: حداقل ارتفاع عنصر را تعیین می کند.

min-height: length | initial | inherit;

مقدار پیش فرض: *

• max-width: حداکثر عرض عنصر را تعیین می کند.

max-width: none | length | initial | inherit;

مقدار پیش فرض

• min-width: حداقل عرض عنصر را تعیین می کند.

min-width: length | initial | inherit;

مقدار پیش فرض: *

ویژگی های width و max-width را بی تاثیر می کند.

CSS Padding

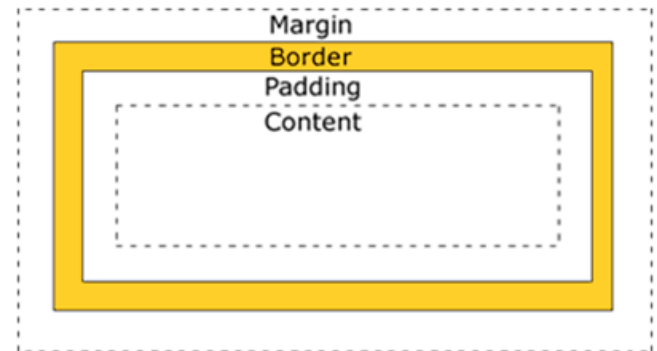
Padding: حاشیه‌ی دور Content را مشخص می‌کند، به عبارتی فاصله بین Content و Border است.

این حاشیه شفاف بوده و از رنگ پس‌زمینه تاثیر می‌پذیرد. ویژگی‌های CSS جهت تعیین Padding:

- padding
- padding-top
- padding-right
- padding-bottom
- padding-left

padding: AllPaddings;
padding: top&bottom right&left;
padding: top right&left bottom;
padding: top right bottom left;

1 2
1 2 3
1 2 3 4



مقدار این ویژگی‌ها:

length: مقدار عددی یکی از واحدهای CSS مانند px، cm و ...

%: درصدی از عرض عنصر حاوی آن

initial: مقدار پیش‌فرض (0)

inherit: ارث‌بری از عنصر والد

CSS Padding

- padding: 10px;
 - padding-top: 10px;
 - padding-right: 10px;
 - padding-bottom: 10px;
 - padding-left: 10px;
- padding: 10px 20px;
 - padding-top: 10px;
 - padding-right: 20px;
 - padding-bottom: 10px;
 - padding-left: 20px;
- padding: 10px 20px 30px;
 - padding-top: 10px;
 - padding-right: 20px;
 - padding-bottom: 30px;
 - padding-left: 20px;
- padding: 10px 20px 30px 40px;
 - padding-top: 10px;
 - padding-right: 20px;
 - padding-bottom: 30px;
 - padding-left: 40px;

padding: AllPaddings;

1

padding: top&bottom right&left;

1

2

padding: top right&left bottom;

1

2

3

padding: top right bottom left;

1

2

3

4

این روش برای تعیین مقادیر بالا، راست، پایین و چپ عنصر، در سایر موارد مشابه نیز کاربرد دارد.

۱ مقدار: تعیین مقدار چهار جهت

۲ مقدار: تعیین مقادیر بالا-پایین و چپ-راست

۳ مقدار: تعیین مقادیر بالا، چپ-راست و پایین

۴ مقدار: تعیین مقادیر بالا، راست، پایین و چپ (موافق عقربه‌های ساعت)

CSS Border

border-style

مقدار پیش فرض

none

hidden

dotted

dashed

solid

double

groove

ridge

inset

outset

با استفاده از این صفت می‌توان سبک، اندازه و رنگ ناحیه بین padding و margin عناصر را تعیین کرد. تا زمانی که سبک تعیین نشود، تعیین اندازه و رنگ تاثیری ندارد.

```
<style>
p {
  border-width: 1px;
  border-style: solid;
  border-color: black;
}
</style>
```

border-X: AllBorders;
border-X: top&bottom right&left;
border-X: top right&left bottom;
border-X: top right bottom left;

```
<style>
p {
  border: 1px solid black;
}
</style>
```

خلاصه نویسی

border: width style color;

CSS Border

ویژگی‌های هر جهت **border** را می‌توان به صورت مجزا نیز تعیین نمود:

- border-top
- border-right
- border-bottom
- border-left
- border-top-style
- border-right-style
- border-bottom-style
- border-left-style
- border-top-width
- border-right-width
- border-bottom-width
- border-left-width
- border-top-color
- border-right-color
- border-bottom-color
- border-left-color

علاوه بر مقادیر عددی، می‌تواند ویژگی **width** را معادل یکی از مقادیر **thin**، **medium** یا **thick** قرار داد.

