

# کارگاه طراحی صفحات وب

هدف: طراحی وب سایت‌های ایستا

CSS

نسخه: ۱.۳

kishuniv@montajab.com

<http://montajab.com/kishuniv>

# 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 آبهاری 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، پیش‌فرض 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 Selectorها از Combinatorها استفاده می‌کنیم و عبارتند از:

### 1. Child Selector (>):

انتخاب تمامی عناصر مرتبگی که بی‌واسطه، فرزند عنصر والد تعیین شده هستند.

### 2. Descendant Selector (فاصله):

انتخاب تمامی عناصر مرتبگی که از نوادگان (زیرشاخه‌های) 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 مرتبگی 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` 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` 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` 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]

عناصری 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]

عناصری 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]

عناصری 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]

عناصری 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; }
```