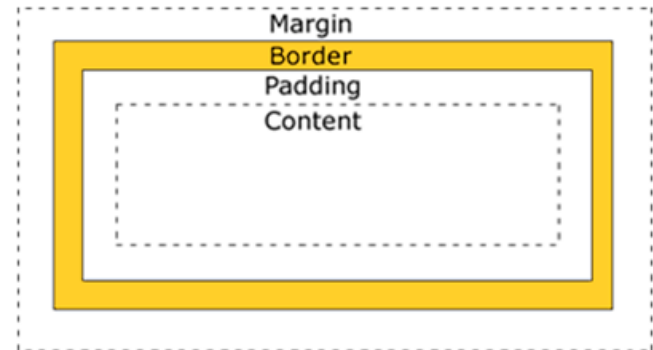
CSS Margin

Margin: حاشیه‌ی دور عناصر را مشخص می‌کند، به عبارتی اندازه فضای بعد از Border است.

این حاشیه شفاف بوده و نمی‌تواند رنگ پس‌زمینه داشته باشد. ویژگی‌های CSS جهت تعیین Margin:

- margin
- margin-top
- margin-right
- margin-bottom
- margin-left

margin: AllMargins;
1
margin: top&bottom right&left;
1 2
margin: top right&left bottom;
1 2 3
margin: top right bottom left;
1 2 3 4



مقدار این ویژگی‌ها:

length: مقدار عددی یکی از واحدهای CSS مانند px، cm و ...

%: درصدی از عرض عنصر حاوی آن

auto: توسط مرورگر محاسبه می‌شود (وسط چین: Margin-left=Margin-right).

inherit: ارث‌بری از عنصر والد

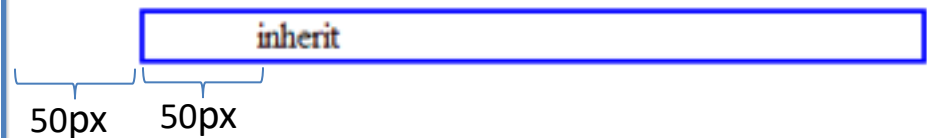
می‌تواند منفی باشد.

CSS Margin

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div.class1 {
        border: 2px solid blue;
        margin-left: 50px;
      }
      span.class2 {
        margin-left: inherit;
      }
    </style>
  </head>
  <body>
    <h1>CSS inherit Value</h1>
    <div class="class1">
      <span class="class2">inherit</span>
    </div>
  </body>
</html>
```

در اینجا برای عنصر `div` کلاسی به نام `class1` تعریف و `margin` چپ آن برابر ۵۰ پیکسل تعیین شده است. برای عنصر `span` نیز کلاسی به نام `class2` تعریف و مقدار `margin` چپ آن برابر `inherit` قرار داده شده است؛ یعنی این کلاس از `parent` خود ارث‌بری می‌کند. در بخش `body`، عنصر `span` با کلاس `class2` دورن عنصر `div` با کلاس `class1` قرار گرفته؛ به عبارتی در اینجا عنصر `div` والد عنصر `span` است، در نتیجه مقدار `margin-left` عنصر `span` نیز برابر ۵۰ پیکسل شده است.

CSS inherit Value

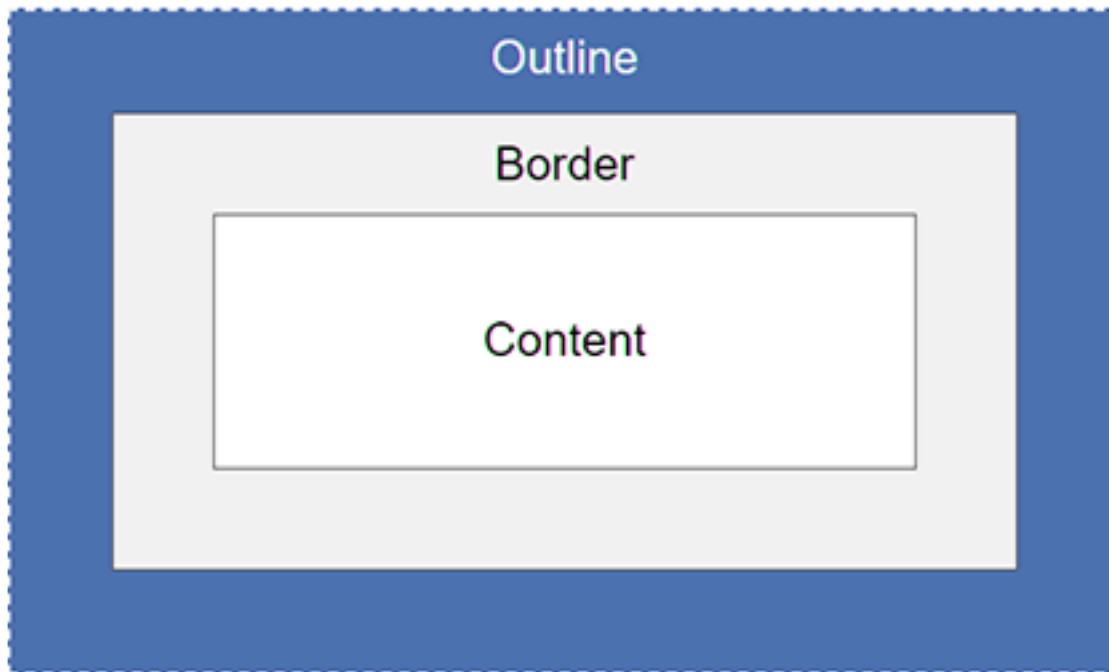


CSS Outline

outline، خطی است که دور یک عنصر (خارج از border)، رسم می‌شود.

ویژگی‌ها:

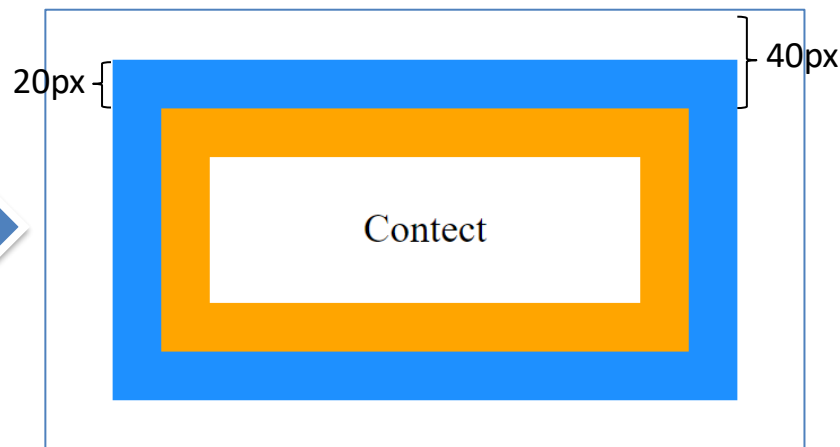
- outline-style
- outline-color
- outline-width
- outline-offset
- outline



CSS Outline

outline، خطی است که دور یک عنصر (خارج از border)، رسم می‌شود.

```
<html>
<head>
<style>
  p {
    padding: 20px;
    border: 20px solid orange;
    margin: 40px;
    outline: DodgerBlue solid 20px;
    text-align: center;
  }
</style>
</head>
<body>
  <p>Contect</p>
</body>
</html>
```



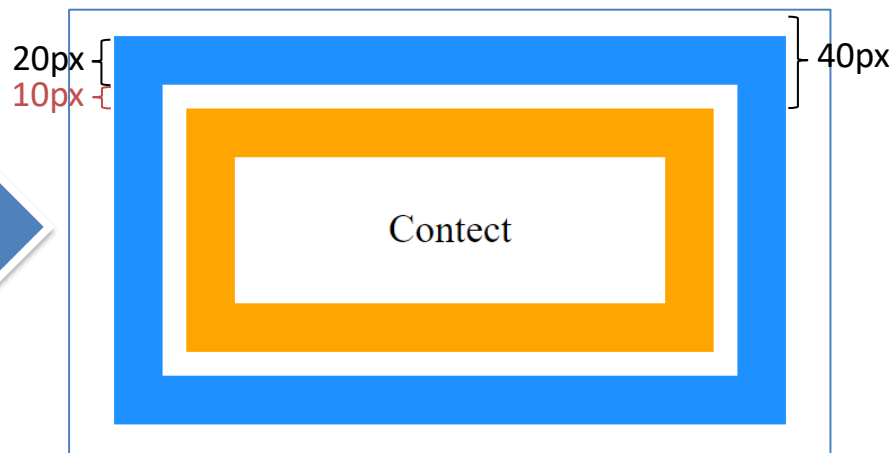
CSS Outline

outline، خطی است که دور یک عنصر (خارج از border)، رسم می‌شود.

outline-offset: بین border و outline عنصر،

فاصله (شفاف) ایجاد می‌کند.

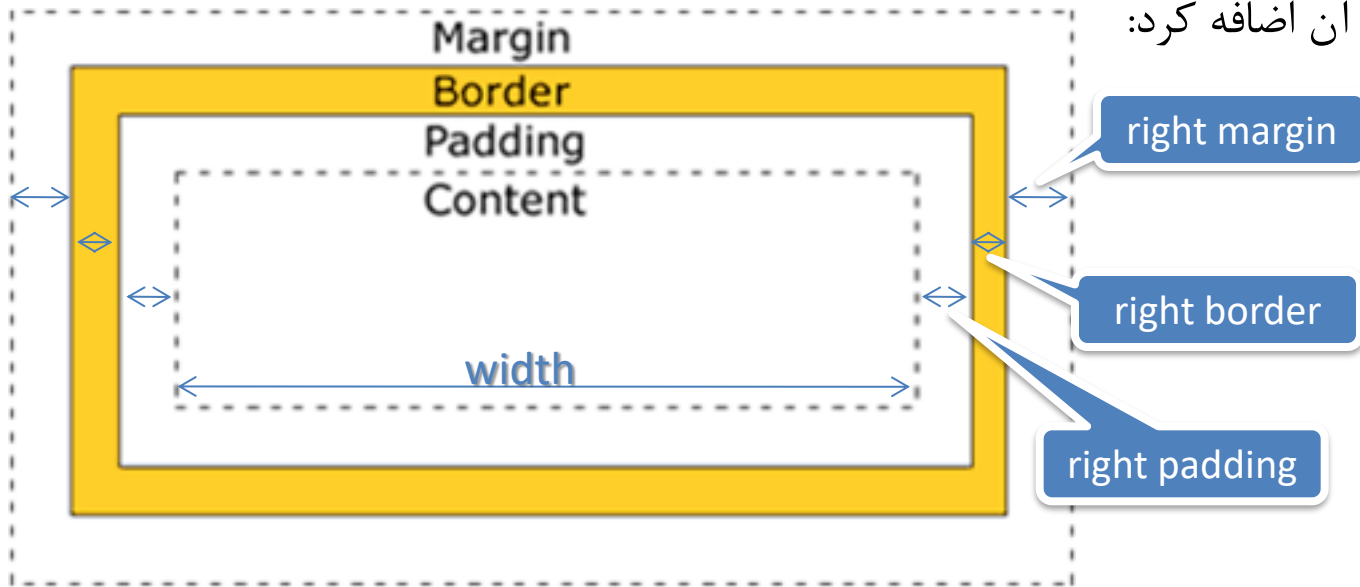
```
<html>
<head>
  <style>
    p {
      padding: 20px;
      border: 20px solid orange;
      margin: 40px;
      outline: DodgerBlue solid 20px;
      outline-offset: 10px;
      text-align: center;
    }
  </style>
</head>
<body>
  <p>Contect</p>
</body>
</html>
```



CSS Box Model

ویژگی‌های `width` و `height` هر عنصر در CSS، عرض و ارتفاع محدوده محتوا (Content) را مشخص می‌کنند. در نتیجه برای تعیین اندازه کامل یک عنصر، باید مقادیر `Padding`، `Border` و `Margin` را نیز

در نظر گرفت و به آن اضافه کرد:



Total element width = width + left padding + right padding + left border + right border + left margin + right margin

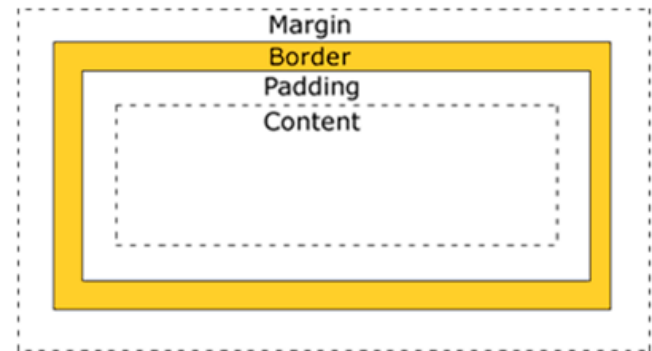
Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

CSS Box Model

به عنوان مثال، عرض Box، در عنصر div زیر ۳۰۰ پیکسل است:

```
div {  
  width: 280px;  
  padding: 5px;  
  border: 5px solid black;  
  margin: 0;  
}
```

280px (width) +
10px (left + right padding) +
10px (left + right border) +
0px (left + right margin)
= 300px



Total element width = width + left padding + right padding + left border + right border + left margin + right margin

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin