

# CSS 튜토리얼

CSS 는 HTML 문서의 스타일을 설명하는 언어입니다.

CSS 는 HTML 요소를 표시하는 방법을 설명합니다.

```
<html>
<head>
<style>
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>
<body>

<h1>My First CSS Example</h1>
<h1>This is a paragraph.</h1>

</body>
</html>
```

## CSS 소개

### CSS 란 무엇입니까?

- **CSS** 는 의미 **Cascading Style Sheets**
- CSS 는 **HTML** 요소가 화면, 종이 또는 다른 매체에 표시되는 방법을 설명 합니다
- CSS 는 많은 작업을 저장합니다. 여러 웹 페이지의 레이아웃을 한 번에 제어 할 수 있습니다.
- 외부 스타일 시트는 **CSS 파일**에 저장됩니다.

# CSS 데모-하나의 HTML 페이지-다양한 스타일!

여기서 우리는 네 개의 다른 스타일 시트로 표시되는 하나의 HTML 페이지를 보여줍니다. 다른 스타일을 보려면 아래의 "스타일 시트 1", "스타일 시트 2", "스타일 시트 3", "스타일 시트 4"링크를 클릭하십시오.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>데모</title>
</head>

<body>
  <div id="wrap">
    <div class="header">
      <h1>Welcome to My Homepage</h1>
      <p>Use the menu to select different Stylesheets</p>
    </div>

    <div class="section clearfix">
      <ul>
        <li><a href="stylesheet_1.html">Stylesheet 1</a></li>
        <li><a href="stylesheet_2.html">Stylesheet 2</a></li>
        <li><a href="stylesheet_3.html">Stylesheet 3</a></li>
        <li><a href="stylesheet_4.html">Stylesheet 4</a></li>
        <li><a href="no_stylesheet.html">No Stylesheet</a></li>
      </ul>

      <div class="content">
        <h1>Same Page Different Stylesheets</h1>
        <p>This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:<br>
        <a href="stylesheet_1.html">Stylesheet 1</a>,</p>
      </div>
    </div>
  </div>
</body>
</html>
```

```

        <a href="stylesheet_2.html">Stylesheet 2</a>,
        <a href="stylesheet_3.html">Stylesheet 3</a>,
        <a href="stylesheet_4.html">Stylesheet 4</a>.
    </p>
    <h2>No Styles</h2>
    <p>This page uses DIV elements to group different sections of the
HTML page. Click here to see how the page
    looks
    like with no stylesheet:<br><a href="no_stylesheet.html">No Sty
lesheet</a>.</p>

    </div>

    <div class="aside">
        <h3>Side-Bar</h3>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed d
iam nonummy nibh euismod tincidunt ut
        laoreet
        dolore magna aliquam erat volutpat.</p>
    </div>
</div>

<div class="footer">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet
        dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam,
quis nostrud exerci tation ullamcorper
        suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis au
tem vel eum iriure dolor in hendrerit in
        vulputate velit esse molestie consequat, vel illum dolore eu feugi
at nulla facilisis at vero eros et
        accumsan
        et iusto odio dignissim qui blandit praesent luptatum zzril delenit
augue duis dolore te feugait nulla
        facilisi.</p>
    </div>
</div>
</body>

</html>

```

# 스타일 시트 1

```

* {box-sizing: border-box;}
a {text-decoration: none; color: #000;}

```

```

body {background:#ccc;}
#wrap {width: 1050px; background:#fff; padding:20px;}

div.header {background: #4caf50; border-
radius: 5px; padding: 10px 20px 20px 20px;}
div.header h1, div.header p {margin: 0; color: #fff;}

div.section {margin: 5px 0; height: 380px;}

ul {margin: 0; padding: 0; float: left; width: 20%;}
ul li {list-style: none; background: #f1f1f1; border-radius: 5px; margin-
bottom: 5px; padding: 5px; border: 1px solid #ccc;}
ul li a {display:block;}
ul li:first-of-type {background: #4caf50;}
ul li:hover {background: #fff;}
ul li:first-of-type:hover {background: #4caf50;}
ul li:first-of-type a {color: #fff; font-weight: bold;}

div.content {width: 60%; float: left; padding: 20px; line-height:1.6;}
div.content h1 {margin:0;}
div.content a {text-decoration: underline;}
div.content a:hover {text-decoration: none;}

div.aside {background: #32a4e7; width: 20%; float: right; border-
radius: 5px; color: #fff; padding: 10px; height: 100%; line-height:1.6;}

div.footer {background: #f1f1f1; border-radius: 5px; font-size: 0.7em; text-
align: center; padding: 2px 10px;}

```

## 스타일 시트 2

```

* {box-sizing: border-box;}
a {text-decoration: none; color: #000;}
body {font-size:13px; background:#ccc;}
#wrap {width: 1050px; background:#d14836; padding:10px;}

div.header {border-radius: 5px; padding: 10px 20px 20px 20px; font-size:2em;}
div.header h1, div.header p {margin: 0; color: #fff;}

div.section {background:#fff; margin:5px 0; color:#d14836;}

ul {margin:20px 10px 0 0; padding: 0; float: right; width: 16%; font-
size:1.2em;}
ul li {list-style: none; background:#d14836; border-radius: 20px; margin-
bottom: 5px; padding: 5px; border: 1px solid #ccc; text-align:center;}

```

```

ul li:hover {background: #fff;}
ul li:hover a {color: #d14836;}
ul li a {display:block; color:#fff;}
ul li:nth-of-type(2) {background: #fff; font-weight: bold;}
ul li:nth-of-type(2) a {color:#d14836;}

div.content {width: 83%; float: left; padding:20px 10px; line-height:1.6;}
div.content p {color:#000;}
div.content h1 {margin:0;}
div.content a {color:#d14836;}
div.content a:hover {text-decoration: underline;}

div.aside {clear:both; background:#f6dad7; color: #fff; padding: 10px; line-
height:1.6; color:#d14836; font-size:1.1em;}

div.footer {font-size: 0.7em; text-
align: center; padding: 2px 10px; color:#fff; line-height:2;}

```

## 스타일 시트 3

```

* {box-sizing: border-box;}
a {text-decoration: none; color: #000;}
body {font-size:17px; background:#ccc;}
#wrap {width: 1080px; padding:20px; background:#fff;}

div.header {border-radius: 5px;}
h1,h2,h3 {margin: 0; color: #4caf50;}

ul {margin:40px 0; padding: 0;}
ul li {float:left; width:15%; text-align:center; list-
style: none; background: #555; margin-bottom: 5px; padding: 5px; margin-
right:10px; border: 1px solid #ccc;}
ul li a {display:block; color:#fff;}
ul li:nth-of-type(3) {background:#4caf50;}
ul li:hover {background:#999;}

div.content {clear:both; line-height:1.6;}
div.content h1 {margin:0;}
div.content a {text-decoration: underline;}
div.content a:hover {color:#84c754;}

div.aside {border:1px solid #ccc; padding: 10px; height: 100%; line-
height:1.6; background: #f1f1f1;}

div.footer {border-radius: 5px; font-size: 0.7em; text-

```

```
align: center; padding: 2px 10px;}
```

## 스타일 시트 4

```
* {box-sizing: border-box;}
a {text-decoration: none; color: #000;}
body {background:#ccc; color:#84c754; font-family:'Courier New', Courier, monospace;}
#wrap {position:relative; width: 1080px; padding:20px; background:#000;}

div.header {border-radius: 5px;}
div.header p {margin:0;}
h1,h2 {margin: 0; color: #84c754;}

ul {padding-left:30px; margin:30px 0; line-height:1.8;}
ul li {margin-right:10px;}
ul li a {display:block; color:#fff;}
ul li a {color:#84c745;}
ul li:nth-of-type(4) a {color:#fff;}
ul li:hover a {color:#fff;}

div.content {clear:both; line-height:1.6;}
div.content h1 {margin:0;}
div.content a {text-decoration: underline; color:#fff;}
div.content a:hover {color:#84c754;}

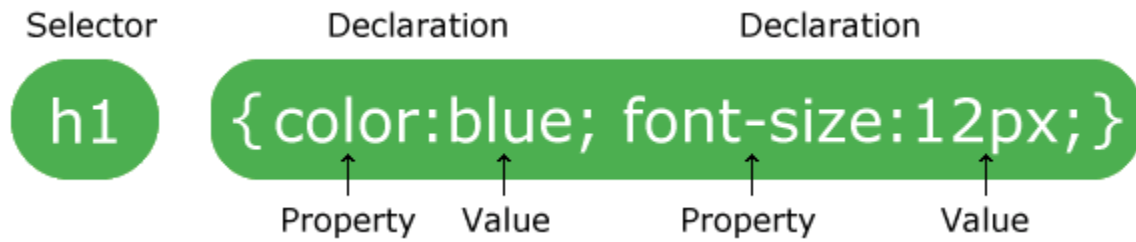
div.aside {position:absolute; top:100px; right:20px; border:1px solid #ccc; padding: 10px; line-height:1.6; color:#fff; border:1px solid #fff; border-radius:5px; font-size:0.8em; width:32%;}

div.footer {border-radius: 5px; font-size: 0.7em; text-align: center; padding: 2px 10px; color:#fff; border:1px solid #fff; border-radius:5px;}
```

## CSS 구문

## CSS 구문

CSS 규칙 세트는 선택기와 선언 블록으로 구성됩니다.



선택기는 스타일을 지정할 HTML 요소를 가리 킵니다.

선언 블록에는 세미콜론으로 구분 된 하나 이상의 선언이 포함됩니다.

각 선언에는 CSS 속성 이름과 값이 콜론으로 구분되어 포함됩니다.

CSS 선언은 항상 세미콜론으로 끝나고 선언 블록은 중괄호로 묶습니다.

```
<html>
<head>
<style>
p {
  color: red;
  text-align: center;
}
</style>
</head>
<body>

<p>Hello World!</p>
<p>These paragraphs are styled with CSS.</p>

</body>
</html>
```

## CSS 주석

주석은 코드를 설명하는 데 사용되며 나중에 소스 코드를 편집 할 때 도움이 될 수 있습니다.

브라우저에서는 주석을 무시합니다.

```
<html>
<head>
<style>
p {
  color: red;
  /* This is a single-line comment */
  text-align: center;
}

/* This is
```

```
a multi-line
comment */
</style>
</head>
<body>

<p>Hello World!</p>
<p>These paragraphs are styled with CSS.</p>

</body>
</html>
```

CSS 주석은 / \*로 시작하고 \*/로 끝납니다. 주석은 여러 줄에 걸쳐있을 수도 있습니다.

## CSS 선택기

## CSS 선택기

CSS 선택기는 스타일을 지정할 HTML 요소를 "찾기"(또는 선택)하는 데 사용됩니다.

CSS 선택기를 5 가지 범주로 나눌 수 있습니다.

- 간단한 선택기 (이름, ID, 클래스를 기준으로 요소 선택)
- [조합기 선택기](#) (그들 간의 특정 관계에 따라 요소를 선택)
- [의사 클래스 선택기](#) (특정 상태에 따라 요소 선택)
- [의사 요소 선택기](#) ( [요소](#) 의 일부를 선택하고 스타일 지정)
- [속성 선택기](#) (특성 또는 속성 값을 기반으로 요소 선택)

## CSS 요소 선택기

요소 선택기는 요소 이름을 기준으로 HTML 요소를 선택합니다.

```
<html>
<head>
<style>
p {
  text-align: center;
  color: red;
}
```



```

</style>
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>

```

여기에서 페이지의 모든 <p> 요소는 빨간색 텍스트 색상으로 가운데 정렬됩니다.

## CSS id 선택기

id 선택기는 HTML 요소의 id 속성을 사용하여 특정 요소를 선택합니다.

요소의 id 는 페이지 내에서 고유하므로 id 선택기는 하나의 고유 한 요소를 선택하는 데 사용됩니다!

특정 ID 를 가진 요소를 선택하려면 해시 (#) 문자 다음에 요소의 ID 를 씁니다.

```

<html>
<head>
<style>
#para1 {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>

</body>
</html>

```

CSS 규칙은 id = "para1"인 HTML 요소에 적용됩니다.

**참고 :** 아이디 이름은 숫자로 시작할 수 없습니다!

# CSS 클래스 선택기

클래스 선택기는 특정 클래스 속성을 가진 HTML 요소를 선택합니다.

특정 클래스를 가진 요소를 선택하려면 마침표 (.) 문자와 클래스 이름을 차례로 쓰십시오.

```
<html>
<head>
<style>
.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center"> Red and center-aligned paragraph.</p>

</body>
</html>
```

class = "center"인 모든 HTML 요소는 빨간색과 가운데 정렬됩니다.

특정 HTML 요소 만 클래스의 영향을 받도록 지정할 수도 있습니다.

```
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>

</body>
</html>
```

class = "center"인 <p> 요소 만 가운데 정렬됩니다.

HTML 요소는 둘 이상의 클래스를 참조 할 수도 있습니다.

```
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red;
}
p.large {
  font-size: 300%;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
<p class="center large">This paragraph will be red, center-aligned, and in a
large font-size.</p>

</body>
</html>
```

<p> 요소는 class = "center" 및 class = "large"에 따라 스타일이 지정됩니다.

**참고 :** 클래스 이름은 숫자로 시작할 수 없습니다!

## CSS 유니버설 셀렉터

범용 선택기 (\*)는 페이지에서 모든 HTML 요소를 선택합니다.

```
<html>
<head>
<style>
* {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>

<h1>Hello world!</h1>
<p>Every element on the page will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

CSS 규칙은 페이지의 모든 HTML 요소에 영향을 줍니다.

## CSS 그룹화 선택기

그룹화 선택기는 동일한 스타일 정의를 가진 모든 HTML 요소를 선택합니다.

다음 CSS 코드를 보십시오 (h1, h2 및 p 요소의 스타일 정의는 동일 함).

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

코드를 최소화하기 위해 선택기를 그룹화하는 것이 좋습니다.

선택기를 그룹화하려면 각 선택기를 쉼표로 구분하십시오.

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

## 모든 CSS 단순 선택기

Selector	Example	Example description
<a href="#">.class</a>	.intro	Selects all elements with class="intro"
<a href="#">#id</a>	#firstname	Selects the element with id="firstname"
<a href="#">*</a>	*	Selects all elements
<a href="#">element</a>	p	Selects all <p> elements
<a href="#">element,element,..</a>	div, p	Selects all <div> elements and all <p> elements

# CSS 를 추가하는 방법

브라우저는 스타일 시트를 읽을 때 스타일 시트의 정보에 따라 HTML 문서를 형식화합니다.

## CSS를 삽입하는 세 가지 방법

스타일 시트를 삽입하는 방법에는 세 가지가 있습니다.

- 외부 CSS
- 내부 CSS
- 인라인 CSS

## 외부 CSS

외부 스타일 시트를 사용하면 하나의 파일 만 변경하여 전체 웹 사이트의 모양을 변경할 수 있습니다!

각 HTML 페이지에는 헤드 섹션 내부의 `<link>` 요소 내부에 있는 외부 스타일 시트 파일에 대한 참조가 포함되어야 합니다.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

외부 스타일은 HTML 페이지의 `<head>` 섹션 내에 있는 `<link>` 요소 내에 정의됩니다.

외부 스타일 시트는 모든 텍스트 편집기에서 쓸 수 있으며 확장자는 `.css` 로 저장해야 합니다.

외부 .css 파일에는 HTML 태그가 포함되지 않아야합니다.

"mystyle.css"파일은 다음과 같습니다.

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```

**참고** : 속성 값과 단위 사이에 공백을 추가하지 마십시오 (예 :) `margin-left: 20px;`. 올바른 방법은 다음과 같습니다. `margin-left: 20px;`

## 내부 CSS

하나의 단일 HTML 페이지에 고유 한 스타일이 있는 경우 내부 스타일 시트를 사용할 수 있습니다.

내부 스타일은 헤드 섹션 내부의 `<style>` 요소 내에 정의됩니다.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    background-color: linen;  
}  
  
h1 {  
    color: maroon;  
    margin-left: 40px;  
}  
</style>  
</head>  
<body>  
  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
  
</body>  
</html>
```

내부 스타일은 HTML 페이지의 `<head>` 섹션 내에있는 `<style>` 요소 내에 정의됩니다.

# 인라인 CSS

인라인 스타일을 사용하여 단일 요소에 고유 한 스타일을 적용 할 수 있습니다.

인라인 스타일을 사용하려면 관련 요소에 스타일 속성을 추가하십시오. **style** 속성은 모든 CSS 속성을 포함 할 수 있습니다.

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

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

인라인 스타일은 관련 요소의 "style"속성 내에 정의됩니다.

**팁 :** 인라인 스타일은 콘텐츠와 프리젠테이션을 혼합하여 스타일 시트의 많은 이점을 잃습니다. 이 방법은 드물게 사용하십시오.

## 여러 스타일 시트

다른 스타일 시트에서 동일한 선택기 (요소)에 대해 일부 특성이 정의 된 경우 마지막 읽기 스타일 시트의 값이 사용됩니다.

**외부 스타일 시트 의 <H1> 요소 다음 스타일을 갖는다 :**

```
h1 {
  color: navy;
}
```

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

```

<style>
h1 {
  color: orange;
}
</style>
</head>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>

```

외부 스타일 시트에 대한 링크 **이후** 내부 스타일이 정의 되면 <h1> 요소는 "주황색"입니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>

```

외부 스타일 시트에 대한 링크 **전에** 내부 스타일이 정의 된 경우 <h1> 요소는 "네이비"가 됩니다.

## 계단식 순서

HTML 요소에 대해 둘 이상의 스타일이 지정된 경우 어떤 스타일이 사용됩니까?

페이지의 모든 스타일은 다음 규칙에 따라 새로운 "가상"스타일 시트로 "캐스케이드"됩니다.

### 1. 인라인 스타일 (HTML 요소 내부)



2. 외부 및 내부 스타일 시트 (헤드 섹션)
3. 브라우저 기본값

따라서 인라인 스타일의 우선 순위가 가장 높으며 외부 및 내부 스타일과 브라우저 기본값을 재정의합니다.

## CSS 색상

색상은 사전 정의 된 색상 이름 또는 RGB, HEX, HSL, RGBA, HSLA 값을 사용하여 지정됩니다.

## CSS 색상 이름

CSS 에서 색상 이름을 사용하여 색상을 지정할 수 있습니다.

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

<h1 style="background-color:Tomato;">Tomato</h1>
<h1 style="background-color:Orange;">Orange</h1>
<h1 style="background-color:DodgerBlue;">DodgerBlue</h1>
<h1 style="background-color:MediumSeaGreen;">MediumSeaGreen</h1>
<h1 style="background-color:Gray;">Gray</h1>
<h1 style="background-color:SlateBlue;">SlateBlue</h1>
<h1 style="background-color:Violet;">Violet</h1>
<h1 style="background-color:LightGray;">LightGray</h1>

</body>
</html>
```

HTML은 [140 개의 표준 색상 이름을](#) 지원 [합니다](#).

## CSS 배경색

HTML 요소의 배경색을 설정할 수 있습니다.

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

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>

</body>
</html>
```

## CSS 텍스트 색상

텍스트 색상을 설정할 수 있습니다.

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

<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>

</body>
</html>
```

## CSS 테두리 색상

테두리 색상을 설정할 수 있습니다.

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

<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>

</body>
</html>
```

## CSS 색상 값

HTML에서 RGB 값, HEX 값, HSL 값, RGBA 값 및 HSLA 값을 사용하여 색상을 지정할 수도 있습니다.

```

<!DOCTYPE html>
<html>
<body>

<p>Same as color name "Tomato":</p>

<h1 style="background-color:rgb(255, 99, 71);">rgb(255, 99, 71)</h1>
<h1 style="background-color:#ff6347;">#ff6347</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">hsl(9, 100%, 64%)</h1>

<p>Same as color name "Tomato", but 50% transparent:</p>
<h1 style="background-color:rgba(255, 99, 71, 0.5);">rgba(255, 99, 71, 0.5)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">hsla(9, 100%, 64%, 0.5)</h1>

<p>In addition to the predefined color names, colors can be specified using RGB,
HEX, HSL, or even transparent colors using RGBA or HSLA color values.</p>

</body>
</html>

```

## CSS RGB 값

HTML 에서 다음 공식을 사용하여 색상을 RGB 값으로 지정할 수 있습니다.

### **rgb(*red*, *green*, *blue*)**

각 매개 변수 (빨간색, 녹색 및 파란색)는 0 에서 255 사이의 색상 강도를 정의합니다.

예를 들어, rgb (255, 0, 0)는 빨간색이 가장 높은 값 (255)으로 설정되고 다른 값은 0 으로 설정되므로 빨간색으로 표시됩니다.

검은 색을 표시하려면 rgb (0, 0, 0)과 같이 모든 색상 매개 변수를 0 으로 설정하십시오.

흰색을 표시하려면 rgb (255, 255, 255)와 같이 모든 색상 매개 변수를 255 로 설정하십시오

```

<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:rgb(255, 0, 0);">rgb(255, 0, 0)</h1>
<h1 style="background-color:rgb(0, 0, 255);">rgb(0, 0, 255)</h1>
<h1 style="background-color:rgb(60, 179, 113);">rgb(60, 179, 113)</h1>
<h1 style="background-color:rgb(238, 130, 238);">rgb(238, 130, 238)</h1>
<h1 style="background-color:rgb(255, 165, 0);">rgb(255, 165, 0)</h1>
<h1 style="background-color:rgb(106, 90, 205);">rgb(106, 90, 205)</h1>

```

```
<p>In HTML, you can specify colors using RGB values.</p>
</body>
</html>
```

회색 음영은 종종 세 가지 광원 모두에 대해 동일한 값을 사용하여 정의됩니다.

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

<h1 style="background-color:rgb(0, 0, 0);">rgb(0, 0, 0)</h1>
<h1 style="background-color:rgb(60, 60, 60);">rgb(60, 60, 60)</h1>
<h1 style="background-color:rgb(120, 120, 120);">rgb(120, 120, 120)</h1>
<h1 style="background-color:rgb(180, 180, 180);">rgb(180, 180, 180)</h1>
<h1 style="background-color:rgb(240, 240, 240);">rgb(240, 240, 240)</h1>
<h1 style="background-color:rgb(255, 255, 255);">rgb(255, 255, 255)</h1>

<p>By using equal values for red, green, and blue, you will get different shades
of gray.</p>

</body>
</html>
```

## CSS 16 진수 값

HTML에서는 다음과 같은 형식의 16 진수 값을 사용하여 색상을 지정할 수 있습니다.

### # *rrggbb*

rr (빨간색), gg (녹색) 및 bb (파란색)는 00 과 ff 사이의 16 진수 값입니다 (소수점 0-255 와 동일).

예를 들어, 빨간색은 최대 값 (ff)으로 설정되고 나머지는 최저값 (00)으로 설정되므로 # ff0000 은 빨간색으로 표시됩니다.

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

<h1 style="background-color:#ff0000;">#ff0000</h1>
<h1 style="background-color:#0000ff;">#0000ff</h1>
<h1 style="background-color:#3cb371;">#3cb371</h1>
<h1 style="background-color:#ee82ee;">#ee82ee</h1>
<h1 style="background-color:#ffa500;">#ffa500</h1>
<h1 style="background-color:#6a5acd;">#6a5acd</h1>
```

```
<p>In HTML, you can specify colors using Hex values.</p>
</body>
</html>
```

회색 음영은 종종 세 가지 광원 모두에 대해 동일한 값을 사용하여 정의됩니다.

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

<h1 style="background-color:#000000;">#000000</h1>
<h1 style="background-color:#3c3c3c;">#3c3c3c</h1>
<h1 style="background-color:#787878;">#787878</h1>
<h1 style="background-color:#b4b4b4;">#b4b4b4</h1>
<h1 style="background-color:#f0f0f0;">#f0f0f0</h1>
<h1 style="background-color:#ffffff;">#ffffff</h1>

<p>By using equal values for red, green, and blue, you will get different shades
of gray.</p>

</body>
</html>
```

## HSL 값

HTML에서 색조, 채도 및 명도 (HSL)를 사용하여 다음과 같은 형식으로 색상을 지정할 수 있습니다.

### **hsl(hue, saturation, lightness)**

색조는 0에서 360 사이의 색상환 각도입니다. 0은 빨간색, 120은 녹색, 240은 파란색입니다.

채도는 백분율 값이고 0%는 회색 음영을 의미하고 100%는 풀 컬러입니다.

밝기도 백분율이며 0%는 검은 색이고 50%는 밝거나 어둡지 않으며 100%는 흰색입니다.

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

<h1 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h1>
<h1 style="background-color:hsl(240, 100%, 50%);">hsl(240, 100%, 50%)</h1>
<h1 style="background-color:hsl(147, 50%, 47%);">hsl(147, 50%, 47%)</h1>
```

```
<h1 style="background-color:hsl(300, 76%, 72%);">hsl(300, 76%, 72%)</h1>
<h1 style="background-color:hsl(39, 100%, 50%);">hsl(39, 100%, 50%)</h1>
<h1 style="background-color:hsl(248, 53%, 58%);">hsl(248, 53%, 58%)</h1>

<p>In HTML, you can specify colors using HSL values.</p>

</body>
</html>
```

## Saturation

채도는 색상의 강도로 설명 할 수 있습니다.

100 %는 순수한 색상이며 회색 음영이 없습니다.

50 %는 50 % 회색이지만 여전히 색상을 볼 수 있습니다.

0 %는 완전히 회색이며 더 이상 색상을 볼 수 없습니다.

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

<h1 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h1>
<h1 style="background-color:hsl(0, 80%, 50%);">hsl(0, 80%, 50%)</h1>
<h1 style="background-color:hsl(0, 60%, 50%);">hsl(0, 60%, 50%)</h1>
<h1 style="background-color:hsl(0, 40%, 50%);">hsl(0, 40%, 50%)</h1>
<h1 style="background-color:hsl(0, 20%, 50%);">hsl(0, 20%, 50%)</h1>
<h1 style="background-color:hsl(0, 0%, 50%);">hsl(0, 0%, 50%)</h1>

<p>With HSL colors, less saturation mean less color. 0% is completely gray.</p>

</body>
</html>
```

## Lightness

색상의 명도는 색상을 제공하려는 빛의 양으로 설명 할 수 있습니다. 0 %는 빛이 없음 (검정)을 의미하고 50 %는 50 % 빛을 의미합니다 (어둡지도 밝지 않음) 100 %는 완전 밝기 (흰색)를 의미합니다.

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

<h1 style="background-color:hsl(0, 100%, 0%);">hsl(0, 100%, 0%)</h1>
```

```

<h1 style="background-color:hsl(0, 100%, 25%);">hsl(0, 100%, 25%)</h1>
<h1 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h1>
<h1 style="background-color:hsl(0, 100%, 75%);">hsl(0, 100%, 75%)</h1>
<h1 style="background-color:hsl(0, 100%, 90%);">hsl(0, 100%, 90%)</h1>
<h1 style="background-color:hsl(0, 100%, 100%);">hsl(0, 100%, 100%)</h1>

<p>With HSL colors, 0% lightness means black, and 100 lightness means white.</p>

</body>
</html>

```

회색 음영은 종종 색조와 채도를 0으로 설정하고 밝기를 0 %에서 100 %로 조정하여 더 어둡고 밝은 음영을 얻습니다.

```

<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:hsl(0, 0%, 0%);">hsl(0, 0%, 0%)</h1>
<h1 style="background-color:hsl(0, 0%, 24%);">hsl(0, 0%, 24%)</h1>
<h1 style="background-color:hsl(0, 0%, 47%);">hsl(0, 0%, 47%)</h1>
<h1 style="background-color:hsl(0, 0%, 71%);">hsl(0, 0%, 71%)</h1>
<h1 style="background-color:hsl(0, 0%, 94%);">hsl(0, 0%, 94%)</h1>
<h1 style="background-color:hsl(0, 0%, 100%);">hsl(0, 0%, 100%)</h1>

<p>With HSL colors, shades of gray are made by setting the saturation to 0%, and adjusting the lightness according to how dark/light the gray color should be.</p>

</body>
</html>

```

## RGBA 값

RGBA 색상 값은 알파 채널을 사용하여 RGB 색상 값을 확장 한 것으로 색상의 불투명도를 지정합니다.

RGBA 색상 값은 다음과 같이 지정됩니다.

**rgba(*red*, *green*, *blue*, *alpha*)**

alpha 매개 변수는 0.0 (완전 투명)과 1.0 (완전 투명하지 않음) 사이의 숫자입니다.

```

<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:rgba(255, 99, 71, 0);">rgba(255, 99, 71, 0)</h1>

```

```

<h1 style="background-color:rgba(255, 99, 71, 0.2);">rgba(255, 99, 71, 0.2)</h1>
<h1 style="background-color:rgba(255, 99, 71, 0.4);">rgba(255, 99, 71, 0.4)</h1>
<h1 style="background-color:rgba(255, 99, 71, 0.6);">rgba(255, 99, 71, 0.6)</h1>
<h1 style="background-color:rgba(255, 99, 71, 0.8);">rgba(255, 99, 71, 0.8)</h1>
<h1 style="background-color:rgba(255, 99, 71, 1);">rgba(255, 99, 71, 1)</h1>

<p>You can make transparent colors by using the RGBA color value.</p>

</body>
</html>

```

## HSLA 값

HSLA 색상 값은 색상의 불투명도를 지정하는 알파 채널을 사용한 HSL 색상 값의 확장입니다.

HSLA 색상 값은 다음과 같이 지정됩니다.

**hsla(hue, saturation, lightness, alpha)**

alpha 매개 변수는 0.0 (완전 투명)과 1.0 (완전 투명하지 않음) 사이의 숫자입니다.

```

<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:hsla(9, 100%, 64%, 0);">hsla(9, 100%, 64%, 0)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.2);">hsla(9, 100%, 64%, 0.2)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.4);">hsla(9, 100%, 64%, 0.4)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.6);">hsla(9, 100%, 64%, 0.6)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.8);">hsla(9, 100%, 64%, 0.8)</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 1);">hsla(9, 100%, 64%, 1)</h1>

<p>You can make transparent colors by using the HSLA color value.</p>

</body>
</html>

```

## CSS 배경

CSS 배경 속성은 요소의 배경 효과를 정의하는 데 사용됩니다.



CSS 배경 속성 :

- 배경색
- 배경 이미지
- 배경 반복
- 배경 첨부
- 배경 위치

## CSS 배경색

이 `background-color` 속성은 요소의 배경색을 지정합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightblue;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>This page has a light blue background color!</p>

</body>
</html>
```

CSS 를 사용하면 색상이 가장 자주 지정됩니다.

- "빨간색"과 같은 유효한 색상 이름
- "#ff0000"과 같은 16 진수 값
- "rgb (255,0,0)"와 같은 RGB 값

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  background-color: green;
}

div {
  background-color: lightblue;
}
```

```

p {
  background-color: yellow;
}
</style>
</head>
<body>

<h1>CSS background-color example!</h1>
<div>
This is a text inside a div element.
<p>This paragraph has its own background color.</p>
We are still in the div element.
</div>

</body>
</html>

```

<h1>, <p> 및 <div> 요소는 서로 다른 배경색을 갖습니다.

## CSS 배경 이미지

이 `background-image` 속성은 요소의 배경으로 사용할 이미지를 지정합니다.

기본적으로 이미지는 반복되어 전체 요소를 덮습니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("paper.gif");
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>This page has an image as the background!</p>

</body>
</html>

```

```

<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("bgdesert.jpg");
}

```

```

</style>
</head>
<body>

<h1>Hello World!</h1>
<p>This page has an image as the background!</p>

</body>
</html>

```

이 예는 텍스트와 배경 이미지 의 잘못된 조합 을 보여줍니다 . 텍스트는 거의 읽을 수 없습니다.

**참고 :** 배경 이미지를 사용할 때는 텍스트를 방해하지 않는 이미지를 사용하십시오.

## CSS 배경 반복

기본적 **background-image** 으로이 속성은 이미지를 가로 및 세로로 반복합니다.

일부 이미지는 가로 또는 세로로만 반복해야 합니다. 그렇지 않으면 다음과 같이 이상하게 보일 것입니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("gradient_bg.png");
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>Strange background image...</p>

</body>
</html>

```

위의 이미지를 가로로만 반복하면 (**background-repeat: repeat-x;**) 배경이 더 좋아 보입니다.

```

<!DOCTYPE html>
<html>
<head>
<style>

```

```

body {
  background-image: url("gradient_bg.png");
  background-repeat: repeat-x;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>Here, a background image is repeated only horizontally!</p>

</body>
</html>

```

**팁** : 이미지를 세로로 반복하려면 `background-repeat: repeat-y;`

## CSS 배경 반복 : 반복 없음

배경 이미지를 한 번만 표시하는 것은 `background-repeat` 속성에 의해 지정됩니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>W3Schools background image example.</p>
<p>The background image is only showing once, but it is disturbing the
reader!</p>

</body>
</html>

```

위의 예에서 배경 이미지는 텍스트와 같은 위치에 배치됩니다. 이미지의 위치를 변경하여 텍스트를 너무 방해하지 않도록 하고 싶습니다.

# CSS 배경 위치

이 `background-position` 속성은 배경 이미지의 위치를 지정하는 데 사용됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  margin-right: 200px;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p> W3Schools background no-repeat, set position example.</p>
<p>Now the background image is only shown once, and positioned away from the
text.</p>
<p>In this example we have also added a margin on the right side, so the
background image will never disturb the text.</p>

</body>
</html>
```

위의 예에서 배경 이미지는 텍스트와 같은 위치에 배치됩니다. 이미지의 위치를 변경

# CSS 배경 첨부

이 `background-attachment` 속성은 배경 이미지를 스크롤할지 고정 할지를 지정합니다 (페이지의 나머지 부분과 함께 스크롤되지 않음).

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: fixed;
  margin-right: 200px;
}
</style>
</head>
```

[illegible]

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: scroll;
  margin-right: 200px;
}
</style>
</head>
<body>

<h1>The background-attachment Property</h1>
<p> The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page).</p>
<p><strong>Tip:</strong> If you do not see any scrollbars, try to resize the browser window.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
```

```

<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
<p> The background-image scrolls. Try to scroll down the page.</p>
</body>
</html>

```

배경 이미지가 나머지 페이지와 함께 스크롤되도록 지정하십시오.

## CSS 배경-속기 속성

코드를 단축하기 위해 하나의 단일 속성에 모든 배경 속성을 지정할 수도 있습니다. 이것을 속기 속성이라고 합니다.

배경의 속기 속성은 **background** 입니다.

속기 속성을 사용하여 한 번의 선언으로 모든 배경 속성을 설정하십시오.

```

<!DOCTYPE html>
<html>
<head>
<style>
body {
  background: #ffffff url("img_tree.png") no-repeat right top;
  margin-right: 200px;
}
</style>
</head>
<body>

<h1>The background Property</h1>
<p>The background property is a shorthand property for specifying all the
background properties in one declaration.</p>
<p>Here, the background image is only shown once, and it is also positioned in

```

```
the top-right corner.</p>
<p>We have also added a right margin, so that the text will not write over the
background image.</p>

</body>
</html>
```

속기 속성을 사용하는 경우 속성 값의 순서는 다음과 같습니다.

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

다른 값이 이 순서에 있는 한 속성 값 중 하나가 누락 되어도 문제가 되지 않습니다.

## 모든 CSS 배경 속성

Property	Description
<a href="#">background</a>	Sets all the background properties in one declaration
<a href="#">background-attachment</a>	Sets whether a background image is fixed or scrolls with the rest of the page
<a href="#">background-clip</a>	Specifies the painting area of the background
<a href="#">background-color</a>	Sets the background color of an element
<a href="#">background-image</a>	Sets the background image for an element
<a href="#">background-origin</a>	Specifies where the background image(s) is/are positioned
<a href="#">background-position</a>	Sets the starting position of a background image
<a href="#">background-repeat</a>	Sets how a background image will be repeated
<a href="#">background-size</a>	Specifies the size of the background image(s)

## CSS 테두리

### CSS 테두리 속성

CSS **border** 속성을 사용하면 요소 테두리의 스타일, 너비 및 색상을 지정할 수 있습니다.



나는 모든 면에 테두리가 있습니다.

빨간색 아래쪽 테두리가 있습니다.

둥근 테두리가 있습니다.

파란색 왼쪽 테두리가 있습니다.

## CSS 테두리 스타일

이 `border-style` 속성은 표시 할 테두리 종류를 지정합니다.

다음과 같은 값이 허용됩니다.

- `dotted` -점선 테두리를 정의
- `dashed` -점선 테두리를 정의합니다
- `solid` -단단한 테두리를 정의
- `double` -이중 테두리를 정의
- `groove` -3D 그루브 테두리를 정의합니다. 효과는 테두리 색상 값에 따라 다릅니다.
- `ridge` -3D 능선 테두리를 정의합니다. 효과는 테두리 색상 값에 따라 다릅니다.
- `inset` -3D 삽입 테두리를 정의합니다. 효과는 테두리 색상 값에 따라 다릅니다.
- `outset` -3D 시작 경계를 정의합니다. 효과는 테두리 색상 값에 따라 다릅니다.
- `none` -경계를 정의하지 않습니다
- `hidden` -숨겨진 테두리를 정의

이 `border-style` 속성은 1-4 개의 값을 가질 수 있습니다 (상단 테두리, 오른쪽 경계, 아래쪽 경계 및 왼쪽 경계).

```
<!DOCTYPE html>
<html>
<head>
<style>
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
```

```

p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
</style>
</head>
<body>

<h1>The border-style Property</h1>
<p>This property specifies what kind of border to display:</p>
<p class="dotted"> A dotted border.</p>
<p class="dashed"> A dashed border.</p>
<p class="solid"> A solid border.</p>
<p class="double"> A double border.</p>
<p class="groove"> A groove border.</p>
<p class="ridge"> A ridge border.</p>
<p class="inset"> A inset border.</p>
<p class="outset"> A outset border.</p>
<p class="none"> No border.</p>
<p class="hidden"> A hidden border.</p>
<p class="mix"> A mixed border.</p>

</body>
</html>

```

**참고 :** 아래에 설명 된 OTHER CSS 테두리 속성은 **border-style** 속성을 설정 하지 않으면 어떤 영향도 미치지 않습니다 !

## CSS 테두리 너비

이 **border-width** 속성은 네 테두리의 너비를 지정합니다.

너비는 특정 크기 (px, pt, cm, em 등)로 설정하거나 세 가지 사전 정의 된 값 중 하나를 사용하여 설정할 수 있습니다: **thin**, **medium** 또는 **thick**.

이 **border-width** 속성은 1-4 개의 값을 가질 수 있습니다 (상단 테두리, 오른쪽 경계, 아래쪽 경계 및 왼쪽 경계).

```

<!DOCTYPE html>
<html>
<head>
<style>
p.one {
  border-style: solid;
  border-width: 5px;
}
p.two {
  border-style: solid;

```

```

    border-width: medium;
}
p.three {
    border-style: dotted;
    border-width: 2px;
}
p.four {
    border-style: dotted;
    border-width: thick;
}
p.five {
    border-style: double;
    border-width: 15px;
}
p.six {
    border-style: double;
    border-width: thick;
}
p.seven {
    border-style: solid;
    border-width: 2px 10px 4px 20px;
}
</style>
</head>
<body>

<h1>The border-width Property</h1>
<p>This property specifies the width of the four borders:</p>

<p class="one">Some text.</p>
<p class="two">Some text.</p>
<p class="three">Some text.</p>
<p class="four">Some text.</p>
<p class="five">Some text.</p>
<p class="six">Some text.</p>
<p class="seven">Some text.</p>

<p><b>Note:</b> The "border-width" property does not work if it is used alone.
Always specify the "border-style" property to set the borders first.</p>

</body>
</html>

```

## CSS 테두리 색상

이 `border-color` 속성은 네 테두리의 색상을 설정하는 데 사용됩니다.

색상은 다음과 같이 설정할 수 있습니다.

- name- "red"와 같은 색상 이름을 지정하십시오
- 16 진수- "# ff0000"과 같은 16 진수 값을 지정하십시오.

- RGB- "rgb (255,0,0)"와 같은 RGB 값을 지정하십시오.
- 투명한

이 **border-color** 속성은 1-4 개의 값을 가질 수 있습니다 (상단 테두리, 오른쪽 경계, 아래쪽 경계 및 왼쪽 경계).

**border-color** 설정되지 않은 경우, 그것은 요소의 색상을 상속합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
  border-style: solid;
  border-color: red;
}

p.two {
  border-style: solid;
  border-color: green;
}

p.three {
  border-style: solid;
  border-color: red green blue yellow;
}
</style>
</head>
<body>

<h1>The border-color Property</h1>
<p>This property specifies the width of the four borders:</p>

<p class="one">A solid red border</p>
<p class="two">A solid green border</p>
<p class="three">A solid multicolor border</p>

<p><b>Note:</p>The "border-color" property does not work if it is used alone. Use
the "border-style" property to set the borders first.</p>

</body>
</html>
```

## CSS 테두리-개별 측면

위의 예에서 각 면에 대해 서로 다른 테두리를 지정할 수 있음을 알았습니다.

CSS 에는 각 테두리 (위쪽, 오른쪽, 아래쪽 및 왼쪽)를 지정하기 위한 속성도 있습니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
p {
  border-top-style: dotted;
  border-right-style: solid;
  border-bottom-style: dotted;
  border-left-style: solid;
}
</style>
</head>
<body>

<p>2 different border styles.</p>

</body>
</html>

```

```

<!DOCTYPE html>
<html>
<head>
<style>
p {
  border-style: dotted solid;
}
</style>
</head>
<body>

<p>2 different border styles.</p>

</body>
</html>

```

작동 방식은 다음과 같습니다.

은 if **border-style** 속성은 네 개의 값이 있다 :

- 테두리 스타일 : 점선으로 된 이중 점선;
  - 상단 테두리가 점선
  - 오른쪽 테두리가 튼튼합니다
  - 아래쪽 테두리는 두 배입니다
  - 왼쪽 테두리가 점선

경우 **border-style** 속성은 세 가지 값을 가집니다 :

- 테두리 스타일 : 점선으로 된 이중 이중;
  - 상단 테두리가 점선
  - 오른쪽과 왼쪽 테두리는 단단합니다
  - 아래쪽 테두리는 두 배입니다

경우 `border-style` 속성은 두 값이있다 :

- 테두리 스타일 : 점선으로 된 점선;
  - 위쪽 및 아래쪽 테두리가 점선으로 표시됩니다
  - 오른쪽과 왼쪽 테두리는 단단합니다

경우 `border-style` 속성은 하나 개의 값을 가지고 :

- 테두리 스타일 : 점선;
  - 네 개의 테두리가 모두 점선

이 `border-style` 예에서는 이 속성이 사용됩니다. 그러나 `border-width` 및 `border-color` 와도 작동합니다.

## CSS 테두리-속기 속성

위의 예에서 볼 수 있듯이 테두리를 다룰 때 고려해야 할 많은 속성이 있습니다.

코드를 단축하기 위해 하나의 속성에 모든 개별 테두리 속성을 지정할 수도 있습니다.

이 `border` 속성은 다음과 같은 개별 테두리 속성의 속기 속성입니다.

- `border-width`
- `border-style` (필수)
- `border-color`

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border: 5px solid red;
}
</style>
</head>
<body>

<p>The border Property</p>
<p>This property is a shorthand property for border-width, border-style, and
border-color.</p>

</body>
</html>
```

한쪽에만 모든 개별 테두리 속성을 지정할 수도 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border-left: 6px solid red;
  background-color: lightgrey;
}
</style>
</head>
<body>

<p>The border-left Property</p>
<p>This property is a shorthand property for border-left-width, border-left-style, and border-left-color.</p>

</body>
</html>
```

아래쪽 테두리

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border-left: 6px solid red;
  background-color: lightgrey;
}
</style>
</head>
<body>

<p>The border-bottom Property</p>
<p>This property is a shorthand property for border-bottom-width, border-bottom-style, and border-bottom-color.</p>

</body>
</html>
```

## CSS 둥근 테두리

이 `border-radius` 속성은 요소에 둥근 테두리를 추가하는 데 사용됩니다

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```

p.normal {
  border: 2px solid red;
}
p.round1 {
  border: 2px solid red;
  border-radius: 5px;
}
p.round2 {
  border: 2px solid red;
  border-radius: 8px;
}
p.round3 {
  border: 2px solid red;
  border-radius: 12px;
}
</style>
</head>
<body>

<h2>The border-radius Property</h2>
<p>This property is used to add rounded borders to an element:</p>

<p class="normal">Normal border</p>
<p class="round1">Round border</p>
<p class="round2">Rounder border</p>
<p class="round3">Roundest border</p>

<p><b>Note:</b><p>The "border-radius" property is not supported in IE8 and earlier versions.</p>

</body>
</html>

```

**참고** : 이 **border-radius** 속성은 IE8 및 이전 버전에서는 지원되지 않습니다.

## 모든 CSS 테두리 속성

Property	Description
<a href="#">border</a>	Sets all the border properties in one declaration
<a href="#">border-bottom</a>	Sets all the bottom border properties in one declaration
<a href="#">border-bottom-color</a>	Sets the color of the bottom border
<a href="#">border-bottom-style</a>	Sets the style of the bottom border
<a href="#">border-bottom-width</a>	Sets the width of the bottom border
<a href="#">border-color</a>	Sets the color of the four borders
<a href="#">border-left</a>	Sets all the left border properties in one declaration
<a href="#">border-left-color</a>	Sets the color of the left border
<a href="#">border-left-style</a>	Sets the style of the left border
<a href="#">border-left-width</a>	Sets the width of the left border
<a href="#">border-radius</a>	Sets all the four border-*-radius properties for



	rounded corners
<a href="#">border-right</a>	Sets all the right border properties in one declaration
<a href="#">border-right-color</a>	Sets the color of the right border
<a href="#">border-right-style</a>	Sets the style of the right border
<a href="#">border-right-width</a>	Sets the width of the right border
<a href="#">border-style</a>	Sets the style of the four borders
<a href="#">border-top</a>	Sets all the top border properties in one declaration
<a href="#">border-top-color</a>	Sets the color of the top border
<a href="#">border-top-style</a>	Sets the style of the top border
<a href="#">border-top-width</a>	Sets the width of the top border
<a href="#">border-width</a>	Sets the width of the four borders

## CSS 여백

## CSS 여백

CSS `margin` 속성은 정의 된 경계 외부에서 요소 주위에 공간을 만드는 데 사용됩니다.

CSS 를 사용하면 여백을 완전히 제어 할 수 있습니다. 요소의 각 측면에 대한 여백을 설정하는 속성이 있습니다 (위쪽, 오른쪽, 아래쪽 및 왼쪽).

## 여백-개인 측

CSS 에는 요소의 각 측면에 대한 여백을 지정하는 속성이 있습니다.

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

모든 여백 속성은 다음 값을 가질 수 있습니다.

- `auto`-브라우저가 마진을 계산합니다
- `length` -px, pt, cm 등의 여백을 지정합니다.
- `%` -포함 요소 너비의 여백을 %로 지정합니다
- `inherit`-마진이 부모 요소에서 상속되도록 지정합니다

팁 : 음수 값이 허용됩니다.

<p> 요소의 네면 모두에 대해 서로 다른 여백을 설정하십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>Using individual margin properties </h2>
<div>This div element has a top margin of 100px, a right margin of 150px, a
bottom margin of 100px, and a left margin of 80px. </div>

</body>
</html>
```

## 여백-속기 속성

코드를 줄이려면 한 속성에 모든 여백 속성을 지정할 수 있습니다.

이 **margin** 속성은 다음과 같은 개별 여백 속성의 속기 속성입니다.

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

작동 방식은 다음과 같습니다.

은 if **margin** 속성은 네 개의 값이 있다 :

- **여백 : 25px 50px 75px 100px;**
  - 상단 여백은 25px 입니다
  - 오른쪽 여백은 50px 입니다
  - 아래쪽 여백은 75px 입니다

- 왼쪽 여백은 100px 입니다

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin: 25px 50px 75px 100px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>The margin shorthand property - 4 values</h2>
<div>This div element has a top margin of 25px, a right margin of 50px, a bottom
margin of 75px, and a left margin of 100px.</div>

</body>
</html>
```

경우 **margin** 속성은 세 가지 값을가집니다 :

- **여백 : 25px 50px 75px;**
  - 상단 여백은 25px 입니다
  - 좌우 여백은 50px 입니다
  - 아래쪽 여백은 75px 입니다

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin: 25px 50px 75px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>The margin shorthand property - 3 values</h2>
<div>This div element has a top margin of 25px, a right and left margin of 50px,
and a bottom margin of 75px.</div>

</body>
</html>
```

경우 **margin** 속성은 두 값이있다 :

- **여백 : 25px 50px;**
  - 위쪽과 아래쪽 여백은 25px 입니다
  - 좌우 여백은 50px 입니다

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin: 25px 50px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>The margin shorthand property - 2 values</h2>
<div>This div element has a top and bottom margin of 25px, and a right and left
margin of 50px.</div>

</body>
</html>
```

경우 **margin** 속성은 하나 개의 값을 가지고 :

- **여백 : 25px;**
  - 네 마진은 모두 25px 입니다

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin: 25px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>The margin shorthand property - 1 value</h2>
<div>This div element has a top, bottom, left, and right margin of 25px.</div>

</body>
</html>
```

# 자동 가치

컨테이너 내에서 요소를 가로로 가운데로 배치 하도록 `margin` 속성을 `auto` 설정할 수 있습니다.

그러면 요소가 지정된 너비를 차지하고 남은 공간이 왼쪽과 오른쪽 여백 사이에서 균등하게 분할됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  margin: auto;
  border: 1px solid red;
}
</style>
</head>
<body>

<h2>Use of margin:auto</h2>
<p>You can set the margin property to auto to horizontally center the element
within its container. The element will then take up the specified width, and the
remaining space will be split equally between the left and right margins:</p>

<div>This div will be horizontally centered because it has margin: auto;</div>

</body>
</html>
```

# 상속 가치

이 예에서는 `<p class = "ex1">` 요소의 왼쪽 여백을 부모 요소 (`<div>`)에서 상속합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid red;
  margin-left: 100px;
}

p.ex1 {
  margin-left: inherit;
}
```

```

</style>
</head>
<body>

<h2>Use of the inherit value</h2>
<p>Let the left margin be inherited from the parent element:</p>

<div>
<p class="ex1">This paragraph has an inherited left margin (from the div
element).</p>
</div>

</body>
</html>

```

## 여백 붕괴

요소의 상단 및 하단 여백이 때때로 두 여백 중 가장 큰 단일 여백으로 축소됩니다.

왼쪽 및 오른쪽 여백에서는 발생하지 않습니다! 위쪽과 아래쪽 여백 만!

다음 예를 보십시오.

```

<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  margin: 0 0 50px 0;
}

h2 {
  margin: 20px 0 0 0;
}
</style>
</head>
<body>

<p>In this example the h1 element has a bottom margin of 50px and the h2 element
has a top margin of 20px. Then, the vertical margin between h1 and h2 should have
been 70px (50px + 20px). However, due to margin collapse, the actual margin ends
up being 50px.</p>

<h1>Heading 1</h1>
<h2>Heading 2</h2>

</body>
</html>

```

위의 예에서 <h1> 요소의 아래쪽 여백은 50px 이고 <h2> 요소의 위쪽 여백은 20px 로 설정되어 있습니다.

상식은 <h1>과 <h2> 사이의 수직 여백이 총 70px (50px + 20px)임을 암시하는 것 같습니다. 그러나 마진 축소로 인해 실제 마진은 50px 입니다.

## 모든 CSS 여백 속성

Property	Description
<a href="#">margin</a>	A shorthand property for setting the margin properties in one declaration
<a href="#">margin-bottom</a>	Sets the bottom margin of an element
<a href="#">margin-left</a>	Sets the left margin of an element
<a href="#">margin-right</a>	Sets the right margin of an element
<a href="#">margin-top</a>	Sets the top margin of an element

## CSS 패딩

## CSS 패딩

CSS `padding` 속성은 정의 된 테두리 내에서 요소 내용 주위에 공간을 생성하는 데 사용됩니다.

CSS 를 사용하면 패딩을 완전히 제어 할 수 있습니다. 요소의 각 측면에 패딩을 설정하는 속성이 있습니다 (위쪽, 오른쪽, 아래쪽 및 왼쪽).

## 패딩-개인면

CSS 에는 요소의 각 측면에 패딩을 지정하는 속성이 있습니다.

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

모든 패딩 속성은 다음 값을 가질 수 있습니다.

- `길/0/-px`, `pt`, `cm` 등으로 패딩을 지정합니다.
- `%` -포함 요소 너비의 `%`로 패딩을 지정합니다

- inherit-패딩이 부모 요소에서 상속되도록 지정합니다

참고 : 음수 값은 허용되지 않습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  background-color: lightblue;
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
</style>
</head>
<body>

<h2>Using individual padding properties</h2>

<div>This div element has a top padding of 50px, a right padding of 30px, a
bottom padding of 50px, and a left padding of 80px. </div>

</body>
</html>
```

## 패딩-속기 속성

코드를 줄이려면 하나의 속성에 모든 패딩 속성을 지정할 수 있습니다.

이 **padding** 속성은 다음과 같은 개별 패딩 속성의 속기 속성입니다.

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

작동 방식은 다음과 같습니다.

은 if **padding** 속성은 네 개의 값이 있다 :

- 패딩 : **25px 50px 75px 100px;**
  - 상단 패딩은 25px 입니다
  - 오른쪽 여백은 50px 입니다



- 하단 패딩은 75px 입니다
- 왼쪽 여백은 100px 입니다

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  background-color: lightblue;
  padding: 25px 50px 75px 100px;
}
</style>
</head>
<body>

<h2>The padding shorthand property - 4 values</h2>

<div>This div element has a top padding of 25px, a right padding of 50px, a
bottom padding of 75px, and a left padding of 100px.</div>

</body>
</html>
```

경우 **padding** 속성은 세 가지 값을 가집니다 :

- **패딩 : 25px 50px 75px;**
  - 상단 패딩은 25px 입니다
  - 좌우 패딩은 50px 입니다
  - 하단 패딩은 75px 입니다

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  background-color: lightblue;
  padding: 25px 50px 75px;
}
</style>
</head>
<body>

<h2>The padding shorthand property - 3 values</h2>

<div>This div element has a top padding of 25px, a right and left padding of
50px, and a bottom padding of 75px.</div>

</body>
</html>
```

경우 **padding** 속성은 두 값이있다 :

- **패딩 : 25px 50px;**
  - 상단 및 하단 패딩은 25px 입니다
  - 좌우 패딩은 50px 입니다

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  background-color: lightblue;
  padding: 25px 50px;
}
</style>
</head>
<body>

<h2>The padding shorthand property - 2 values</h2>

<div>This div element has a top and bottom padding of 25px, and a right and left
padding of 50px.</div>

</body>
</html>
```

경우 **padding** 속성은 하나 개의 값을 가지고 :

- **패딩 : 25px;**
  - 네 개의 패딩은 모두 25px 입니다

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  background-color: lightblue;
  padding: 25px;
}
</style>
</head>
<body>

<h2>The padding shorthand property - 1 value</h2>

<div>This div element has a top, bottom, left, and right padding of 25px.</div>

</body>
</html>
```

# 패딩 및 요소 너비

CSS `width` 속성은 요소 내용 영역의 너비를 지정합니다. 내용 영역은 요소의 패딩, 테두리 및 여백 내부의 부분입니다 ( [상자 모델](#) ).

따라서 요소에 지정된 너비가 있으면 해당 요소에 추가 된 패딩이 요소의 총 너비에 추가됩니다. 이것은 종종 바람직하지 않은 결과입니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
  width: 300px;
  background-color: yellow;
}
div.ex2 {
  width: 300px;
  height: 25px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>Padding and element width</h2>

<div class="ex1">This div is 300px wide.</div>
<br>

<div class="ex2">The width of this div is 350px, even though it is defined as
300px in the CSS.</div>

</body>
</html>
```

여기서 `<div>` 요소는 너비가 300px입니다. 그러나 `<div>` 요소의 실제 너비는 350px (300px + 25px의 왼쪽 여백 + 25px의 오른쪽 여백)입니다.

패딩 양에 관계없이 너비를 300px로 유지하려면 `box-sizing` 속성을 사용할 수 있습니다. 이로 인해 요소의 너비가 유지됩니다. 패딩을 늘리면 사용 가능한 콘텐츠 공간이 줄어 듭니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
```

```

width: 300px;
background-color: yellow;
}
div.ex2 {
width: 300px;
padding: 25px;
box-sizing: border-box;
background-color: lightblue;
}
</style>
</head>
<body>

<h2> Padding and element width - with box-sizing</h2>

<div class="ex1">This div is 300px wide.</div>
<br>

<div class="ex2"> The width of this div remains at 300px, in spite of the 50px of
total left and right padding, because of the box-sizing: border-box
property.</div>

</body>
</html>

```

## 모든 CSS 패딩 속성

Property	Description
<a href="#">padding</a>	A shorthand property for setting all the padding properties in one declaration
<a href="#">padding-bottom</a>	Sets the bottom padding of an element
<a href="#">padding-left</a>	Sets the left padding of an element
<a href="#">padding-right</a>	Sets the right padding of an element
<a href="#">padding-top</a>	Sets the top padding of an element

## CSS 높이와 너비

## CSS 높이와 너비 설정

**height** 및 **width** 특성 요소의 높이와 폭을 설정하는 데 사용된다.

높이 및 너비 속성에는 패딩, 테두리 또는 여백이 포함되지 않습니다. 패딩, 테두리 및 요소의 여백 내부 영역의 높이 / 너비를 설정합니다.

# CSS 높이 / 폭 값

`height` 및 `width` 속성은 다음과 같은 값을 가질 수 있습니다 :

- `auto`-이것이 기본값입니다. 브라우저는 높이와 너비를 계산합니다
- `length` -높이, 너비를 px, cm 등으로 정의합니다.
- `%` -포함 블록의 높이 / 폭을 퍼센트로 정의
- `initial` -높이 / 너비를 기본값으로 설정
- `inherit` -높이 / 너비는 부모 값에서 상속됩니다

## CSS 높이 / 너비 예

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}
</style>
</head>
<body>

<h2> Set the height and width of an element</h2>

<p>This div element has a height of 200px and a width of 50%:</p>

<div></div>

</body>
</html>
```

**참고** : 기억 `height`과 `width`특성이 패딩, 테두리, 또는 여백을 포함하지 않습니다! 패딩, 테두리 및 요소의 여백 내부 영역의 높이 / 폭을 설정합니다!

# 최대 너비 설정

이 `max-width` 속성은 요소의 최대 너비를 설정하는 데 사용됩니다.

이 `max-width` 지정 될 수 있는 *길이의 값* (이 기본이다. 수단에는 최대 폭이 없음) 등 PX, cm, 원한다면, 또는 퍼센트 (%)을 함유하는 블록들 또는 없음으로 설정.

위의 문제 `<div>`는 브라우저 창이 요소 너비 (500px)보다 작을 때 발생합니다. 그런 다음 브라우저는 페이지에 가로 스크롤 막대를 추가합니다.

`max-width` 대신이 상황에서 사용 하면 브라우저의 작은 창 처리가 향상됩니다.

**팁 :** 두 div 의 차이점을 보려면 브라우저 창을 500px 미만으로 드래그하십시오!

이 요소의 높이는 100 픽셀이고 최대 너비는 500 픽셀입니다.

**참고 :** `max-width` 속성 값이 재정의 `width` 됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  max-width: 500px;
  height: 100px;
  background-color: powderblue;
}
</style>
</head>
<body>

<h2> Set the max-width of an element </h2>

<p>This div element has a height of 100px and a max-width of 500px:</p>

<div></div>

<p>Resize the browser window to see the effect.</p>

</body>
</html>
```

## 모든 CSS 차원 속성

Property	Description
<a href="#">height</a>	Sets the height of an element

<a href="#">max-height</a>	Sets the maximum height of an element
<a href="#">max-width</a>	Sets the maximum width of an element
<a href="#">min-height</a>	Sets the minimum height of an element
<a href="#">min-width</a>	Sets the minimum width of an element
<a href="#">width</a>	Sets the width of an element

# CSS 박스 모델

## CSS 박스 모델

모든 HTML 요소는 상자로 간주 될 수 있습니다. CSS 에서 "상자 모델"이라는 용어는 디자인과 레이아웃에 대해 이야기 할 때 사용됩니다.

CSS 상자 모델은 본질적으로 모든 HTML 요소를 둘러싸는 상자입니다. 여백, CSS 상자 모델은 본질적으로 모든 HTML 요소를 둘러싸는 상자입니다. 여백, 테두리, 패딩 및 실제 내용으로 구성됩니다. 아래 이미지는 박스 모델을 보여줍니다.

다른 부분에 대한 설명 :

- **내용** -텍스트와 이미지가 나타나는 상자의 내용
- **패딩** -컨텐츠 주변의 영역을 지 웁니다. 패딩은 투명하다
- **테두리** -패딩 및 내용을 둘러싼 테두리
- **여백** -테두리 외부의 영역을 지 웁니다. 여백은 투명하다

상자 모델을 사용하면 요소 주위에 테두리를 추가하고 요소 사이의 공간을 정의 할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>

<h2> Demonstrating the Box Model</h2>

<p> The CSS box model is essentially a box that wraps around every HTML element.
It consists of: borders, padding, margins, and the actual content.</p>
```

```
<div> This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</div>

</body>
</html>
```

## 요소의 너비와 높이

모든 브라우저에서 요소의 너비와 높이를 올바르게 설정하려면 상자 모델의 작동 방식을 알아야 합니다.

**중요 사항 :** CSS 를 사용하여 요소의 너비 및 높이 특성을 설정하는 경우 **컨텐츠 영역** 의 너비 및 높이 만 설정하면 됩니다. 요소의 전체 크기를 계산하려면 안쪽 여백, 테두리 및 여백도 추가해야 합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 320px;
  padding: 10px;
  border: 5px solid gray;
  margin: 0;
}
</style>
</head>
<body>

<h2> Calculate the total width:</h2>


<div>The picture above is 350px wide. The total width of this element is also 350px.</div>

</body>
</html>
```

계산은 다음과 같습니다.

320px (너비)  
+ 20px (왼쪽 + 오른쪽 패딩)



+ 10px (왼쪽 + 오른쪽 테두리)  
+ 0px (왼쪽 + 오른쪽 여백)  
**= 350px**

요소의 총 너비는 다음과 같이 계산해야 합니다.

총 요소 너비 = 너비 + 왼쪽 여백 + 오른쪽 여백 + 왼쪽 테두리 + 오른쪽 테두리  
+ 왼쪽 여백 + 오른쪽 여백

요소의 총 높이는 다음과 같이 계산해야 합니다.

총 요소 높이 = 높이 + 위쪽 패딩 + 아래쪽 패딩 + 위쪽 테두리 + 아래쪽 테두리  
+ 위쪽 여백 + 아래쪽 여백

## CSS 윤곽선

이 요소에는 검은 색 테두리와 10px 너비의 녹색 윤곽선이 있습니다.

## CSS 윤곽선

외곽선은 요소를 "눈에 띄게"만들기 위해 테두리 외부에 그려진 요소입니다.

CSS에는 다음과 같은 개요 속성이 있습니다.

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

**참고 :** 윤곽선은 테두리와 다릅니다! 테두리와 달리 윤곽선은 요소 테두리 외부에 그려지며 다른 내용과 겹칠 수 있습니다. 또한 외곽선은 요소 치수의 일부가 아닙니다. 요소의 총 너비와 높이는 윤곽의 너비에 영향을 받지 않습니다.

# CSS 아웃 라인 스타일

이 `outline-style` 속성은 윤곽선의 스타일을 지정하며 다음 값 중 하나를 가질 수 있습니다.

- `dotted` -점선 윤곽을 정의
- `dashed` -파선 개요를 정의합니다
- `solid` -견고한 윤곽을 정의
- `double` -이중 개요를 정의
- `groove` -3D 그루브 아웃 라인 정의
- `ridge` -3D 능선 윤곽을 정의합니다
- `inset` -3D 삽입 개요를 정의합니다
- `outset` -3D 초기 외곽선을 정의합니다
- `none` -개요를 정의하지 않습니다
- `hidden` -숨겨진 개요를 정의

다음 예제는 다른 `outline-style` 값을 보여 줍니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {outline-color: red;}

p.dotted {outline-style: dotted;}
p.dashed {outline-style: dashed;}
p.solid {outline-style: solid;}
p.double {outline-style: double;}
p.groove {outline-style: groove;}
p.ridge {outline-style: ridge;}
p.inset {outline-style: inset;}
p.outset {outline-style: outset;}
</style>
</head>
<body>

<h2>The outline-style Property </h2>

<p class="dotted"> A dotted outline.</p>
<p class="dashed"> A dashed outline.</p>
<p class="solid"> A solid outline.</p>
<p class="double"> A double outline.</p>
<p class="groove"> A groove outline. The effect depends on the outline-color
value. </p>
<p class="ridge"> A ridge outline. The effect depends on the outline-color
value. </p>
```

```
<p class="inset"> A inset outline. The effect depends on the outline-color value.
</p>
<p class="outset"> A outset outline. The effect depends on the outline-color
value. </p>

</body>
</html>
```

**참고** :`outline-style` 속성을 설정 하지 않으면 다른 외곽선 속성 중 아무 것도 영향을 미치지 않습니다!

## CSS 아웃 라인 색상

이 `outline-color` 속성은 윤곽선의 색을 설정하는 데 사용됩니다.

색상은 다음과 같이 설정할 수 있습니다.

- name- "red"와 같은 색상 이름을 지정하십시오
- RGB- "rgb (255,0,0)"와 같은 RGB 값을 지정하십시오.
- 16 진수- "# ff0000"과 같은 16 진수 값을 지정하십시오.
- 반전-색상 반전을 수행합니다 (색상 배경에 관계없이 윤곽선이 표시되도록 함)

다음 예는 색상이 다른 몇 가지 윤곽선을 보여줍니다. 또한 이 요소들에는 윤곽선 안에 얇은 검은 색 테두리가 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.ex1 {
  border: 1px solid black;
  outline-style: solid;
  outline-color: red;
}

p.ex2 {
  border: 1px solid black;
  outline-style: double;
  outline-color: green;
}

p.ex3 {
  border: 1px solid black;
  outline-style: outset;
  outline-color: yellow;
}
</style>
```

```

</head>
<body>

<h2>The outline-color Property</h2>

<p class="ex1"> A solid red outline. </p>
<p class="ex2"> A double green outline. </p>
<p class="ex3"> An outset yellow outline. </p>

</body>
</html>

```

다음 예제에서는 색상 반전을 수행하는 `outline-color: invert`을 사용합니다. 이렇게 하면 색상 배경에 관계없이 윤곽선이 표시됩니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
p.ex1 {
  border: 1px solid yellow;
  outline-style: solid;
  outline-color: invert;
}
</style>
</head>
<body>

<h2>Using outline-color:invert</h2>

<p class="ex1"> A solid invert outline.</p>

</body>
</html>

```

## CSS 윤곽선 너비

이 `outline-width` 속성은 윤곽선의 너비를 지정하며 다음 값 중 하나를 가질 수 있습니다.

- 얇음 (일반적으로 1px)
- 중간 (일반적으로 3px)
- 두꺼운 (일반적으로 5px)
- 특정 크기 (px, pt, cm, em 등)

다음은 너비가 다른 윤곽선을 보여주는 예입니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
p.ex1 {
  border: 1px solid black;
  outline-style: solid;
  outline-color: red;
  outline-width: thin;
}

p.ex2 {
  border: 1px solid black;
  outline-style: solid;
  outline-color: red;
  outline-width: medium;
}

p.ex3 {
  border: 1px solid black;
  outline-style: solid;
  outline-color: red;
  outline-width: thick;
}

p.ex4 {
  border: 1px solid black;
  outline-style: solid;
  outline-color: red;
  outline-width: 4px;
}
</style>
</head>
<body>

<h2>The outline-width Property</h2>

<p class="ex1"> A thin outline.</p>
<p class="ex2"> A medium outline.</p>
<p class="ex3"> A thick outline.</p>
<p class="ex4"> A 4px thick outline.</p>

</body>
</html>

```

## CSS 아웃 라인-속기 속성

이 **outline** 속성은 다음과 같은 개별 개요 속성을 설정하기 위한 속기 속성입니다.

- **outline-width**
- **outline-style** (필수)
- **outline-color**

이 **outline** 특성은 위 목록에서 하나, 둘 또는 세 개의 값으로 지정됩니다. 값의 순서는 중요하지 않습니다.

다음 예제는 속기 **outline** 속성으로 지정된 개요를 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.ex1 {outline: dashed;}
p.ex2 {outline: dotted red;}
p.ex3 {outline: 5px solid yellow;}
p.ex4 {outline: thick ridge pink;}
</style>
</head>
<body>

<h2>The outline-width Property</h2>

<p class="ex1"> A dashed outline.</p>
<p class="ex2"> A dotted red outline.</p>
<p class="ex3"> A 5px solid yellow outline.</p>
<p class="ex4"> A thick ridge pink outline.</p>

</body>
</html>
```

## CSS 아웃 라인 오프셋

이 **outline-offset** 속성은 윤곽선과 요소의 가장자리 / 테두리 사이에 공간을 추가합니다. 요소와 윤곽선 사이의 공간은 투명합니다.

다음 예제는 테두리 가장자리 외부에서 **15px**의 윤곽선을 지정합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  margin: 30px;
  border: 1px solid black;
  outline: 1px solid red;
  outline-offset: 15px;
}
</style>
</head>
<body>

<h2>The outline-offset Property</h2>

<p>This paragraph has an outline 15px outside the border edge.</p>
```

```
</body>
</html>
```

다음 예제는 요소와 윤곽선 사이의 공간이 투명 함을 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  margin: 30px;
  background: yellow;
  border: 1px solid black;
  outline: 1px solid red;
  outline-offset: 15px;
}
</style>
</head>
<body>

<h2>The outline-offset Property</h2>

<p>This paragraph has an outline 15px outside the border edge.</p>

</body>
</html>
```

## CSS 텍스트

### 텍스트 서식

이 텍스트는 일부 텍스트 서식 속성으로 스타일이 지정됩니다. 제목에는 텍스트 정렬, 텍스트 변형 및 색상 속성이 사용됩니다. 단락이 들여쓰기 되고 정렬되며 문자 사이의 공백이 지정됩니다. 밑줄이 "색칠하기" 링크에서 제거됩니다.

### 글자 색

이 **color** 속성은 텍스트 색상을 설정하는 데 사용됩니다. 색상은 다음과 같이 지정됩니다.

- "빨간색"과 같은 색상 이름
- "# ff0000"과 같은 16 진수 값
- "rgb (255,0,0)"와 같은 RGB 값

보 [CSS 색상 값](#) 가능한 색상 값의 전체 목록.

페이지의 기본 텍스트 색상은 **body** 선택기에 정의되어 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  color: blue;
}

h1 {
  color: green;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>

<p>This is an ordinary paragraph. Notice that this text is blue. The default text
color for a page is defined in the body selector.</p>

</body>
</html>
```

**참고** : W3C 호환 CSS의 경우 : **color** 속성 을 정의하는 경우 **background-color** 을 정의해야 합니다.

## 텍스트 정렬

이 **text-align** 속성은 텍스트의 가로 정렬을 설정하는 데 사용됩니다.

텍스트는 왼쪽 또는 오른쪽 정렬, 가운데 정렬 또는 정렬 할 수 있습니다.

다음 예제는 가운데 정렬 및 왼쪽 및 오른쪽 정렬 텍스트를 보여줍니다 (텍스트 방향이 왼쪽에서 오른쪽 인 경우 왼쪽 정렬이 기본값이고 텍스트 방향이 오른쪽에서 왼쪽 인 경우 오른쪽 정렬이 기본값 임).

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



```

<head>
<style>
h1 {
  text-align: center;
}

h2 {
  text-align: left;
}

h3 {
  text-align: right;
}
</style>
</head>
<body>

<h1>Heading 1 (center)</h1>
<h2>Heading 2 (left)</h2>
<h3>Heading 3 (right)</h3>

<p>The three headings above are aligned center, left and right.</p>

</body>
</html>

```

**text-align** 속성이 "justify"로 설정 되면 모든 줄의 너비가 같고 잡지 및 신문과 같이 왼쪽과 오른쪽 여백이 똑 바르도록 각 줄이 늘어납니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  padding: 10px;
  width: 200px;
  height: 200px;
  text-align: justify;
}
</style>
</head>
<body>

<h1> Example text-align: justify;</h1>

<p>The text-align: justify; value stretches the lines so that each line has equal width (like in newspapers and magazines).</p>

<div>In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'</div>

</body>

```

```
</html>
```

## 텍스트 장식

이 `text-decoration` 속성은 텍스트에서 장식을 설정하거나 제거하는 데 사용됩니다.

이 값 `text-decoration: none;`은 종종 링크에서 밑줄을 제거하는 데 사용됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
a {
  text-decoration: none;
}
</style>
</head>
<body>

<p>A link with no underline:
  <a href="http://www.w3schools.com">W3Schools.com</a>
</p>

</body>
</html>
```

다른 `text-decoration` 값은 텍스트를 장식하는 데 사용됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-decoration: overline;
}

h2 {
  text-decoration: line-through;
}

h3 {
  text-decoration: underline;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
```

```
<h3>This is heading 3</h3>
```

```
</body>  
</html>
```

**참고** : 링크가 아닌 텍스트에는 밑줄을 긋지 않는 것이 좋습니다 . 이는 종종 독자를 혼란스럽게 합니다.

## 텍스트 변환

이 **text-transform** 속성은 텍스트에서 대문자와 소문자를 지정하는 데 사용됩니다.

모든 것을 대문자 또는 소문자로 바꾸거나 각 단어의 첫 글자를 대문자로 쓰는 데 사용할 수 있습니다.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.uppercase {  
  text-transform: uppercase;  
}  
  
p.lowercase {  
  text-transform: lowercase;  
}  
  
p.capitalize {  
  text-transform: capitalize;  
}  
</style>  
</head>  
<body>  
  
<p class="uppercase">This is some text.</p>  
<p class="lowercase">This is some text.</p>  
<p class="capitalize">This is some text.</p>  
  
</body>  
</html>
```

## 텍스트 들여 쓰기

이 **text-indent** 속성은 텍스트의 첫 줄 들여 쓰기를 지정하는 데 사용됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  text-indent: 50px;
}
</style>
</head>
<body>

<p>In my younger and more vulnerable years my father gave me some advice that
I've been turning over in my mind ever since. 'Whenever you feel like criticizing
anyone,' he told me, 'just remember that all the people in this world haven't had
the advantages that you've had.'</p>

</body>
</html>
```

## 문자 간격

이 **letter-spacing** 속성은 텍스트에서 문자 사이의 간격을 지정하는 데 사용됩니다.

다음 예제는 문자 사이의 간격을 늘리거나 줄이는 방법을 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  letter-spacing: 3px;
}

h2 {
  letter-spacing: -3px;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>

</body>
</html>
```

# 선 높이

이 `line-height` 속성은 줄 사이의 간격을 지정하는 데 사용됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.small {
  line-height: 0.8;
}

p.big {
  line-height: 1.8;
}
</style>
</head>
<body>

<p>
This is a paragraph with a standard line-height.<br>
The default line height in most browsers is about 110% to 120%.<br>
</p>

<p class="small">
This is a paragraph with a smaller line-height.<br>
This is a paragraph with a smaller line-height.<br>
</p>

<p class="big">
This is a paragraph with a bigger line-height.<br>
This is a paragraph with a bigger line-height.<br>
</p>

</body>
</html>
```

# 텍스트 방향

이 `direction` 속성은 요소의 텍스트 방향을 변경하는 데 사용됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.ex1 {
  direction: rtl;
}
</style>
</head>
```

```
<body>

<p>This is the default text direction.</p>

<p class="ex1"><bdo dir="rtl">This is right-to-left text direction.</bdo></p>

</body>
</html>
```

## 단어 간격

이 `word-spacing` 속성은 텍스트에서 단어 사이의 간격을 지정하는 데 사용됩니다.

다음 예제는 단어 사이의 간격을 늘리거나 줄이는 방법을 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  word-spacing: 10px;
}

h2 {
  word-spacing: -5px;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>

</body>
</html>
```

## 텍스트 그림자

이 `text-shadow` 속성은 텍스트에 그림자를 추가합니다.

다음 예제에서는 가로 그림자의 위치 (3px), 세로 그림자의 위치 (2px) 및 그림자의 색상 (빨간색)을 지정합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```

h1 {
  text-shadow: 3px 2px red;
}
</style>
</head>
<body>

<h1>Text-shadow effect</h1>
<p><b>Note:</b>Internet Explorer 9 and earlier do not support the text-shadow
property.</p>

</body>
</html>

```

## 모든 CSS 텍스트 속성

Property	Description
<a href="#">color</a>	Sets the color of text
<a href="#">direction</a>	Specifies the text direction/writing direction
<a href="#">letter-spacing</a>	Increases or decreases the space between characters in a text
<a href="#">line-height</a>	Sets the line height
<a href="#">text-align</a>	Specifies the horizontal alignment of text
<a href="#">text-decoration</a>	Specifies the decoration added to text
<a href="#">text-indent</a>	Specifies the indentation of the first line in a text-block
<a href="#">text-shadow</a>	Specifies the shadow effect added to text
<a href="#">text-transform</a>	Controls the capitalization of text
<a href="#">text-overflow</a>	Specifies how overflowed content that is not displayed should be signaled to the user
<a href="#">unicode-bidi</a>	Used together with the <a href="#">direction</a> property to set or return whether the text should be overridden to support multiple languages in the same document
<a href="#">vertical-align</a>	Sets the vertical alignment of an element
<a href="#">white-space</a>	Specifies how white-space inside an element is handled
<a href="#">word-spacing</a>	Increases or decreases the space between words in a text

## CSS 글꼴

CSS 글꼴 속성은 글꼴 모음, 굵게, 크기 및 텍스트 스타일을 정의합니다.

## Serif와 Sans-serif 글꼴의 차이점



## CSS 폰트 패밀리

CSS에는 두 가지 유형의 글꼴 모음이 있습니다.

- **일반 패밀리** - 모양이 비슷한 글꼴 패밀리 그룹 (예 : "Serif" 또는 "Monospace")
- **글꼴 모음** - 특정 글꼴 모음 (예 : "Times New Roman" 또는 "Arial")

Generic family	Font family	Description
Serif	Times New Roman Georgia	Serif fonts have small lines at the ends on some characters
Sans-serif	Arial Verdana	"Sans" means without - these fonts do not have the lines at the ends of characters
Monospace	Courier New Lucida Console	All monospace characters have the same width

**참고 :** 컴퓨터 화면에서 sans-serif 글꼴은 serif 글꼴보다 읽기 쉬운 것으로 간주됩니다.

## 폰트 패밀리

텍스트의 글꼴 군은 `font-family` 속성으로 설정됩니다 .



이 **font-family** 속성에는 "대체"시스템으로 여러 글꼴 이름이 있어야합니다. 브라우저가 첫 번째 글꼴을 지원하지 않으면 다음 글꼴 등을 시도합니다.

다른 글꼴을 사용할 수 없는 경우 브라우저가 일반 패밀리에서 유사한 글꼴을 선택할 수 있도록 원하는 글꼴로 시작하고 일반 패밀리로 끝냅니다.

**참고 :** 글꼴 모음 이름이 두 단어 이상인 경우 "Times New Roman"과 같이 따옴표로 묶어야 합니다.

첨표로 구분 된 목록에 둘 이상의 글꼴 패밀리가 지정되었습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.serif {
  font-family: "Times New Roman", Times, serif;
}
p.sansserif {
  font-family: Arial, Helvetica, sans-serif;
}
</style>
</head>
<body>

<h1>CSS font-family</h1>
<p class="serif">This is a paragraph, shown in the Times New Roman font.</p>
<p class="sansserif">This is a paragraph, shown in the Arial font.</p>

</body>
</html>
```

일반적으로 사용되는 글꼴 조합은 [웹 안전 글꼴 조합을 참조하십시오](#) .

## 폰트 스타일

이 **font-style** 속성은 주로 기울임 꼴 텍스트를 지정하는 데 사용됩니다.

이 속성에는 세 가지 값이 있습니다.

- normal-텍스트가 정상적으로 표시됩니다
- italic-텍스트가 이탤릭체로 표시됩니다
- oblique (사위)-텍스트가 "기울입니다"(oblique 는 기울임 꼴과 매우 유사하지만 덜 지원됨)

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
  font-style: normal;
}

p.italic {
  font-style: italic;
}

p.oblique {
  font-style: oblique;
}
</style>
</head>
<body>

<p class="normal">This is a paragraph in normal style.</p>
<p class="italic">This is a paragraph in italic style.</p>
<p class="oblique">This is a paragraph in oblique style.</p>

</body>
</html>
```

## 폰트 크기

이 `font-size` 속성은 텍스트 크기를 설정합니다.

웹 디자인에서 텍스트 크기를 관리 할 수 있어야 합니다. 그러나 단락을 제목처럼 보이거나 제목을 단락처럼 보이도록 글꼴 크기 조정을 사용해서는 안됩니다.

제목에는 `<h1>`-`<h6>`, 단락에는 `<p>`와 같은 올바른 HTML 태그를 항상 사용하십시오.

글꼴 크기 값은 절대 또는 상대 크기 일 수 있습니다.

절대 크기 :

- 텍스트를 지정된 크기로 설정합니다
- 사용자가 모든 브라우저에서 텍스트 크기를 변경할 수 없음 (접근성 이유로 인해 나쁨)
- 절대 크기는 출력의 실제 크기를 알 때 유용합니다

상대 크기 :

- 주변 요소를 기준으로 크기를 설정합니다
- 사용자가 브라우저에서 텍스트 크기를 변경할 수 있습니다

**참고 :** 글꼴 크기를 지정하지 않으면 단락과 같은 일반 텍스트의 기본 크기는 16px (16px = 1em)입니다.

## 픽셀로 글꼴 크기 설정

텍스트 크기를 픽셀로 설정하면 텍스트 크기를 완전히 제어 할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  font-size: 40px;
}

h2 {
  font-size: 30px;
}

p {
  font-size: 14px;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

**팁 :** 픽셀을 사용하는 경우에도 확대 도구를 사용하여 전체 페이지의 크기를 조정할 수 있습니다.

# Em으로 글꼴 크기 설정

사용자가 브라우저 메뉴에서 텍스트 크기를 조정할 수 있도록 많은 개발자가 픽셀 대신 **em** 을 사용합니다.

**em** 크기 단위는 W3C 에서 권장합니다.

**1em** 은 현재 글꼴 크기와 같습니다. 브라우저의 기본 텍스트 크기는 **16px** 입니다. 따라서 **1em** 의 기본 크기는 **16px** 입니다.

다음 공식을 사용하여 픽셀에서 **em** 까지 크기를 계산할 수 있습니다.  $pixels / 16 = em$

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  font-size: 2.5em; /* 40px/16=2.5em */
}
h2 {
  font-size: 1.875em; /* 30px/16=1.875em */
}
p {
  font-size: 0.875em; /* 14px/16=0.875em */
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>Specifying the font-size in em allows all major browsers to resize the text.
Unfortunately, there is still a problem with older versions of IE. When resizing
the text, it becomes larger/smaller than it should.</p>

</body>
</html>
```

위의 예에서 **em** 의 텍스트 크기는 이전 예와 동일합니다 (픽셀 단위). 그러나 **em** 크기를 사용하면 모든 브라우저에서 텍스트 크기를 조정할 수 있습니다.

불행히도 이전 버전의 **IE** 에는 여전히 문제가 있습니다. 텍스트는 크게 만들 때 보다 커지고 작아 질 때 보다 작아집니다.

# 퍼센트와 Em의 조합 사용

모든 브라우저에서 작동하는 솔루션은 `<body>` 요소의 기본 글꼴 크기를 백분율로 설정하는 것입니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  font-size: 100%;
}

h1 {
  font-size: 2.5em;
}

h2 {
  font-size: 1.875em;
}

p {
  font-size: 0.875em;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>Specifying the font-size in percent and em displays the same size in all major
browsers, and allows all browsers to resize the text!</p>

</body>
</html>
```

우리의 코드는 이제 훌륭하게 작동합니다! 모든 브라우저에서 동일한 텍스트 크기를 표시하며 모든 브라우저가 텍스트를 확대하거나 크기를 조정할 수 있습니다!

## 폰트 무게

이 `font-weight` 속성은 글꼴의 무게를 지정합니다.

```
<!DOCTYPE html>
<html>
<head>
```

```

<style>
p.normal {
  font-weight: normal;
}
p.light {
  font-weight: lighter;
}
p.thick {
  font-weight: bold;
}
p.thicker {
  font-weight: 900;
}
</style>
</head>
<body>

<p class="normal">This is a paragraph.</p>
<p class="light">This is a paragraph.</p>
<p class="thick">This is a paragraph.</p>
<p class="thicker">This is a paragraph.</p>

</body>
</html>

```

## 반응 형 글꼴 크기

텍스트 크기는 **vw** "뷰포트 너비"를 의미 하는 단위 로 설정할 수 있습니다.

그렇게 하면 텍스트 크기가 브라우저 창의 크기를 따릅니다.

```

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>

<h1 style="font-size:10vw"> Responsive Text</h1>
<p style="font-size:5vw">Resize the browser window to see how the text size
scales.</p>
<p style="font-size:5vw">Use the "vw" unit when sizing the text. 10vw will set
the size to 10% of the viewport width.</p>
<p>Viewport is the browser window size. 1vw = 1% of viewport width. If the
viewport is 50cm wide, 1vw is 0.5cm.</p>

</body>
</html>

```

뷰포트는 브라우저 창 크기입니다. **1vw** = 뷰포트 너비의 1 % 뷰포트의 너비가 50cm 인 경우 1vw 는 0.5cm 입니다.

# 글꼴 변형

이 `font-variant` 속성은 텍스트를 작은 대문자로 표시할지 여부를 지정합니다.

작은 대문자 글꼴에서는 모든 소문자가 대문자로 변환됩니다. 그러나 변환 된 대문자는 텍스트의 원래 대문자보다 작은 글꼴 크기로 나타납니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
  font-variant: normal;
}

p.small {
  font-variant: small-caps;
}
</style>
</head>
<body>

<p class="normal">My name is Hege Refsnes.</p>
<p class="small">My name is Hege Refsnes.</p>

</body>
</html>
```

## CSS 아이콘

### 아이콘을 추가하는 방법

HTML 페이지에 아이콘을 추가하는 가장 간단한 방법은 Font Awesome 과 같은 아이콘 라이브러리를 사용하는 것입니다.

지정된 아이콘 클래스의 이름을 인라인 HTML 요소 (예 : `<i>` 또는 `<span>`)에 추가하십시오.

아래 아이콘 라이브러리의 모든 아이콘은 CSS (크기, 색상, 그림자 등)로 사용자 정의 할 수 있는 확장 가능한 벡터입니다.

# 글꼴 멋진 아이콘

Font Awesome 아이콘을 사용하려면 [fontawesome.com](https://fontawesome.com) 으로 이동하여 로그인한 `<head>` 다음 HTML 페이지 섹션에 추가 할 코드를 얻습니다.

```
<script src="https://kit.fontawesome.com/yourcode.js"></script>
```

[Font Awesome 5 튜토리얼](#) 에서 Font Awesome 을 시작하는 방법에 대해 자세히 알아보십시오.

**참고 :** 다운로드 또는 설치가 필요하지 않습니다!

```
<!DOCTYPE html>
<html>
<head>
<title>Font Awesome Icons</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<script src="https://kit.fontawesome.com/a076d05399.js"></script>
<!-- Get your own code at fontawesome.com-->
</head>
<body>

<p>Some Font Awesome Icons:</p>
<i class="fas fa-cloud"></i>
<i class="fas fa-heart"></i>
<i class="fas fa-car"></i>
<i class="fas fa-file"></i>
<i class="fas fa-bars"></i>

<p>Styled Font Awesome icons (size and color):</p>
<i class="fas fa-cloud" style="font-size:24px;"></i>
<i class="fas fa-cloud" style="font-size:36px;"></i>
<i class="fas fa-cloud" style="font-size:48px;color:red;"></i>
<i class="fas fa-cloud" style="font-size:60px;color:lightblue;"></i>

</body>
</html>
```

모든 Font Awesome 아이콘에 대한 자세한 내용을 보려면 [Icon Reference](#)를 방문하십시오 .

# 부트 스트랩 아이콘

부트 스트랩 글리프 콘을 사용하려면 `<head>` HTML 페이지 섹션 안에 다음 줄을 추가 하십시오.



```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

참고 : 다운로드 또는 설치가 필요하지 않습니다!

```
<!DOCTYPE html>
<html>
<head>
<title>Bootstrap Icons</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>
<body class="container">

<p>Some Bootstrap Icons:</p>
<i class="glyphicon glyphicon-cloud"></i>
<i class="glyphicon glyphicon-remove"></i>
<i class="glyphicon glyphicon-user"></i>
<i class="glyphicon glyphicon-envelope"></i>
<i class="glyphicon glyphicon-thumbs-up"></i>
<br><br>

<p>Styled Bootstrap icons (size and color):</p>
<i class="glyphicon glyphicon-cloud" style="font-size:24px;"></i>
<i class="glyphicon glyphicon-cloud" style="font-size:36px;"></i>
<i class="glyphicon glyphicon-cloud" style="font-size:48px;color:red;"></i>
<i class="glyphicon glyphicon-cloud" style="font-size:60px;color:lightblue;"></i>

</body>
</html>
```

## 구글 아이콘

Google 아이콘을 사용하려면 **<head>** HTML 페이지 섹션 안에 다음 줄을 추가 하십시오.

```
<link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">
```

참고 : 다운로드 또는 설치가 필요하지 않습니다!

```
<!DOCTYPE html>
<html>
<head>
<title>Google Icons</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
</head>
```

```

</head>
<body>

<p>Some Google Icons:</p>
<i class="material-icons">cloud</i>
<i class="material-icons">favorite</i>
<i class="material-icons">attachment</i>
<i class="material-icons">computer</i>
<i class="material-icons">traffic</i>
<br><br>

<p>Styled Google icons (size and color):</p>
<i class="material-icons" style="font-size:24px;">cloud</i>
<i class="material-icons" style="font-size:36px;">cloud</i>
<i class="material-icons" style="font-size:48px;color:red;">cloud</i>
<i class="material-icons" style="font-size:60px;color:lightblue;">cloud</i>

</body>
</html>

```

## CSS 링크

CSS 를 사용하면 링크를 다양한 방식으로 스타일링 할 수 있습니다.

[텍스트 링크](#)
[텍스트 링크](#)
[링크 버튼](#)
[링크 버튼](#)

## 스타일링 링크

링크는 어떤 CSS 속성 (예와 스타일을 지정할 수 있습니다 **color**, **font-family**, **background**, 등).

```

<!DOCTYPE html>
<html>
<head>
<style>
a {
  color: hotpink;
}
</style>
</head>
<body>

<p><b><a href="#" target="_blank"> This is a link</a></b></p>

</body>
</html>

```

또한 링크의 **상태** 에 따라 링크 스타일을 다르게 지정할 수 있습니다.

네 가지 링크 상태는 다음과 같습니다.

- **a:link** -방문하지 않은 정상적인 링크
- **a:visited** -사용자가 방문한 링크
- **a:hover** -사용자가 마우스를 가져 가면 링크
- **a:active** -클릭 순간의 링크

```
<!DOCTYPE html>
<html>
<head>
<style>
/* unvisited link */
a:link {
  color: red;
}

/* visited link */
a:visited {
  color: green;
}

/* mouse over link */
a:hover {
  color: hotpink;
}

/* selected link */
a:active {
  color: blue;
}
</style>
</head>
<body>

<p><b><a href="#" target="_blank"> This is a link</a></b></p>
<p><b>Note:</b>a:hover MUST come after a:link and a:visited in the CSS definition
in order to be effective.</p>
<p><b>Note:</b>a:active MUST come after a:hover in the CSS definition in order to
be effective.</p>

</body>
</html>
```

여러 링크 상태에 대한 스타일을 설정할 때 몇 가지 순서 규칙이 있습니다.

- **a : 호버**는 반드시 **a : link** 와 **a : visited** 뒤에 와야 합니다.
- **a : active** 는 반드시 다음에 와야 합니다.

# 텍스트 장식

이 `text-decoration` 속성은 주로 링크에서 밑줄을 제거하는 데 사용됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link {
  text-decoration: none;
}

a:visited {
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
}

a:active {
  text-decoration: underline;
}
</style>
</head>
<body>

<p><b><a href="#" target="_blank"> This is a link</a></b></p>
<p><b>Note:</b>a:hover MUST come after a:link and a:visited in the CSS definition
in order to be effective.</p>
<p><b>Note:</b>a:active MUST come after a:hover in the CSS definition in order to
be effective.</p>

</body>
</html>
```

# 배경색

이 `background-color` 속성을 사용하여 링크의 배경색을 지정할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link {
  background-color: yellow;
}

a:visited {
```

```

background-color: cyan;
}

a:hover {
background-color: lightgreen;
}

a:active {
background-color: hotpink;
}
</style>
</head>
<body>

<p><b><a href="#" target="_blank"> This is a link</a></b></p>
<p><b>Note:</b>a:hover MUST come after a:link and a:visited in the CSS definition
in order to be effective.</p>
<p><b>Note:</b>a:active MUST come after a:hover in the CSS definition in order to
be effective.</p>

</body>
</html>

```

## 고급-링크 버튼

이 예는 여러 CSS 속성을 결합하여 링크를 상자 / 단추로 표시하는 고급 예를 보여줍니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
a:link, a:visited {
background-color: #f44336;
color: white;
padding: 14px 25px;
text-align: center;
text-decoration: none;
display: inline-block;
}

a:hover, a:active {
background-color: red;
}
</style>
</head>
<body>

<h2>Link Button</h2>
<p>A link styled as a button:</p>
<a href="#" target="_blank"> This is a link</a>

```

```
</body>  
</html>
```

## CSS 목록

### 정렬되지 않은 목록 :

- 커피
- 차
- 코카콜라
  
- 커피
- 차
- 코카콜라

### 정렬 된 목록 :

1. 커피
  2. 차
  3. 코카콜라
- 
- I. 커피
  - II. 차
  - III. 코카콜라

## HTML 목록 및 CSS 목록 속성

HTML에는 두 가지 주요 유형의 목록이 있습니다.

- 순서가 없는 목록 (<ul>)-목록 항목에 글 머리 기호가 표시됩니다
- 정렬 된 목록 (<ol>)-목록 항목에 숫자 나 문자가 표시되어 있습니다

CSS 목록 속성을 통해 다음을 수행 할 수 있습니다.

- 순서가 지정된 목록에 대해 다른 목록 항목 마커 설정
- 정렬되지 않은 목록에 대해 다른 목록 항목 마커 설정
- 이미지를 목록 항목 마커로 설정
- 목록 및 목록 항목에 배경색 추가

## 다른 목록 항목 마커

이 `list-style-type` 속성은 목록 항목 마커의 유형을 지정합니다.

다음 예제는 사용 가능한 일부 목록 항목 마커를 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
ul.a {
  list-style-type: circle;
}

ul.b {
  list-style-type: square;
}

ol.c {
  list-style-type: upper-roman;
}

ol.d {
  list-style-type: lower-alpha;
}
</style>
</head>
<body>

<p>Example of unordered lists:</p>
<ul class="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

<ul class="b">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```

```

<p>Example of ordered lists:</p>
<ol class="c">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>

<ol class="d">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>

</body>
</html>

```

참고 : 일부 값은 정렬되지 않은 목록에 대한 것이고 일부는 정렬 된 목록에 대한 것입니다.

## 목록 항목 마커로서의 이미지

이 `list-style-image` 속성은 이미지를 목록 항목 마커로 지정합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-image: url('sqpurple.gif');
}
</style>
</head>
<body>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

</body>
</html>

```



# 목록 항목 마커 배치

이 `list-style-position` 속성은 목록 항목 마커 (글 머리 기호)의 위치를 지정합니다.

"목록 스타일 위치 : 외부;" 글 머리 기호가 목록 항목 외부에 있음을 의미합니다. 목록 항목의 각 줄의 시작은 세로로 정렬됩니다. 이것이 기본값 입니다 :

- |                                |
|--------------------------------|
| • 커피- 볶은 커피 원두에서 준비된 양조 음료 ... |
| • 차                            |
| • 코카콜라                         |

"목록 스타일 위치 : 내부;" 글 머리 기호가 목록 항목 안에 있음을 의미합니다. 목록 항목의 일부이므로 텍스트의 일부가 되고 시작 시 텍스트를 푸시합니다.

- |                                |
|--------------------------------|
| • 커피- 볶은 커피 원두에서 준비된 양조 음료 ... |
| • 차                            |
| • 코카콜라                         |

```
<!DOCTYPE html>
<html>
<head>
<style>
ul.a {
  list-style-position: outside;
}

ul.b {
  list-style-position: inside;
}
</style>
</head>
<body>

<h1>The list-style-position Property</h1>
<h2>list-style-position: outside (default):</h2>
<ul class="a">
  <li>Coffee - A brewed drink prepared from roasted coffee beans, which are the
seeds of berries from the Coffea plant</li>
  <li>Tea - An aromatic beverage commonly prepared by pouring hot or boiling water
over cured leaves of the Camellia sinensis, an evergreen shrub (bush) native to
Asia</li>
  <li>Coca Cola - A carbonated soft drink produced by The Coca-Cola Company. The
drink's name refers to two of its original ingredients, which were kola nuts (a
```

```

source of caffeine) and coca leaves</li>
</ul>

<h2>list-style-position: inside:</h2>
<ul class="b">
  <li>Coffee - A brewed drink prepared from roasted coffee beans, which are the
  seeds of berries from the Coffea plant</li>
  <li>Tea - An aromatic beverage commonly prepared by pouring hot or boiling water
  over cured leaves of the Camellia sinensis, an evergreen shrub (bush) native to
  Asia</li>
  <li>Coca Cola - A carbonated soft drink produced by The Coca-Cola Company. The
  drink's name refers to two of its original ingredients, which were kola nuts (a
  source of caffeine) and coca leaves</li>
</ul>

</body>
</html>

```

## 기본 설정 제거

이 `list-style-type:none` 속성을 사용하여 마커 / 글 머리 기호를 제거 할 수도 있습니다. 목록에는 기본 여백과 패딩도 있습니다. 이를 제거하려면 `<ul>` 또는 `<ol>` 에 `margin:0` and `padding:0` 를 추가하십시오 .

```

<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}
</style>
</head>
<body>

<p>Default list:</p>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

<p>Remove bullets, margin and padding:</p>
<ul class="demo">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

</body>

```

</html>

## 리스트-속기 속성

이 **list-style** 속성은 속기 속성입니다. 하나의 선언으로 모든 목록 속성을 설정하는 데 사용됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style: square inside url("sqpurple.gif");
}
</style>
</head>
<body>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

</body>
</html>
```

속기 속성을 사용할 때 속성 값의 순서는 다음과 같습니다.

- **list-style-type** (목록 스타일 이미지가 지정된 경우, 어떤 이유로 이미지를 표시 할 수 없는 경우 이 특성의 값이 표시됩니다)
- **list-style-position** (목록 항목 마커가 콘텐츠 플로우 내부 또는 외부에 표시되어야 하는지 지정)
- **list-style-image** (이미지를 목록 항목 마커로 지정)

위의 속성 값 중 하나가 누락 된 경우 누락 된 속성의 기본값이 있으면 삽입됩니다.

## 색상 스타일리스트

또한 좀 더 흥미롭게 보이도록 색상으로 목록 스타일을 지정할 수도 있습니다.

<ol> 또는 <ul> 태그에 추가 된 것은 전체 목록에 영향을 미치며 <li> 태그에 추가 된 속성은 개별 목록 항목에 영향을 미칩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
ol {
  background: #ff9999;
  padding: 20px;
}

ul {
  background: #3399ff;
  padding: 20px;
}

ol li {
  background: #ffe5e5;
  padding: 5px;
  margin-left: 35px;
}

ul li {
  background: #cce5ff;
  margin: 5px;
}
</style>
</head>
<body>

<h1>Styling Lists With Colors:</h1>

<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

</body>
</html>
```

## 모든 CSS 목록 속성

Property	Description
<a href="#">list-style</a>	Sets all the properties for a list in one declaration
<a href="#">list-style-image</a>	Specifies an image as the list-item marker

<a href="#">list-style-position</a>	Specifies the position of the list-item markers (bullet points)
<a href="#">list-style-type</a>	Specifies the type of list-item marker

# CSS 테이블

CSS를 사용하면 HTML 테이블의 모양을 크게 향상시킬 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
#customers {
  font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
  border-collapse: collapse;
  width: 100%;
}

#customers td, #customer th {
  border: 1px solid #ddd;
  padding: 8px;
}

#customers tr:nth-child(even) {
  background-color: #f2f2f2;
}

#customers tr:hover {
  background-color: #ddd;
}

#customers th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #4caf50;
  color: white;
}
</style>
</head>
<body>

<table id="customers">
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futtarkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Berglunds snabbköp</td>
    <td>Christina Berglund</td>
```

```

        <td>Sweden</td>
    </tr>
    <tr>
        <td>Centro comercial Moctezuma</td>
        <td>Francisco Chang</td>
        <td>Mexico</td>
    </tr>
    <tr>
        <td>Ernst Handel</td>
        <td>Roland Mendel</td>
        <td>Austria</td>
    </tr>
    <tr>
        <td>Island Trading</td>
        <td>Helen Bennett</td>
        <td>UK</td>
    </tr>
    <tr>
        <td>Königlich Essen</td>
        <td>Philip Cramer</td>
        <td>Germany</td>
    </tr>
    <tr>
        <td>Laughing Bacchus Winecellars</td>
        <td>Yoshi Tannamuri</td>
        <td>Canada</td>
    </tr>
    <tr>
        <td>Magazzini Alimentari Riuniti</td>
        <td>Giovanni Rovelli</td>
        <td>Italy</td>
    </tr>
    <tr>
        <td>North/South</td>
        <td>Simon Crowther</td>
        <td>UK</td>
    </tr>
    <tr>
        <td>Paris spécialités</td>
        <td>Marie Bertrand</td>
        <td>France</td>
    </tr>
</table>

</body>
</html>

```

## 테이블 테두리

CSS 에서 표 테두리를 지정하려면 **border** 속성을 사용하십시오.

아래 예제는 <table>, <th> 및 <td> 요소에 검은 색 테두리를 지정합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
}
</style>
</head>
<body>

<h2>Add a border to a table:</h2>

<table id="customers">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
  </tr>
</table>

</body>
</html>

```

위 예의 표에는 이중 테두리가 있습니다. 이는 테이블과 <th> 및 <td> 요소에 별도의 경계가 있기 때문입니다.

## 테이블 테두리 축소

이 **border-collapse** 속성은 테이블 테두리를 단일 테두리로 축소할지 여부를 설정합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
table {
  border-collapse: collapse;
}

table, th, td {
  border: 1px solid black;
}
</style>

```

```

</head>
<body>

<h2> Let the borders collapse:</h2>

<table id="customers">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
  </tr>
</table>

<p><b>Note:</b>If a !DOCTYPE is not specified, the border-collapse property can
produce unexpected results in IE8 and earlier versions.</p>

</body>
</html>

```

테이블 주위에 테두리 만 원하는 경우 <table> **border** 의 속성 만 지정하십시오.

```

<!DOCTYPE html>
<html>
<head>
<style>
table {
  border: 1px solid black;
}
</style>
</head>
<body>

<h2>Single Border Around The Table:</h2>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
  </tr>
</table>

```



```
</body>
</html>
```

## 테이블 너비와 높이

테이블의 너비와 높이는 `width` 및 `height` 속성으로 정의됩니다.

아래 예제는 테이블 너비를 100 %로 설정하고 `<th>` 요소의 높이를 50px 로 설정합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
table, td, th {
    border: 1px solid black;
}
table {
    width: 100%;
    border-collapse: collapse;
}

th {
    height: 50px;
}
</style>
</head>
<body>

<h2>The width and height Properties</h2>
<p>Set the width of the table, and the height of the table header row:</p>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
  <tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$300</td>
  </tr>
</table>
```

```

</tr>
<tr>
  <td>Cleveland</td>
  <td>Brown</td>
  <td>$250</td>
</tr>
</table>

</body>
</html>

```

## 수평 정렬

이 `text-align` 속성은 `<th>` 또는 `<td>`에서 콘텐츠의 가로 정렬 (예 : 왼쪽, 오른쪽 또는 가운데)을 설정합니다.

기본적으로 `<th>` 요소의 내용은 가운데 정렬되고 `<td>` 요소의 내용은 왼쪽 정렬됩니다.

다음 예제는 `<th>` 요소에서 텍스트를 왼쪽 정렬합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
table, td, th {
  border: 1px solid black;
}
table {
  width: 100%;
  border-collapse: collapse;
}

th {
  text-align: left;
}
</style>
</head>
<body>

<h2>The text-align Property</h2>
<p>This property sets the horizontal alignment (like left, right, or center) of
the content in th or td:</p>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>

```

```

        <td>Peter</td>
        <td>Griffin</td>
        <td>$100</td>
    </tr>
    <tr>
        <td>Lois</td>
        <td>Griffin</td>
        <td>$150</td>
    </tr>
    <tr>
        <td>Joe</td>
        <td>Swanson</td>
        <td>$300</td>
    </tr>
    <tr>
        <td>Cleveland</td>
        <td>Brown</td>
        <td>$250</td>
    </tr>
</table>

</body>
</html>

```

## 수직 정렬

이 **vertical-align** 속성은 <th> 또는 <td>에서 내용의 수직 정렬 (예 : 위쪽, 아래쪽 또는 중간)을 설정합니다.

기본적으로 테이블 내용의 세로 정렬은 중간입니다 (<th> 및 <td> 요소 모두).

다음 예제는 <td> 요소의 세로 텍스트 정렬을 **bottom** 으로 설정합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
table, td, th {
    border: 1px solid black;
}
table {
    width: 100%;
    border-collapse: collapse;
}

td {
    height: 50px;
    vertical-align: bottom;
}
</style>
</head>

```

```

<body>

<h2>The vertical-align Property</h2>
<p>This property sets the vertical alignment (like top, bottom, or middle) of the
content in th or td.</p>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
  <tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$300</td>
  </tr>
  <tr>
    <td>Cleveland</td>
    <td>Brown</td>
    <td>$250</td>
  </tr>
</table>

</body>
</html>

```

## 테이블 패딩

테두리와 표의 내용 사이의 간격을 제어하려면 `padding` <td> 및 <th> 요소의 속성을 사용하십시오.

```

<!DOCTYPE html>
<html>
<head>
<style>
table, td, th {
  border: 1px solid black;
  text-align: left;
}
table {
  width: 100%;
  border-collapse: collapse;

```

```

}

th, td {
  padding: 15px;
}
</style>
</head>
<body>

<h2>The padding Property</h2>
<p>This property adds space between the border and the content in a table.</p>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
  <tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$300</td>
  </tr>
  <tr>
    <td>Cleveland</td>
    <td>Brown</td>
    <td>$250</td>
  </tr>
</table>

</body>
</html>

```

## 수평 분배기

**border-bottom** 가로 구분선의 경우 <th> 및 <td>에 속성을 추가하십시오 .

```

<!DOCTYPE html>
<html>
<head>
<style>
table {
  width: 100%;

```

```

border-collapse: collapse;
}

th, td {
padding: 8px;
text-align: left;
border-bottom: 1px solid #ddd;
}
</style>
</head>
<body>

<h2>Bordered Table Dividers</h2>
<p>Add the border-bottom property to th and td for horizontal dividers:</p>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
  <tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$300</td>
  </tr>
  <tr>
    <td>Cleveland</td>
    <td>Brown</td>
    <td>$250</td>
  </tr>
</table>

</body>
</html>

```

## 움직일 수있는 테이블

<tr> 의 **:hover** 선택기를 사용하여 마우스 위로 테이블 행을 강조 표시하십시오.

```

<!DOCTYPE html>
<html>
<head>

```

```

<style>
table {
  width: 100%;
  border-collapse: collapse;
}

th, td {
  padding: 8px;
  text-align: left;
  border-bottom: 1px solid #ddd;
}
tr:hover {background-color: #f5f5f5;}
</style>
</head>
<body>

<h2>Hoverable Table</h2>
<p>Move the mouse over the table rows to see the effect.</p>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
  <tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$300</td>
  </tr>
  <tr>
    <td>Cleveland</td>
    <td>Brown</td>
    <td>$250</td>
  </tr>
</table>

</body>
</html>

```

## 스트라이프 테이블

얼룩말 줄무늬 테이블의 경우 `nth-child()` 선택기를 사용하고 모든 짝수 (또는 홀수)

테이블 행에 **background-color**를 추가하십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
table {
  width: 100%;
  border-collapse: collapse;
}

th, td {
  padding: 8px;
  text-align: left;
}
tr:nth-child(even) {background-color: #f2f2f2;}
</style>
</head>
<body>

<h2>Striped Table</h2>
<p>For zebra-striped tables, use the nth-child() selector and add a background-
color to all even (or odd) table rows:</p>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
  <tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$300</td>
  </tr>
  <tr>
    <td>Cleveland</td>
    <td>Brown</td>
    <td>$250</td>
  </tr>
</table>

</body>
</html>
```



# 테이블 색상

아래 예제는 <th> 요소의 배경색과 텍스트 색을 지정합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
table {
  width: 100%;
  border-collapse: collapse;
}

th, td {
  padding: 8px;
  text-align: left;
}
tr:nth-child(even) {background-color: #f2f2f2;}

th {
  background-color: #4CAF50;
  color: white;
}
</style>
</head>
<body>

<h2>Colored Table Header</h2>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
  <tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$300</td>
  </tr>
  <tr>
    <td>Cleveland</td>
    <td>Brown</td>
    <td>$250</td>
  </tr>
</table>
```

```

</tr>
</table>

</body>
</html>

```

## 반응형 테이블

화면이 너무 작아서 전체 내용을 표시 할 수 없는 경우 반응 형 테이블에 가로 스크롤 막대가 표시됩니다.

<table> 요소 주위에 컨테이너 요소 (예 : <div>)를 **overflow-x:auto** 로 추가하여 반응형으로 만듭니다.

[illegible]

```

<th>Points</th>
<th>Points</th>
<th>Points</th>
</tr>
<tr>
<td>Jill</td>
<td>Smith</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>94</td>
</tr>
<tr>
<td>Adam</td>
<td>Johnson</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
</tr>
</table>

</body>
</html>

```

**참고** : OS X Lion (Mac)에서는 스크롤 막대가 기본적으로 숨겨져 있으며 "overflow : scroll"이 설정되어 있어도 사용할 때만 표시됩니다.

# CSS 레이아웃-표시 속성

이 `display` 속성은 레이아웃을 제어하는 데 가장 중요한 CSS 속성입니다.

## 디스플레이 속성

이 `display` 속성은 요소가 표시되는지 여부를 지정합니다.

모든 HTML 요소에는 요소 유형에 따라 기본 표시 값이 있습니다. 대부분의 요소에 대한 기본 표시 값은 `block` 또는 `inline`입니다.

## 블록 레벨 요소

블록 수준 요소는 항상 새 줄에서 시작하여 사용 가능한 전체 너비를 차지합니다 (가능한 한 왼쪽과 오른쪽으로 확장).

`<div>` 요소는 블록 레벨 요소입니다.

블록 레벨 요소의 예 :

- `<div>`
- `<h1>` - `<h6>`
- `<p>`
- `<form>`
- `<header>`
- `<footer>`
- `<section>`

## 인라인 요소

인라인 요소는 새 줄에서 시작되지 않으며 필요한 만큼만 폭을 차지합니다.

이것은 단락 `안의 인라인 <span> 요소`입니다.

인라인 요소의 예 :

- `<span>`
- `<a>`
- `<img>`

## 디스플레이 : 없음;

`display: none;` 일반적으로 JavaScript 와 함께 사용하여 요소를 삭제하거나 다시 만들지 않고 요소를 숨기고 표시합니다. 이 방법을 알고 싶다면 이 페이지의 마지막 예를 살펴보십시오.

`<script>`요소는 `display: none;` 기본값으로 사용 한다.

## 기본 표시 값 무시

언급 한대로 모든 요소에는 기본 표시 값이 있습니다. 그러나 이를 무시할 수 있습니다.

인라인 요소를 블록 요소로 변경하거나 그 반대로 변경하면 페이지를 특정 방식으로 보이게하고 웹 표준을 계속 따르는 데 유용 할 수 있습니다.

일반적인 예는 `<li>` 가로 메뉴에 인라인 요소를 만드는 것입니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
li {
  display: inline;
}
</style>
</head>
<body>

<p>Display a list of links as a horizontal menu:</p>

<ul>
  <li><a href="#" target="_blank">HTML</a></li>
  <li><a href="#" target="_blank">CSS</a></li>
  <li><a href="#" target="_blank">JavaScript</a></li>
```

```
</ul>

</body>
</html>
```

**참고** : 요소의 표시 속성을 설정하면 요소의 종류가 아니라 요소가 표시되는 방식만 변경됩니다. 따라서 인라인 요소 `display: block;`에는 다른 블록 요소가 포함될 수 없습니다.

다음 예제는 `<span>` 요소를 블록 요소로 표시합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
span {
  display: block;
}
</style>
</head>
<body>

<span>A display property with a value of "block" results in</span>
<span>a line break between the two elements.</span>

</body>
</html>
```

다음 예제는 `<a>` 요소를 블록 요소로 표시합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
a {
  display: block;
}
</style>
</head>
<body>

<p>Display links as block elements:</p>

<a href="#" target="_blank">HTML</a>
<a href="#" target="_blank">CSS</a>
<a href="#" target="_blank">JavaScript</a>

</body>
</html>
```

# 요소 숨기기-표시 : 없음 또는 가시성 : 숨김?

`display` 속성을 `none` 로 설정하여 요소를 숨길 수 있습니다. 요소가 숨겨지고 요소가 없는 것처럼 페이지가 표시됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {
  display: none;
}
</style>
</head>
<body>

<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p> Notice that the h1 element with display: none; does not take up any space.</p>

</body>
</html>
```

`visibility:hidden;` 또한 요소를 숨깁니다.

그러나 요소는 여전히 이전과 동일한 공간을 차지합니다. 요소는 숨겨 지지만 여전히 레이아웃에 영향을 줍니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {
  visibility: hidden;
}
</style>
</head>
<body>

<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the hidden heading still takes up space.</p>

</body>
</html>
```

# CSS 표시 / 가시성 속성

Property	Description
<a href="#">display</a>	Specifies how an element should be displayed
<a href="#">visibility</a>	Specifies whether or not an element should be visible

## CSS 레이아웃-너비 및 최대 너비

### 너비, 최대 너비 및 여백 사용 : 자동;

이전 장에서 언급 한 바와 같이; 블록 레벨 요소는 항상 사용 가능한 전체 너비를 차지합니다 (가능한 한 왼쪽과 오른쪽으로 확장).

**width** 블록 레벨 요소를 설정하면 컨테이너의 가장자리까지 늘어나지 않습니다. 그런 다음 여백을 자동으로 설정하여 컨테이너 내에서 요소를 가로로 가운데에 맞출 수 있습니다. 요소는 지정된 너비를 차지하고 나머지 공간은 두 여백 사이에서 동일하게 분할됩니다.

**참고 :** `<div>` 위 의 문제 는 브라우저 창이 요소 너비보다 작을 때 발생합니다. 그런 다음 브라우저는 페이지에 가로 스크롤 막대를 추가합니다.

**max-width** 대신 이 상황에서 사용 하면 브라우저의 작은 창 처리가 향상됩니다. 작은 장치에서 사이트를 사용할 수 있게 할 때 중요합니다.

**팁 :** 두 `div` 의 차이점을 보려면 브라우저 창의 크기를 500px 미만으로 조정하십시오!

위의 두 `div` 예는 다음과 같습니다

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
  width: 500px;
  margin: auto;
  border: 3px solid #73AD21;
}

div.ex2 {
  max-width: 500px;
  margin: auto;
  border: 3px solid #73AD21;
```



```
}
</style>
</head>
<body>

<div class="ex1"> This div element has width: 500px;</div>
<br>
<div class="ex2"> This div element has max-width: 500px;</div>

<p><strong>Tip:</strong> Drag the browser window to smaller than 500px wide, to
see the difference between the two divs! </p>

</body>
</html>
```

# CSS 레이아웃-위치 속성

이 **position** 속성은 요소에 사용되는 위치 결정 방법 유형 (정적, 상대적, 고정, 절대 또는 고정)을 지정합니다.

## 위치 속성

이 **position** 속성은 요소에 사용되는 위치 결정 방법의 유형을 지정합니다.

다섯 가지 위치 값이 있습니다.

- **static**
- **relative**
- **fixed**
- **absolute**
- **sticky**

그런 다음 요소는 **top**, **bottom**, **left** 및 **right** 속성을 사용하여 배치됩니다. 그러나 속성을 **position** 먼저 설정 하지 않으면 이러한 속성이 작동하지 않습니다. 위치 값에 따라 다르게 작동합니다.

## 위치 : 정적;

HTML 요소는 기본적으로 정적 위치에 있습니다.

정적 위치 요소는 `top`, `bottom`, `left` 및 `right` 속성의 영향을 받지 않습니다.

`position: static;`를 가진 요소는 특별한 방식으로 배치되지 않습니다. 항상 페이지의 정상적인 흐름에 따라 배치됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.static {
  position: static;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: static;</h2>

<p>An element with position: static; is not positioned in any special way; it is
always positioned according to the normal flow of the page:</p>

<div class="static">
  This div element has position: static;
</div>

</body>
</html>
```

## 위치 : 상대;

`position: relative;` 요소는 원래 위치를 기준으로 배치됩니다.

상대적으로 배치 된 요소의 위쪽, 오른쪽, 아래쪽 및 왼쪽 속성을 설정하면 일반 위치에서 멀어 지도록 조정됩니다. 다른 내용은 요소가 남긴 간격에 맞게 조정되지 않습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  left: 30px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>
```

```
<h2>position: relative;</h2>

<p>An element with position: relative; is positioned relative to its normal position:</p>

<div class="relative">
  This div element has position: relative;
</div>

</body>
</html>
```

## 위치 : 고정;

**position: fixed;** 요소는 뷰포트를 기준으로 배치되므로 페이지가 스크롤 되더라도 항상 동일한 위치에 유지됩니다. **top, right, bottom** 및 **left** 속성은 요소를 배치하는 데 사용됩니다.

고정 된 요소는 일반적으로 위치했던 페이지에 공백을 남기지 않습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: fixed;</h2>

<p>An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled:</p>

<div class="fixed">
  This div element has position: fixed;
</div>

</body>
</html>
```

# 위치 : 절대;

`position: absolute;` 는 고정 요소와 같이 뷰포트를 기준으로 배치하는 대신 가장 가까운 위치를 조상을 기준으로 요소가 배치됩니다.

절대 위치 요소에 위치된 조상이 없으면 문서 본문을 사용하고 페이지 스크롤과 함께 이동합니다.

참고 : "위치"요소는 위치가 `static` 이외의 요소입니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  width: 400px;
  height: 200px;
  border: 3px solid #73AD21;
}

div.absolute {
  position: absolute;
  top: 80px;
  right: 0;
  width: 200px;
  height: 100px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: absolute;</h2>

<p>An element with position: absolute; is positioned relative to the nearest
positioned ancestor (instead of positioned relative to the viewport, like
fixed):</p>

<div class="relative">
  This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>

</body>
</html>
```

# 위치 : 끈적임;

**position: sticky;** 요소는 사용자의 스크롤 위치를 기준으로 배치됩니다.

스티커 요소는 스크롤 위치에 따라 **relative** 및 **fixed** 사이를 전환 합니다. 주어진 오프셋 위치가 뷰포트에서 충족 될 때까지 상대적으로 배치 된 다음 위치에 고정됩니다 (예 : 위치 : 고정).

**참고 :** Internet Explorer, Edge 15 및 이전 버전은 고정 위치를 지원하지 않습니다. Safari에는 `-webkit-` 접두사가 필요합니다 (아래 예 참조).

또한 **top**, **right**, **bottom** 또는 **left** 중 하나 이상 지정해야 끈적 위치가 작동 할 수 있습니다.

이 예에서, 고정 **top: 0** 위치는 스크롤 위치에 도달 할 때 페이지 상단 ( )에 고정됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.sticky {
  position: -webkit-sticky; /* Safari */
  position: sticky;
  top: 0;
  background-color: green;
  border: 2px solid #4CAF50;
}
</style>
</head>
<body>

<p>Try to <b>scroll</b>inside this frame to understand how sticky positioning works.</p>
<p>Note: IE/Edge 15 and earlier versions do not support sticky position.</p>

<div class="sticky">I am sticky!</div>

<div style="padding-bottom:2000px">
<p>In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.</p>
<p>Scroll back up to remove the stickyness.</p>
<p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.</p>
<p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec
```

```
et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.</p>
</div>

</body>
</html>
```

## 겹치는 요소

요소가 배치되면 다른 요소와 겹칠 수 있습니다.

이 **z-index** 속성은 요소의 스택 순서를 지정합니다 (요소는 다른 요소의 앞 또는 뒤에 배치되어야 함).

요소는 양수 또는 음수 스택 순서를 가질 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: -1;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

<p> Because the image has a z-index of -1, it will be placed behind the text.</p>

</body>
</html>
```

스택 순서가 큰 요소는 항상 스택 순서가 낮은 요소 앞에 있습니다.

**참고 :** 위치가 **z-index** 지정된 두 요소가 지정 없이 겹치는 경우 **HTML** 코드에서 마지막에 배치 된 요소가 맨 위에 표시됩니다.

## 이미지에서 텍스트 배치

이미지 위에 텍스트를 배치하는 방법 :

## topLeft

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  position: relative;
}

.topleft {
  position: absolute;
  top: 8px;
  left: 16px;
  font-size: 18px;
}

img {
  width: 100%;
  height: auto;
  opacity: 0.3;
}

</style>
</head>
<body>

<h2>Image Text</h2>
<p>Add some text to an image in the top left corner:</p>
<div class="container">
  
  <div class="topleft">Top Left</div>
</div>

</body>
</html>
```

## topRight

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  position: relative;
}

.topright {
  position: absolute;
  top: 8px;
  right: 16px;
  font-size: 18px;
}

img {
  width: 100%;
```

```

    height: auto;
    opacity: 0.3;
}

</style>
</head>
<body>

<h2>Image Text</h2>
<p>Add some text to an image in the top right corner:</p>
<div class="container">
    
    <div class="topright">Top Right</div>
</div>

</body>
</html>

```

## bottomLeft

```

<!DOCTYPE html>
<html>
<head>
<style>
.container {
    position: relative;
}

.bottomleft {
    position: absolute;
    bottom: 8px;
    left: 16px;
    font-size: 18px;
}

img {
    width: 100%;
    height: auto;
    opacity: 0.3;
}

</style>
</head>
<body>

<h2>Image Text</h2>
<p>Add some text to an image in the bottom left corner:</p>
<div class="container">
    
    <div class="bottomleft">Bottom Left</div>
</div>

</body>
</html>

```



## bottomRight

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  position: relative;
}

.bottomright {
  position: absolute;
  bottom: 8px;
  right: 16px;
  font-size: 18px;
}

img {
  width: 100%;
  height: auto;
  opacity: 0.3;
}

</style>
</head>
<body>

<h2>Image Text</h2>
<p>Add some text to an image in the bottom right corner:</p>
<div class="container">
  
  <div class="bottomright">Bottom Right</div>
</div>

</body>
</html>
```

## Centered

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  position: relative;
}

.center {
  position: absolute;
  left: 0;
  top: 50%;
  text-align: center;
  font-size: 18px;
}

img {
```

```
width: 100%;  
height: auto;  
opacity: 0.3;  
}  
  
</style>  
</head>  
<body>  
  
<h2>Image Text</h2>  
<p>Center text in image:</p>  
<div class="container">  
    
  <div class="center">Centered</div>  
</div>  
  
</body>  
</html>
```

## 모든 CSS 위치 속성

Property	Description
<a href="#">bottom</a>	Sets the bottom margin edge for a positioned box
<a href="#">clip</a>	Clips an absolutely positioned element
<a href="#">left</a>	Sets the left margin edge for a positioned box
<a href="#">position</a>	Specifies the type of positioning for an element
<a href="#">right</a>	Sets the right margin edge for a positioned box
<a href="#">top</a>	Sets the top margin edge for a positioned box
<a href="#">z-index</a>	Sets the stack order of an element

## CSS 레이아웃-오버플로

CSS **overflow** 속성은 너무 커서 영역에 맞지 않는 내용의 내용을 제어합니다.

## CSS 오버플로

이 **overflow** 속성은 요소의 내용이 너무 커서 지정된 영역에 맞지 않을 때 내용을 자를 것인지 스크롤 막대를 추가 할 것인지를 지정합니다.

이 **overflow** 속성의 값은 다음과 같습니다.

- **visible**- 기본. 오버 플로우가 잘리지 않습니다. 내용은 요소 상자 외부에서 렌더링 됩니다.
- **hidden** -오버플로가 잘리고 나머지 내용은 보이지 않습니다.
- **scroll** -오버플로가 잘리고 스크롤 막대가 추가되어 나머지 내용을 볼 수 있습니다.
- **auto**-와 비슷 **scroll** 하지만 필요할 때만 스크롤 막대를 추가합니다.

참고 : 이 **overflow** 속성은 지정된 높이의 블록 요소에만 작동합니다.

참고 : OS X Lion (Mac)에서는 스크롤 막대가 기본적으로 숨겨져 있으며 "overflow : scroll"이 설정되어 있어도 사용할 때만 표시됩니다.

## 오버플로 : 표시

기본적으로 overflow 는 **visible** 입니다. 즉, 잘리지 않고 요소 상자 외부에서 렌더링 됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 200px;
  height: 50px;
  background-color: #eee;
  overflow: visible;
}
</style>
</head>
<body>

<h2>CSS Overflow</h2>
<p>By default, the overflow is visible, meaning that it is not clipped and it renders outside the element's box:</p>

<div>You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.</div>

</body>
</html>
```

# 오버플로 : 숨겨진

오버 플로우가 **hidden** 값으로 콘텐츠의 나머지는 숨겨 잘리고:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 200px;
  height: 50px;
  background-color: #eee;
  border: 1px dotted black;
  overflow: hidden;
}
</style>
</head>
<body>

<h2>CSS Overflow</h2>
<p>With the hidden value, the overflow is clipped, and the rest of the content is hidden:</p>
<p>Try to remove the overflow property to understand how it works.</p>

<div>You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.</div>

</body>
</html>
```

# 오버플로 : 스크롤

overflow: **scroll** 값으로 설정하면 잘리고 상자 안에 스크롤 막대가 추가됩니다. 이렇게 하면 (필요하지 않더라도) 스크롤 막대가 가로 및 세로로 추가됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 200px;
  height: 100px;
  background-color: #eee;
  border: 1px dotted black;
  overflow: scroll;
}
</style>
</head>
<body>
```

```
</style>
</head>
<body>

<h2>CSS Overflow</h2>
<p>Setting the overflow value to scroll, the overflow is clipped and a scrollbar is added to scroll inside the box. Note that this will add a scrollbar both horizontally and vertically (even if you do not need it):</p>

<div>You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.</div>

</body>
</html>
```

## 오버플로 : 자동

`auto` 는 `scroll` 값과 유사 하지만, 필요한 경우에만 스크롤 바를 추가합니다 :

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 200px;
  height: 50px;
  background-color: #eee;
  border: 1px dotted black;
  overflow: auto;
}
</style>
</head>
<body>

<h2>CSS Overflow</h2>
<p>The auto value is similar to scroll, only it add scrollbars when necessary:</p>

<div>You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.</div>

</body>
</html>
```

# overflow-x 및 overflow-y

**overflow-x** 및 **overflow-y** 특성은 수평 또는 수직 (또는 둘 다)의 콘텐츠 플로우를 변경할지 여부를 지정한다 :

**overflow-x** 내용의 왼쪽 / 오른쪽 가장자리로 수행 할 작업을 지정합니다.

**overflow-y** 내용의 위쪽 / 아래쪽 가장자리로 수행 할 작업을 지정합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 200px;
  height: 50px;
  background-color: #eee;
  border: 1px dotted black;
  overflow-x: hidden; /* Hide horizontal scrollbar */
  overflow-y: scroll; /* Add vertical scrollbar */
}
</style>
</head>
<body>

<h2>CSS Overflow</h2>
<p>You can also change the overflow of content horizontally or vertically.</p>
<p>overflow-x specifies what to do with the left/right edges of the content.<br>
overflow-y specifies what to do with the top/bottom edges of the content.</p>

<div>You can use the overflow property when you want to have better control of
the layout. The overflow property specifies what happens if content overflows an
element's box.</div>

</body>
</html>
```

## 모든 CSS 오버 플로우 속성

Property	Description
<a href="#">overflow</a>	Specifies what happens if content overflows an element's box
<a href="#">overflow-x</a>	Specifies what to do with the left/right edges of the content if it overflows the element's content area
<a href="#">overflow-y</a>	Specifies what to do with the top/bottom edges of the content if it overflows the element's content area

# CSS 레이아웃-float 및 clear

CSS **float** 속성은 요소가 어떻게 떠야 하는지를 지정합니다.

CSS **clear** 속성은 지워진 요소 옆에 어떤 요소를 띄울 수 있는지 지정합니다.

## float 속성

이 **float** 속성은 콘텐츠를 배치하고 형식을 지정하는 데 사용됩니다 (예 : 이미지를 컨테이너의 텍스트 왼쪽에 떠 다니게 함).

**float** 속성은 다음 값 중 하나를 가질 수 있습니다 :

- left-요소가 컨테이너의 왼쪽에 뜬다
- right-요소가 컨테이너의 오른쪽에 뜬다
- none-요소가 플로팅되지 않습니다 (텍스트에서 발생하는 위치에 표시됨). 이것이 기본값입니다
- inherit-요소는 부모의 float 값을 상속받습니다.

가장 간단한 용도로 이 **float** 속성을 사용하여 이미지 주위에 텍스트를 감쌀 수 있습니다.

## 예-float : 오른쪽;

다음 예제는 이미지가 텍스트 에서 **오른쪽** 으로 부동되도록 지정 합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: right;
}
</style>
</head>
<body>
```

```

<p>In this example, the image will float to the right in the paragraph, and the
text in the paragraph will wrap around the image.</p>
<p>Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi
lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl
est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue
ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio.
Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis
dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis
imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis.
Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor
porta. Cras ac leo purus. Mauris quis diam velit.</p>

</body>
</html>

```

## 예-float : 왼쪽;

다음 예제는 이미지가 텍스트 에서 **왼쪽** 으로 부동되도록 지정 합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
img {
float: left;
}
</style>
</head>
<body>

<p>In this example, the image will float to the left in the paragraph, and the
text in the paragraph will wrap around the image.</p>
<p>Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi
lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl
est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue
ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio.
Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis
dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis
imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis.
Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor
porta. Cras ac leo purus. Mauris quis diam velit.</p>

</body>
</html>

```



# 예-플로트 없음

다음 예제에서는 이미지가 텍스트에서 발생하는 위치에 표시됩니다 (float : none;).

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: none;
}
</style>
</head>
<body>

<p>In this example, the image will be displayed just where it occurs in the text
(float: none;).</p>
<p>Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi
lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl
est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue
ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio.
Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis
dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis
imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis.
Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor
porta. Cras ac leo purus. Mauris quis diam velit.</p>

</body>
</html>
```

## 명확한 재산

이 **clear** 속성은 지워진 요소 옆에 어느 요소가 떠 올릴 수 있는지 지정합니다.

**clear** 속성은 다음 값 중 하나를 가질 수 있습니다 :

- none-양면에 플로팅 요소를 허용합니다. 이것이 기본값입니다
- left-왼쪽에 부동 요소가 허용되지 않습니다
- right- 오른쪽에 부동 요소가 허용되지 않습니다
- both-왼쪽 또는 오른쪽에 부동 요소가 허용되지 않습니다
- inherit-요소는 부모의 명확한 값을 상속합니다

**clear** 속성 을 사용하는 가장 일반적인 방법 **float** 은 요소 에서 속성을 사용한 후에 입니다.

플로트를 지울 때는 클리어와 플로트를 일치시켜야합니다. 요소가 왼쪽에 떠 있으면 왼쪽으로 지워야합니다. 플로팅 된 요소는 계속 플로팅되지만 삭제 된 요소는 웹 페이지 아래에 나타납니다.

다음 예제에서는 부동 소수점을 왼쪽으로 지웁니다. **div** 의 왼쪽에 부동 요소가 허용되지 않음을 의미합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div1 {
  float: left;
  width: 100px;
  height: 50px;
  margin: 10px;
  border: 3px solid #37AD21;
}

div2 {
  border: 1px solid red;
}

div3 {
  float: left;
  width: 100px;
  height: 50px;
  margin: 10px;
  border: 3px solid #37AD21;
}

div4 {
  border: 1px solid red;
  clear: left;
}

</style>
</head>
<body>

<h2>Without clear</h2>
<div class="div1">div1</div>
<div class="div2">div2 - Notice that div2 is after div1 in the HTML code.
However, since div1 floats to the left, the text in div2 flows around div1.</div>
</br></br>

<h2>With clear</h2>
<div class="div3">div3</div>
<div class="div4">div4 - Here, clear: left; moves div4 down below the floating
div3. The value "left" clears elements floated to the left. You can also clear
"right" and "both".</div>
</br></br>
```

```
</body>
</html>
```

## Clearfix 핵

요소가 포함 된 요소보다 키가 크고 부동 상태 인 경우 컨테이너 외부로 "오버 플로우"됩니다.

그런 다음 **overflow: auto;** 포함 요소에 추가 하여이 문제를 해결할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  padding: 5px;
  border: 3px solid #4CAF50;
}

.img1 {
  float: right;
}

.clearfix {
  overflow: auto;
}

.img2 {
  float: right;
}
</style>
</head>
<body>
```

**<p>**In this example, the image is taller than the element containing it, and it is floated, so it overflows outside of its container:**</p>**

```
<div>
  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet,
  nulla et dictum interdum...
</div>
<p style="clear:right"> Add a clearfix class with overflow: auto; to the
containing element, to fix this problem:</div>

<div class="clearfix">
  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet,
  nulla et dictum interdum...
</div>
```

```
</body>
</html>
```

`overflow: auto`로 clearfix 작업은 마진과 패딩 (다른 사람이 스크롤바를 볼 수 있습니다)의 제어를 유지할 수 있습니다. 그러나 **새롭고 현대적인 clearfix hack**은 사용하기에 안전하며 대부분의 웹 페이지에 다음 코드가 사용됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  padding: 5px;
  border: 3px solid #4CAF50;
}

.img1 {
  float: right;
}

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}

.img2 {
  float: right;
}
</style>
</head>
<body>

<p>In this example, the image is taller than the element containing it, and it is
floated, so it overflows outside of its container:</p>

<div>
  <h2>Without Clearfix</h2>
  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet,
  nulla et dictum interdum...
</div>

<p style="clear:right">Add the clearfix hack to the containing element, to fix
this problem:</div>

<div class="clearfix">
  <h2>With Clearfix</h2>
  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet,
  nulla et dictum interdum...
</div>

</body>
</html>
```

# 상자 그리드 / 동일한 너비 상자

이 `float` 속성을 사용하면 내용 상자를 나란히 띄울 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

.box {
  float: left;
  width: 33.33%; /* three boxes (use 25% for four, and 50% for two, etc) */
  padding: 50px; /* if you want space between the images */
}

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
</style>
</head>
<body>

<h2>Grid of Boxes</h2>
<p>Float boxes side by side:</p>

<div class="clearfix">
  <div class="box" style="background-color:#bbb">
    <p>Some text inside the box.</p>
  </div>
  <div class="box" style="background-color:#ccc">
    <p>Some text inside the box.</p>
  </div>
  <div class="box" style="background-color:#ddd">
    <p>Some text inside the box.</p>
  </div>
</div>

<p>Note that we also use the clearfix hack to take care of the layout flow, and
that add the box-sizing property to make sure that the box doesn't break due to
extra padding. Try to remove this code to see the effect.</p>

</body>
</html>
```

상자 크기란 무엇입니까?

3 개의 부동 상자를 나란히 쉽게 만들 수 있습니다. 그러나 각 상자의 너비를 넓히는 항목 (예 : 패딩 또는 테두리)을 추가하면 상자가 깨집니다. 이 **box-sizing** 속성을 사용하면 상자의 전체 너비 (및 높이)에 패딩 및 테두리를 포함하여 패딩이 상자 안에 유지되고 끊어지지 않도록 할 수 있습니다.

## 이미지 나란히

상자 그리드를 사용하여 이미지를 나란히 표시 할 수도 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

.img-container {
  float: left;
  width: 33.33%; /* three containers (use 25% for four, and 50% for two, etc) */
  padding: 5px; /* if you want space between the images */
}

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
</style>
</head>
<body>

<h2>Images Side by Side</h2>
<p>Float images side by side:</p>

<div class="clearfix">
  <div class="img-container">
    
  </div>
  <div class="img-container">
    
  </div>
  <div class="img-container">
    
  </div>
</div>

<p>Note that we also use the clearfix hack to take care of the layout flow, and
that we add the box-sizing property to make sure that the image container doesn't
break due to extra padding. Try to remove this code to see the effect.</p>
```

```
</body>
</html>
```

## 동등한 높이 상자

이전 예제에서는 상자를 같은 너비로 나란히 플로팅하는 방법을 배웠습니다. 그러나 같은 높이의 부동 상자를 만드는 것은 쉽지 않습니다. 그러나 빠른 수정은 아래 예와 같이 고정 된 높이를 설정하는 것입니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

.box {
  float: left;
  width: 50%;
  padding: 50px;
  height: 300px;
}

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
</style>
</head>
<body>

<h2>Equal Height Boxes</h2>
<p>Floating boxes with equal heights:</p>

<div class="clearfix">
  <div class="box" style="background-color:#bbb">
    <h2>Box 1</h2>
    <p>Some content, some content, some content</p>
  </div>
  <div class="box" style="background-color:#ccc">
    <h2>Box 2</h2>
    <p>Some content, some content, some content</p>
    <p>Some content, some content, some content</p>
    <p>Some content, some content, some content</p>
  </div>
</div>
```

<p>This example not very flexible. It is ok to use CSS height if you can guarantee that the boxes will always have the same amount of content in them, but that's not always the case. If you try the example above on a mobile phone (or

```
resize the browser window), you will see that the second box's content will be
displayed outside of the box.</p>
<p>Go back to the tutorial and find another solution, if this is not what you
want.</p>

</body>
</html>
```

그러나 이것은 매우 유연하지 않습니다. 상자에 항상 같은 양의 내용이 포함되도록 보장 할 수 있다면 괜찮습니다. 그러나 여러 번 내용이 동일하지 않습니다. 위의 예를 휴대폰에서 시도하면 두 번째 상자의 내용이 상자 외부에 표시됩니다. CSS3 Flexbox가 유용한 이유는 가장 긴 상자만큼 상자를 자동으로 늘릴 수 있기 때문입니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-wrap: nowrap;
  background-color: DodgerBlue;
}

.flex-container .box {
  background-color: #f1f1f1;
  width: 50%;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

<h1> Flexible Boxes</h1>

<div class="flex-container">
  <div class="box">Box 1 - This is some text to make sure that the content gets
really tall. This is some text to make sure that the content gets really tall.
  </div>
  <div class="box">Box 2 - My height will follow Box 1.</div>
</div>

<p> Try to resize the browser window to see the flexible layout.</p>
<p><strong>Note:</strong> Flexbox is not supported in Internet Explorer 10 or
earlier versions.</p>

</body>
</html>
```



Flexbox의 유일한 문제점은 Internet Explorer 10 또는 이전 버전에서는 작동하지 않는다는 것입니다. Flexbox Layout Module에 대한 자세한 내용은 [CSS Flexbox Chapter](#)를 참조하십시오 .

## 탐색 메뉴

사용 `float` 수평 메뉴를 만들 하이퍼 링크의 목록 :

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}

li {
  float: left;
}

li a {
  display: inline-block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

li a:hover {
  background-color: #111;
}

.active {
  background-color: red;
}
</style>
</head>
<body>

<ul>
  <li><a href="#home" class="active">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
```

```
</body>
</html>
```

## 웹 레이아웃 예

**float** 속성을 사용하여 전체 웹 레이아웃을 수행하는 것도 일반적입니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

.header, .footer {
  background-color: grey;
  color: white;
  padding: 15px;
}

.column {
  float: left;
  padding: 15px;
}

.clearfix::after {
  content: "";
  clear: both;
  display: table;
}

.menu {
  width: 25%;
}

.content {
  width: 75%;
}

.menu ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

.menu li {
  background-color: #33b5e5;
  color: #ffffff;
  margin-bottom: 8px;
  padding: 8px;
}
```

```

.menu li:hover {
  background-color: #0099cc;
}

</style>
</head>
<body>

<div class="header">
  <h2>Chania</h2>
</div>

<div class="clearfix">
  <div class="column menu">
    <ul>
      <li>The Flight</li>
      <li>The City</li>
      <li>The Island</li>
      <li>The Food</li>
    </ul>
  </div>

  <div class="column content">
    <h1>The City</h1>
    <p>Chania is the capital of the Chania region on the island of Crete. The city
    can be divided in two parts, the old town and the modern city.</p>
    <p>You will learn more about web layout and responsive web pages in a later
    chapter.</p>
  </div>
</div>

<div class="footer">
  <p>Footer Text</p>
</div>

</body>
</html>

```

## 모든 CSS Float 속성

Property	Description
<a href="#">box-sizing</a>	Defines how the width and height of an element are calculated: should they include padding and borders, or not
<a href="#">clear</a>	Specifies what elements can float beside the cleared element and on which side
<a href="#">float</a>	Specifies how an element should float
<a href="#">overflow</a>	Specifies what happens if content overflows an element's box
<a href="#">overflow-x</a>	Specifies what to do with the left/right edges of the content if it overflows the element's content area
<a href="#">overflow-y</a>	Specifies what to do with the top/bottom edges of the

content if it overflows the element's content area
--

# CSS 레이아웃-표시 : 인라인 블록

## 디스플레이 : 인라인 블록 값

`display: inline` 와 비교할 때 가장 큰 차이점은 `display: inline-block` 요소의 너비와 높이를 설정할 수 있다는 것입니다.

또한 `display: inline-block` 을 사용하면 상단 및 하단 여백 / 패딩이 존중되지만 `display: inline` 그렇지 않습니다.

`display: block` 와 비교할 때 가장 큰 차이점은 `display: inline-block` 요소 뒤에 줄 바꿈을 추가하지 않으므로 요소가 다른 요소 옆에 있을 수 있다는 것입니다.

다음 예제의 다른 동작을 보여줍니다 `display: inline`, `display: inline-block` 그리고 `display: block`:

```
<!DOCTYPE html>
<html>
<head>
<style>
span.a {
  display: inline; /* the default for span */
  width: 100px;
  height: 100px;
  padding: 5px;
  border: 1px solid blue;
  background-color: yellow;
}

span.b {
  display: inline-block;
  width: 100px;
  height: 100px;
  padding: 5px;
  border: 1px solid blue;
  background-color: yellow;
}

span.c {
  display: block;
  width: 100px;
  height: 100px;
}
```

```
padding: 5px;
border: 1px solid blue;
background-color: yellow;
}

</style>
</head>
<body>

<h1>The display Property</h1>

<h2>display: inline</h2>
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum
consequat scelerisque elit sit amet consequat. Aliquam erat volutpat.<span
class="a">Aliquam</span><span class="a">venenatis</span>gravida nisl sit amet
facilisis. Nullam cursus fermentum velit sed laoreet.</div>

<h2>display: inline-block</h2>
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum
consequat scelerisque elit sit amet consequat. Aliquam erat volutpat.<span
class="b">Aliquam</span><span class="b">venenatis</span>gravida nisl sit amet
facilisis. Nullam cursus fermentum velit sed laoreet.</div>

<h2>display: block</h2>
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum
consequat scelerisque elit sit amet consequat. Aliquam erat volutpat.<span
class="c">Aliquam</span><span class="c">venenatis</span>gravida nisl sit amet
facilisis. Nullam cursus fermentum velit sed laoreet.</div>
</body>
</html>
```

## 인라인 블록을 사용하여 탐색 링크 작성

한 가지 일반적인 용도 **display: inline-block** 는 목록 항목을 세로가 아닌 가로로 표시하는 것입니다. 다음 예제는 가로 탐색 링크를 작성합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.nav {
background-color: yellow;
list-style-type: none;
text-align: center;
padding: 0;
margin: 0;
}

.nav li {
display: inline-block;
font-size: 20px;
padding: 20px;
}
```

```

</style>
</head>
<body>

<h1>Horizontal Navigation Links</h1>
<p>By default, list items are displayed vertically. In this example we use
display: inline-block to display them horizontally (side by side).</p>
<p>Note: If you resize the browser window, the links will automatically break
when it becomes too crowded.</p>

<ul class="nav">
  <li><a href="#home">Home</a></li>
  <li><a href="#about">About</a></li>
  <li><a href="#clients">Our Clients</a></li>
  <li><a href="#contact">Contact Us</a></li>
</ul>

</body>
</html>

```

# CSS 레이아웃-가로 및 세로 정렬

## 중심 정렬 요소

<div>와 같은 블록 요소를 가로로 가운데에 맞추려면 `margin: auto;`

요소의 너비를 설정하면 컨테이너의 가장자리로 늘어나지 않습니다.

그러면 요소가 지정된 너비를 차지하고 나머지 공간은 두 여백 사이에서 동일하게 분할됩니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
.center {
  margin: auto;
  width: 50%;
  border: 3px solid green;
  padding: 10px;
}
</style>
</head>
<body>

```

```
<h2>Center Align Elements</h2>
<p>To horizontally center a block element (like div), use margin: auto;</p>

<div class="center">
  <p><b>Note:</b>Using margin:auto will not work in IE8, unless a !DOCTYPE is
declared.</p>
</div>

</body>
</html>
```

**참고** :width 속성이 설정되어 있지 않거나 100 %로 설정되어 있으면 가운데 정렬이 적용 되지 않습니다.

## 가운데 맞춤 텍스트

텍스트를 요소의 가운데에 맞추려면 `text-align: center;`

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  text-align: center;
  border: 3px solid green;
}
</style>
</head>
<body>

<h2>Center Text</h2>

<div class="center">
  <p>This text is centered.</p>
</div>

</body>
</html>
```

## 이미지 중심

이미지를 중앙에 맞추려면 왼쪽 및 오른쪽 여백을 설정 `auto` 하고 `block` 요소로 만듭니다

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

```

<head>
<style>
img {
  display: block;
  margin-left: auto;
  margin-right: auto;
  width: 40%;
}
</style>
</head>
<body>

<h2>Center an Image</h2>
<p>To center an image, set left and right margin to auto, and make it into a
block element.</p>



</body>
</html>

```

## 좌우 정렬-위치 사용

요소를 정렬하는 한 가지 방법은 다음을 사용하는 것입니다 `position: absolute;`.

```

<!DOCTYPE html>
<html>
<head>
<style>
.right {
  position: absolute;
  right: 0px;
  width: 300px;
  border: 3px solid #73AD21;
  padding: 10px;
}
</style>
</head>
<body>

<h2>Right Align</h2>
<p>An example of how to right align elements with the position property:</p>

<div class="right">
  <p>In my younger and more vulnerable years my father gave me some advice that
I've been turning over in my mind ever since.</p>
</div>

</body>
</html>

```

**참고 :** 절대 위치 요소는 정상 흐름에서 제거되며 요소와 겹칠 수 있습니다.



# 좌우 정렬-float 사용

요소를 정렬하는 또 다른 방법은 `float` 속성 을 사용하는 것입니다 .

```
<!DOCTYPE html>
<html>
<head>
<style>
.right {
  float: right;
  width: 300px;
  border: 3px solid #73AD21;
  padding: 10px;
}
</style>
</head>
<body>

<h2>Right Align</h2>
<p>An example of how to right align elements with the float property:</p>

<div class="right">
  <p>In my younger and more vulnerable years my father gave me some advice that
  I've been turning over in my mind ever since.</p>
</div>

</body>
</html>
```

**참고** : 요소가 포함 된 요소보다 키가 크고 부동 상태 인 경우 컨테이너 외부로 오버플로됩니다. " **clearfix** " 핵을 사용하여 이 문제를 해결할 수 있습니다 (아래 예 참조).

그런 다음 `overflow: auto;` 포함 요소에 추가 하여이 문제를 해결할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 3px solid #4CAF50;
  padding: 5px;
}

.img1 {
  float: right;
}

.clearfix {
```

```

    overflow: auto;
}

.img2 {
    float: right;
}

</style>
</head>
<body>

<p>In this example, the image is taller than the element containing it, and it is
floated, so it overflows outside of its container:</p>

<div class="right">
    
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet,
    nulla et dictum interdum...
</div>

<p style="clear:right">Add a clearfix class with overflow: auto; to the
containing element, to fix this problem:</p>

<div class="clearfix">
    
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet,
    nulla et dictum interdum...
</div>

</body>
</html>

```

## 세로 가운데-패딩 사용

CSS 에서 요소를 세로로 가운데 맞추는 방법에는 여러 가지가 있습니다. 간단한 해결책은 **padding top** 과 **bottom** 을 사용하는 것입니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
.center {
    padding: 70px 0;
    border: 3px solid green;
}
</style>
</head>
<body>

<h2>Center Vertically</h2>
<p>In this example, we use the padding property to center the div element
vertically:</p>

```

```
<div class="center">
  <p>I am vertically centered.</p>
</div>

</body>
</html>
```

`padding`하고 `text-align: center` 수직 및 수평으로 모두의 중심을 사용:

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  padding: 70px 0;
  border: 3px solid green;
  text-align: center;
}
</style>
</head>
<body>

<h2>Centering</h2>
<p>In this example, we use padding and text-align to center the div element
vertically and horizontally:</p>

<div class="center">
  <p>I am vertically and horizontally centered.</p>
</div>

</body>
</html>
```

## 세로 중심-선 높이 사용

또 다른 속임수는 `line-height` 속성과 동일한 값으로 `height` 속성 을 사용하는 것입니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  line-height: 200px;
  height: 200px;
  border: 3px solid green;
  text-align: center;
}

/* If the text has multiple lines, add the following: */
```

```

.center p {
  line-height: 1.5;
  display: inline-block;
  vertical-align: middle;
}
</style>
</head>
<body>

<h2>Centering</h2>
<p>In this example, we use the line-height property with a value that is equal to
the height property to center the div element:</p>

<div class="center">
  <p>I am vertically and horizontally centered.</p>
</div>

</body>
</html>

```

## 수직 중심-위치 및 변형 사용

`padding` 와 `line-height` 없는 경우 옵션은 세 번째 솔루션은 위치와 `transform` 속성을 사용하는 :

```

<!DOCTYPE html>
<html>
<head>
<style>
.center {
  height: 200px;
  position: relative;
  border: 3px solid green;
}

.center p {
  margin: 0;
  position: absolute;
  top: 50%;
  left: 50%;
  -ms-transform: translate(-50%, -50%);
  transform: translate(-50%, -50%);
}
</style>
</head>
<body>

<h2>Centering</h2>
<p>In this example, we use positioning and the transform property to vertically
and horizontally center the div element:</p>

<div class="center">
  <p>I am vertically and horizontally centered.</p>

```

```
</div>

<p>Note: The transform property is not supported in IE8 and earlier versions.</p>

</body>
</html>
```

# CSS 조합

## CSS 조합

결합기는 선택기 간의 관계를 설명하는 것입니다.

CSS 선택기는 둘 이상의 간단한 선택기를 포함 할 수 있습니다. 간단한 선택기 사이에 결합기를 포함시킬 수 있습니다.

CSS에는 4 가지 조합이 있습니다 :

- 자손 선택기 (공백)
- 자식 선택기 (>)
- 인접한 형제 선택기 (+)
- 일반 형제 선택기 (~)

## 후손 선택기

자손 선택기는 지정된 요소의 자손 인 모든 요소와 일치합니다.

다음 예제는 <div> 요소 내의 모든 <p> 요소를 선택합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div p {
  background-color: yellow;
}
</style>
</head>
<body>
```

```
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
</div>

<p>Paragraph 4. Not in the div.</p>
<p>Paragraph 5. Not in the div.</p>

</body>
</html>
```

## 자식 선택기

자식 선택기는 지정된 요소의 자식 인 모든 요소를 선택합니다.

다음 예제는 <div> 요소의 하위 요소 인 모든 <p> 요소를 선택합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div > p {
  background-color: yellow;
}
</style>
</head>
<body>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section> <!-- This is a single-line
comment -->
  <p>Paragraph 2 in the div.</p>
</div>

<p>Paragraph 5. Not in the div.</p>
<p>Paragraph 6. Not in the div.</p>

</body>
</html>
```

## 인접 형제 선택기

인접 형제 선택기는 지정된 요소의 인접 형제 인 모든 요소를 선택합니다.

형제 요소는 동일한 부모 요소를 가져야하며 "인접한"은 "즉시 다음"을 의미합니다.

다음 예제는 `<div>` 요소 바로 뒤에 배치 된 모든 `<p>` 요소를 선택합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div + p {
  background-color: yellow;
}
</style>
</head>
<body>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
</div>

<p>Paragraph 3. Not in the div.</p>
<p>Paragraph 4. Not in the div.</p>

</body>
</html>
```

## 일반 형제 선택기

일반 형제 선택기는 지정된 요소의 형제 인 모든 요소를 선택합니다.

다음 예제는 `<div>` 요소의 형제 인 모든 `<p>` 요소를 선택합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div ~ p {
  background-color: yellow;
}
</style>
</head>
<body>

<p>Paragraph 1.</p>

<div>
  <p>Paragraph 2.</p>
</div>

<p>Paragraph 3.</p>
<code>Some code.</code>
```

```
<p>Paragraph 4.</p>
```

```
</body>  
</html>
```

# CSS 의사 클래스

## 의사 클래스란 무엇입니까?

의사 클래스는 요소의 특수 상태를 정의하는 데 사용됩니다.

예를 들어 다음과 같은 용도로 사용할 수 있습니다.

- 사용자가 마우스를 가져 가면 요소 스타일 지정
- 방문 및 방문하지 않은 링크의 스타일이 다르게
- 초점을 맞출 때 요소 스타일 지정

## 통사론

의사 클래스의 구문 :

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

## 앵커 슈도 클래스

링크는 다른 방식으로 표시 될 수 있습니다.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
/* unvisited link */  
a:link {  
  color: #FF0000;
```



```

}

/* visited link */
a:visited {
    color: #00FF00;
}

/* mouse over link */
a:hover {
    color: #FF00FF;
}

/* selected link */
a:active {
    color: #0000FF;
}
</style>
</head>
<body>

<p><b><a href="#" target="_blank"> This is a link</a></b></p>
<p><b> Note:</b>a:hover MUST come after a:link and a:visited in the CSS definition
in order to be effective.</p>
<p><b> Note:</b>a:active MUST come after a:hover in the CSS definition in order to
be effective.</p>

</body>
</html>

```

**주 :** `a:hover` 뒤에 와야합니다 `a:link` 및 `a:visited` 위해 CSS 정의에하는 것이 효과적! 효과적이기 위해서는 `a:active` 반드시 `a:hover` CSS 정의를 따라야합니다! 의사 클래스 이름은 대소 문자를 구분하지 않습니다.

## 의사 클래스 및 CSS 클래스

의사 클래스는 CSS 클래스와 결합 될 수 있습니다.

예제에서 링크 위로 마우스를 가져 가면 색상이 변경됩니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
a.highlight:hover {
    color: #ff0000;
}
</style>
</head>
<body>

```

```
<p><a href="#" class="highlight">CSS Syntax</a></p>
<p><a href="#">CSS Tutorial</a></p>

</body>
</html>
```

## <div>에 마우스를 올리십시오.

<div> 요소 에서 `:hover` 의사 클래스를 사용하는 예 :

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: green;
  color: white;
  padding: 25px;
  text-align: center;
}
div:hover {
  background-color: blue;
}
</style>
</head>
<body>

<p>Mouse over the div element below to change its background color:</p>
<div>Mouse Over Me</div>

</body>
</html>
```

## 간단한 툴팁 호버

<div> 요소 위로 마우스를 가져 가면 툴팁과 같은 <p> 요소가 표시됩니다.

<p> 요소를 표시하려면 나를 가리 킵니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  display: none;
  background-color: yellow;
  padding: 20px;
}
```

```

}

div:hover p {
  display: block;
}
</style>
</head>
<body>

<div>Hover over me to show the p element
  <p>Tada! Here I am!</p>
</div>

</body>
</html>

```

## CSS-첫 번째 하위 클래스

`:first-child` 의사 클래스는 다른 요소의 첫 번째 자식 인 지정된 요소와 일치합니다.

## 첫 번째 <p> 요소와 일치

다음 예제에서 선택기는 요소의 첫 번째 자식 인 <p> 요소와 일치합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
p:first-child {
  color: blue;
}
</style>
</head>
<body>

<p>This is some text.</p>
<p>This is some text.</p>
<p><b> Note:</b>For :first-child to work in IE8 and earlier, a DOCTYPE must be
declared.</p>

</body>
</html>

```

## 모든 <p> 요소에서 첫 번째 <i> 요소와

# 일치

다음 예에서 선택기는 모든 `<p>` 요소의 첫 번째 `<i>` 요소와 일치합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p i:first-child {
  color: blue;
}
</style>
</head>
<body>

<p>I am a <i>strong</i> person. I am a <i>strong</i> person.</p>
<p>I am a <i>strong</i> person. I am a <i>strong</i> person.</p>
<p><b> Note:</b>For :first-child to work in IE8 and earlier, a DOCTYPE must be
declared.</p>

</body>
</html>
```

## 모든 첫 번째 자식 `<p>` 요소의 모든 `<i>` 요소와 일치

다음 예에서 선택기는 다른 요소의 첫 번째 자식 인 `<p>` 요소의 모든 `<i>` 요소와 일치합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p:first-child i {
  color: blue;
}
</style>
</head>
<body>

<p>I am a <i>strong</i> person. I am a <i>strong</i> person.</p>
<p>I am a <i>strong</i> person. I am a <i>strong</i> person.</p>
<p><b> Note:</b>For :first-child to work in IE8 and earlier, a DOCTYPE must be
declared.</p>

</body>
```

```
</html>
```

## CSS-: 의사 클래스

**:lang** 의사 클래스는 다른 언어에 대한 특별한 규칙을 정의 할 수 있습니다.

아래 예 **:lang** 에서 lang = "no"를 사용하여 <q> 요소의 인용 부호를 정의하십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
q:lang(no) {
  quotes: "~" "~";
}
</style>
</head>
<body>

<p>Some text <q lang="no">A quote in a paragraph</q> Some text.</p>
<p>In this example, :lang defines the quotation marks for q elements with
lang="no":</p>
<p><b> Note:</b>IE8 supports the :lang pseudo class only if a !DOCTYPE is
specified.</p>

</body>
</html>
```

## 모든 CSS 의사 클래스

# CSS 이미지 갤러리

CSS는 이미지 갤러리를 만드는 데 사용될 수 있습니다.

## 이미지 갤러리

다음 이미지 갤러리는 CSS로 작성됩니다.

```
<html>
<head>
<style>
div.gallery {
  margin: 5px;
  border: 1px solid #ccc;
  float: left;
  width: 180px;
}

div.gallery:hover {
  border: 1px solid #777;
}

div.gallery img {
  width: 100%;
  height: auto;
}

div.desc {
  padding: 15px;
  text-align: center;
}
</style>
</head>
<body>

<div class="gallery">
  <a target="_blank" href="img_5terre.jpg">
    
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

<div class="gallery">
  <a target="_blank" href="img_forest.jpg">
    
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

<div class="gallery">
```

```

    <a target="_blank" href="img_lights.jpg">
      
    </a>
    <div class="desc">Add a description of the image here</div>
  </div>

  <div class="gallery">
    <a target="_blank" href="img_mountains.jpg">
      
    </a>
    <div class="desc">Add a description of the image here</div>
  </div>

</body>
</html>

```

## 반응형 이미지 갤러리

CSS 미디어 쿼리를 사용하여 데스크톱, 태블릿 및 스마트폰에서 잘 보이는 반응형 이미지 갤러리를 만드는 방법

```

<html>
<head>
<style>
div.gallery {
  border: 1px solid #ccc;
}

div.gallery:hover {
  border: 1px solid #777;
}

div.gallery img {
  width: 100%;
  height: auto;
}

div.desc {
  padding: 15px;
  text-align: center;
}

div.note {
  padding: 6px;
}

* {
  box-sizing: border-box;
}

.responsive {

```

```

width: 24.99999%;
padding: 0 6px;
float: left;
}

@media only screen and (max-width: 700px) {
  .responsive {
    width: 49.99999%;
    margin: 6px 0;
  }
}

@media only screen and (max-width: 500px) {
  .responsive {
    width: 100%;
  }
}

.clearfix::after {
  content: "";
  display: table;
  clear: both;
}

</style>
</head>
<body>

<div class="responsive">
  <div class="gallery">
    <a target="_blank" href="img_5terre.jpg">
      
    </a>
    <div class="desc">Add a description of the image here</div>
  </div>
</div>

<div class="responsive">
  <div class="gallery">
    <a target="_blank" href="img_forest.jpg">
      
    </a>
    <div class="desc">Add a description of the image here</div>
  </div>
</div>

<div class="responsive">
  <div class="gallery">
    <a target="_blank" href="img_lights.jpg">
      
    </a>
    <div class="desc">Add a description of the image here</div>
  </div>
</div>

<div class="responsive">
  <div class="gallery">
    <a target="_blank" href="img_mountains.jpg">

```



```

        
    </a>
    <div class="desc">Add a description of the image here</div>
</div>
</div>

<div class="clearfix"></div>

<div class="note">
    <p>This example use media queries to re-arrange the images on different screen
    sizes: for screens larger than 700px wide, it will show four images side by side,
    for screens smaller than 700px, it will show two images side by side. For screens
    smaller than 500px, the images will stack vertically (100%).</p>
    <p>You will learn more about media queries and responsive web design later in
    our CSS Tutorial.</p>
</div>

</body>
</html>

```

## CSS 이미지 스프라이트

### 이미지 스프라이트

이미지 스프라이트는 단일 이미지에 넣은 이미지 모음입니다.

이미지가 많은 웹 페이지는 여러 서버 요청을 로드하고 생성하는 데 시간이 오래 걸릴 수 있습니다.

이미지 스프라이트를 사용하면 서버 요청 수가 줄어들고 대역폭이 절약됩니다.

### 이미지 스프라이트-간단한 예

세 개의 별도 이미지를 사용하는 대신 이 단일 이미지 ("img\_navsprites.gif")를 사용합니다.



CSS 를 사용하면 필요한 이미지의 일부만 표시 할 수 있습니다.

다음 예제에서 CSS 는 "img\_navsprites.gif" 이미지에서 표시 할 부분을 지정합니다.

```

<html>
<head>
<style>
#home {
  width: 46px;
  height: 44px;
  background: url(img_navsprites.gif) 0 0;
}
#next {
  width: 43px;
  height: 44px;
  background: url(img_navsprites.gif) -91px 0;
}
</style>
</head>
<body>

<br><br>


</body>
</html>

```

예 설명 :

- ``-src 속성은 비워 둘 수 없으므로 작은 투명 이미지만 정의합니다. 표시된 이미지는 CSS 에서 지정한 배경 이미지가 됩니다.
- `width: 46px; height: 44px;` -사용하려는 이미지 부분을 정의합니다.
- `background: url(img_navsprites.gif) 0 0;` -배경 이미지 및 위치를 정의합니다 (왼쪽 0px, 위쪽 0px)

이미지 스프라이트를 사용하는 가장 쉬운 방법입니다. 이제 링크와 호버 효과를 사용하여 이미지 스프라이트를 확장하려고 합니다.

## 이미지 스프라이트-탐색 목록 만들기

스프라이트 이미지 ( "img\_navsprites.gif")를 사용하여 탐색 목록을 만들고 싶습니다.

링크 일 수 있고 배경 이미지도 지원하므로 HTML 목록을 사용합니다.

```

<html>
<head>
<style>

```

```

#navlist {
    position: relative;
}

#navlist li {
    margin: 0;
    padding: 0;
    list-style: none;
    position: absolute;
    top: 0;
}

#navlist li, #navlist a {
    height: 44px;
    display: block;
}

#home {
    left: 0px;
    width: 46px;
    background: url('img_navsprites.gif') 0 0;
}

#prev {
    left: 63px;
    width: 43px;
    background: url('img_navsprites.gif') -47px 0;
}

#next {
    left: 129px;
    width: 43px;
    background: url('img_navsprites.gif') -91px 0;
}
</style>
</head>
<body>

<ul id="navlist">
    <li id="home"><a href="default.asp"></a></li>
    <li id="prev"><a href="css_intro.asp"></a></li>
    <li id="next"><a href="css_syntax.asp"></a></li>
</ul>

</body>
</html>

```

#### 예 설명 :

- #navlist {position : relative;}-내부에 절대 위치를 지정할 수 있도록 위치가 상대적으로 설정됩니다.
- #navlist li {margin : 0; padding : 0; list-style : none; position : absolute; top : 0;}-여백 및 패딩이 0 으로 설정되고 목록 스타일이 제거되며 모든 목록 항목이 절대 위치

- #navlist li, #navlist a {height : 44px; display : block;}-모든 이미지의 높이는 44px 입니다

이제 각 특정 부품의 위치와 스타일을 시작하십시오.

- #home {left : 0px; width : 46px;}-왼쪽 끝까지 배치되며 이미지 너비는 46px
- #home {background : url (img\_navsprites.gif) 0 0;}-배경 이미지와 위치를 정의합니다 (왼쪽 0px, 위쪽 0px)
- #prev {left : 63px; width : 43px;}-오른쪽 63px (#home width 46px + 항목 사이의 추가 공간)이며 너비는 43px 입니다.
- #prev {background : url ( 'img\_navsprites.gif') -47px 0;}-배경 이미지를 오른쪽으로 47px 정의합니다 (#home width 46px + 1px line splitter)
- #next {left : 129px; width : 43px;}-오른쪽에 129px (#prev 의 시작은 63px + #prev 너비 43px + 추가 공간)이며 너비는 43px 입니다.
- #next {background : url ( 'img\_navsprites.gif') -91px 0;}-배경 이미지를 오른쪽으로 91px 정의합니다 (#home width 46px + 1px line divider + #prev width 43px + 1px line divider)

## 이미지 스프라이트-호버 효과

이제 탐색 목록에 호버 효과를 추가하려고 합니다.

**팁 ::hover** 선택은 링크뿐만 아니라, 모든 요소에 사용할 수 있습니다.

새로운 이미지 ( "img\_navsprites\_hover.gif")에는 호버 효과에 사용할 3 가지 탐색 이미지와 3 가지 이미지가 포함되어 있습니다.



이 파일은 6 개의 개별 파일이 아닌 하나의 단일 이미지이므로 사용자가 이미지를 가리킬 때 로드 지연 이 없습니다.

호버 효과를 추가하려면 세 줄의 코드 만 추가하십시오.

```
<html>
<head>
```

```

<style>
#navlist {
    position: relative;
}

#navlist li {
    margin: 0;
    padding: 0;
    list-style: none;
    position: absolute;
    top: 0;
}

#navlist li, #navlist a {
    height: 44px;
    display: block;
}

#home {
    left: 0px;
    width: 46px;
    background: url('img_navsprites.gif') 0 0;
}

#prev {
    left: 63px;
    width: 43px;
    background: url('img_navsprites.gif') -47px 0;
}

#next {
    left: 129px;
    width: 43px;
    background: url('img_navsprites.gif') -91px 0;
}

#home a:hover {
    background: url('img_navsprites_hover.gif') 0 -45px;
}

#prev a:hover {
    background: url('img_navsprites_hover.gif') -47px -45px;
}

#next a:hover {
    background: url('img_navsprites_hover.gif') -91px -45px;
}
</style>
</head>
<body>

<ul id="navlist">
    <li id="home"><a href="default.asp"></a></li>
    <li id="prev"><a href="css_intro.asp"></a></li>
    <li id="next"><a href="css_syntax.asp"></a></li>
</ul>

</body>

```

```
</html>
```

예 설명 :

- #home a : 호버 {background : transparent url ( 'img\_navsprites\_hover.gif') 0 -45px;}-3 개의 호버 이미지 모두 동일한 배경 위치를 지정합니다

## CSS 속성 선택기

### 특정 속성을 가진 스타일 HTML 요소

특정 속성 또는 속성 값이 있는 HTML 요소의 스타일을 지정할 수 있습니다.

## CSS [속성] 선택기

**[attribute]** 선택기는 특정 속성을 가진 요소를 선택하는 데 사용됩니다.

다음 예제는 대상 속성이 있는 모든 `<a>` 요소를 선택합니다.

```
<html>
<head>
<style>
a[target] {
  background-color: yellow;
}
</style>
</head>
<body>

<p>The links with a target attribute gets a yellow background:</p>

<a href="https://www.w3schools.com">w3schools.com</a>
<a href="http://www.disney.com" target="_blank">disney.com</a>
<a href="http://www.wikipedia.org" target="_top">wikipedia.org</a>

<p><b>Note:<b>For [<i>attribute<i>] to work in IE8 and earlier, a DOCTYPE must be
declared.</p>

</body>
</html>
```

# CSS [attribute = "value"] 선택기

[attribute="value"] 선택기는 특정 속성 및 값을 가진 요소를 선택하는 데 사용된다.

다음 예제는 target = "\_blank" 속성이 있는 모든 <a> 요소를 선택합니다.

```
<html>
<head>
<style>
a[target="_blank"] {
  background-color: yellow;
}
</style>
</head>
<body>

<p>The link with target="_blank" gets a yellow background:</p>

<a href="https://www.w3schools.com">w3schools.com</a>
<a href="http://www.disney.com" target="_blank">disney.com</a>
<a href="http://www.wikipedia.org" target="_top">wikipedia.org</a>

<p><b>Note:</b>For [attribute] to work in IE8 and earlier, a DOCTYPE must be
declared.</p>

</body>
</html>
```

# CSS [속성 ~ = "값"] 선택기

[attribute~="value"] 선택기는 특정 단어를 포함하는 속성 값을 가진 요소를 선택하는 데 사용된다.

다음 예는 공백으로 구분된 단어 목록 (하나는 "flower")을 포함하는 title 속성을 가진 모든 요소를 선택합니다.

```
<html>
<head>
<style>
[title~="flower"] {
  border: 5px solid yellow;
}
</style>
</head>
<body>

<p>All images with the title attribute containing the word "flower" get a yellow
```

```
border.</p>
```

```




</body>
</html>
```

위의 예는 title = "flower", title = "summer flower" 및 title = "flower new"는 포함하지만 title = "my-flower" 또는 title = "flowers"는 포함하지 않는 요소와 일치합니다.

## CSS [attribute | = "value"] 선택기

[attribute | = "value"] 선택기 지정된 값으로 시작하는 지정된 속성 요소를 선택하는 데 사용됩니다.

다음 예제는 "top"으로 시작하는 클래스 속성 값을 가진 모든 요소를 선택합니다.

**참고 :** 값은 class = "top"과 같은 단독 단어이거나 class = "top-text"와 같은 하이픈 (-)이어야 합니다!

```
<html>
<head>
<style>
[class|="top"] {
    background: yellow;
}
</style>
</head>
<body>

<h1 class="top-header">Welcome</h1>
<p class="top-text">Hello world!</p>
<p class="topcontent">Are you learning CSS?</p>

<p><b>Note:<b>For [<i>attribute</i>|=<i>value</i>] to work in IE8 and earlier, a
DOCTYPE must be declared.</p>

</body>
</html>
```



# CSS [attribute ^ = "value"] 선택기

[attribute^="value"] 선택기 속성 값이 지정된 값으로 시작 요소를 선택하는데 사용된다.

다음 예제는 "top"으로 시작하는 클래스 속성 값을 가진 모든 요소를 선택합니다.

**참고 :** 값이 전체 단어 일 필요는 없습니다!

```
<html>
<head>
<style>
[class^="top"] {
  background: yellow;
}
</style>
</head>
<body>

<h1 class="top-header">Welcome</h1>
<p class="top-text">Hello world!</p>
<p class="topcontent">Are you learning CSS?</p>

<p><b>Note:<b>For [<i>attribute</i>^=<i>value</i>] to work in IE8 and earlier, a
DOCTYPE must be declared.</p>

</body>
</html>
```

# CSS [attribute \$ = "value"] 선택기

[attribute\$="value"] 선택기 속성 값이 지정된 값으로 끝나는 요소를 선택하는데 사용된다.

다음 예제는 "test"로 끝나는 클래스 속성 값을 가진 모든 요소를 선택합니다.

**참고 :** 값이 전체 단어 일 필요는 없습니다!

```
<html>
<head>
<style>
[class$="test"] {
  background: yellow;
}
</style>
</head>
<body>
```

```
<div class="first_test">The first div element.</div>
<div class="second">The second div element.</div>
<div class="my-test">The third div element.</div>
<p class="mytext">This is some text in a paragraph.</p>

</body>
</html>
```

## CSS [attribute \* = "value"] 선택기

[attribute\*="value"] 선택기 속성 값이 소정 값을 포함하는 요소를 선택하는 데 사용된다.

다음 예제는 "te"를 포함하는 클래스 속성 값을 가진 모든 요소를 선택합니다.

**참고 :** 값이 전체 단어 일 필요는 없습니다!

```
<html>
<head>
<style>
[class*="te"] {
    background: yellow;
}
</style>
</head>
<body>

<div class="first_test">The first div element.</div>
<div class="second">The second div element.</div>
<div class="my-test">The third div element.</div>
<p class="mytext">This is some text in a paragraph.</p>

</body>
</html>
```

## 스타일링 양식

속성 선택기는 클래스 또는 ID 가 없는 양식 양식에 유용 할 수 있습니다.

```
<html>
<head>
<style>
input[type="text"] {
    width: 150px;
    display: block;
    margin-bottom: 10px;
}
```

```

    background-color: yellow;
}

input[type="button"] {
    width: 120px;
    margin-left: 35px;
    display: block;
}
</style>
</head>
<body>

<form name="input" action="" method="get">
    Firstname:<input type="text" name="Name" value="Peter" size="20">
    Lastname:<input type="text" name="Name" value="Griffin" size="20">
    <input type="button" value="Example Button">
</form>

</body>
</html>

```

## 모든 CSS 속성 선택기

Selector	Example	Example description
<a href="#">[attribute]</a>	[target]	Selects all elements with a target attribute
<a href="#">[attribute=value]</a>	[target=_blank]	Selects all elements with target="_blank"
<a href="#">[attribute~=value]</a>	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
<a href="#">[attribute =value]</a>	[lang =en]	Selects all elements with a lang attribute value starting with "en"
<a href="#">[attribute^=value]</a>	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
<a href="#">[attribute\$=value]</a>	a[href\$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
<a href="#">[attribute*=value]</a>	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"

# CSS 양식

CSS를 사용하면 HTML 양식의 모양을 크게 향상시킬 수 있습니다.

## 스타일링 입력 필드

입력 필드의 너비를 판별 하려면 `width` 특성을 사용하십시오.

```
<html>
<head>
<style>
input {
  width: 100%;
}
</style>
</head>
<body>

<p>A full-width input field:</p>
<form>
  <label for="fname">First Name</label>
  <input type="text" name="Name" id="fname">
</form>

</body>
</html>
```

위의 예는 모든 `<input>` 요소에 적용됩니다. 특정 입력 유형 만 스타일 지정하려는 경우 속성 선택기를 사용할 수 있습니다.

- `input[type=text]` -텍스트 필드 만 선택합니다
- `input[type=password]` -비밀번호 필드 만 선택합니다
- `input[type=number]` -숫자 필드 만 선택합니다
- 기타..

## 패딩 된 입력

이 `padding` 속성을 사용하여 텍스트 필드 안에 공백을 추가 하십시오 .

**팁 :** 서로 입력이 많은 경우 **margin** 외부 추가 공간을 추가 하기 위해 일부를 추가 할 수도 있습니다.

```
<html>
<head>
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
}
</style>
</head>
<body>

<p>Padded text fields:</p>
<form>
  <label for="fname">First Name</label>
  <input type="text" name="fname" id="fname">
  <label for="lname">Last Name</label>
  <input type="text" name="lname" id="lname">
</form>

</body>
</html>
```

**box-sizing** 속성을 **border-box** 로 설정 했습니다. 이렇게 하면 패딩 및 테두리가 요소의 총 너비와 높이에 포함됩니다. [CSS Box Sizing](#) 장 에서 **box-sizing** 속성에 대해 자세히 알아보십시오.

## 경계 입력

**border** 속성을 사용하여 테두리 크기와 색상을 변경 하고 **border-radius** 속성을 사용하여 둥근 모서리를 추가하십시오.

```
<html>
<head>
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
  border: 2px solid red;
  border-radius: 4px;
}
</style>
</head>
```

```

<body>

<p> Text fields with borders:</p>

<form>
  <label for="fname">First Name</label>
  <input type="text" name="fname" id="fname">
  <label for="lname">Last Name</label>
  <input type="text" name="lname" id="lname">
</form>

</body>
</html>

```

아래쪽 테두리 만 원하는 경우 다음 `border-bottom`속성을 사용하십시오 .

```

<html>
<head>
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
  border: none;
  border-bottom: 2px solid red;
}
</style>
</head>
<body>

<p> Text fields with only a bottom border:</p>

<form>
  <label for="fname">First Name</label>
  <input type="text" name="fname" id="fname">
  <label for="lname">Last Name</label>
  <input type="text" name="lname" id="lname">
</form>

</body>
</html>

```

## 컬러 입력

이 `background-color` 속성을 사용하여 입력에 배경색을 추가하고 `color` 속성을 사용하여 텍스트 색을 변경하십시오.

```

<html>
<head>

```

```

<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
  border: none;
  background-color: #3CBC8D;
  color: white;
}
</style>
</head>
<body>

<p>Colored text fields:</p>

<form>
  <label for="fname">First Name</label>
  <input type="text" name="fname" id="fname" value="John">
  <label for="lname">Last Name</label>
  <input type="text" name="lname" id="lname" value="Doe">
</form>

</body>
</html>

```

## 집중된 입력

기본적으로 일부 브라우저는 입력이 집중 될 때 입력 주위에 파란색 윤곽선을 추가합니다 (클릭). `outline: none;` 입력에 추가하여 이 동작을 제거 할 수 있습니다 .

`:focus` 선택기를 사용하여 입력 필드에 포커스가 있을 때 무언가를 수행 하십시오 .

```

<html>
<head>
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
  border: 1px solid #555;
  outline: none;
}

input[type=text]:focus {
  background-color: lightblue;
}
</style>
</head>
<body>

```

<p>In this example, we use the :focus selector to add a background color to the text field when it gets focused (clicked on):</p>

```
<form>
  <label for="fname">First Name</label>
  <input type="text" name="fname" id="fname" value="John">
  <label for="lname">Last Name</label>
  <input type="text" name="lname" id="lname" value="Doe">
</form>

</body>
</html>
```

```
<html>
<head>
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
  border: 3px solid #ccc;
  -webkit-transition: 0.5s;
  transition: 0.5s;
  outline: none;
}

input[type=text]:focus {
  border: 3px solid #555;
}
</style>
</head>
<body>
```

<p>In this example, we use the :focus selector to add a black border color to the text field when it gets focused (clicked on):</p>

<p>Note that we have added the CSS transition property to animate the border color (takes 0.5 seconds to change the color on focus).</p>

```
<form>
  <label for="fname">First Name</label>
  <input type="text" name="fname" id="fname" value="John">
  <label for="lname">Last Name</label>
  <input type="text" name="lname" id="lname" value="Doe">
</form>

</body>
</html>
```



# 아이콘 / 이미지로 입력

입력 내부에 아이콘이 필요하면 `background-image` 속성을 사용하고 `background-position` 속성과 함께 배치하십시오. 또한 아이콘의 공간을 확보하기 위해 큰 왼쪽 패딩을 추가합니다.

```
<html>
<head>
<style>
input[type=text] {
  width: 100%;
  font-size: 16px;
  padding: 12px 20px 12px 40px;
  box-sizing: border-box;
  border: 2px solid #ccc;
  border-radius: 4px;
  background-color: white;
  background-image: url('searchicon.png');
  background-position: 10px 10px;
  background-repeat: no-repeat;
}
</style>
</head>
<body>

<p> Input with icon:</p>

<form>
  <input type="text" name="search" placeholder="Search..">
</form>

</body>
</html>
```

# 애니메이션 검색 입력

이 예제에서는 CSS `transition` 속성을 사용하여 포커스가 있을 때 검색 입력의 너비에 애니메이션을 적용합니다. [CSS 전환](#) 장 `transition` 에서 나중에 속성 에 대해 자세히 알아볼 것 입니다.

```
<html>
<head>
<style>
input[type=text] {
  width: 130px;
  font-size: 16px;
  padding: 12px 20px 12px 40px;
  box-sizing: border-box;
```

```

border: 2px solid #ccc;
border-radius: 4px;
background-color: white;
background-image: url('searchicon.png');
background-position: 10px 10px;
background-repeat: no-repeat;
-webkit-transition: width 0.4s ease-in-out;
transition: width 0.4s ease-in-out;
}

input[type=text]:focus {
  width: 100%;
}
</style>
</head>
<body>

<p>Animated search input:</p>

<form>
  <input type="text" name="search" placeholder="Search..">
</form>

</body>
</html>

```

## 스타일링 텍스트 영역

팁 : 이 **resize** 속성을 사용하여 텍스트 영역의 크기가 조정되지 않도록 하십시오 (오른쪽 하단에 있는 "그래버"사용 안함).

```

<html>
<head>
<style>
textarea {
  width: 100%;
  height: 150px;
  padding: 12px 20px;
  box-sizing: border-box;
  border: 2px solid #ccc;
  border-radius: 4px;
  background-color: #f8f8f8;
  resize: none;
  font-size: 16px;
}
</style>
</head>
<body>

<p><strong>Tip:</strong>Use the resize property to prevent textareas from being
resized (disable the "grabber" in the bottom right corner):</p>

```

```
<form>
  <textarea>Some text...</textarea>
</form>

</body>
</html>
```

## 스타일링 선택 메뉴

```
<html>
<head>
<style>
select {
  width: 100%;
  padding: 16px 20px;
  border: none;
  border-radius: 4px;
  background-color: #f1f1f1;
}
</style>
</head>
<body>

<p>A styled select menu.</p>

<form>
  <select id="country" name="country">
    <option value="au">Australia</option>
    <option value="ca">Canada</option>
    <option value="usa">USA</option>
  </select>
</form>

</body>
</html>
```

## 스타일링 입력 버튼

```
<html>
<head>
<style>
input[type=button], input[type=submit], input[type=reset] {
  background-color: #4CAF50;
  border: none;
  color: white;
  padding: 16px 32px;
  text-decoration: none;
  margin: 4px 2px;
  cursor: pointer;
}
}
```

```

/* Tip: use width: 100% for full-width buttons */
</style>
</head>
<body>

<p>Styled input buttons.</p>

<input type="button" value="Button">
<input type="reset" value="Reset">
<input type="submit" value="Submit">

</body>
</html>

```

## 반응 형

효과를 보려면 브라우저 창의 크기를 조정하십시오. 화면의 너비가 600px 미만인 경우 두 열을 나란히 놓지 말고 쌓으십시오.

**고급 :** 다음 예는 [미디어 쿼리](#) 를 사용하여 반응 형 양식을 만듭니다. 이에 대한 자세한 내용은 다음 장에서 설명합니다.

```

<html>
<head>
<style>
* {
  box-sizing: border-box;
}
input[type=text], select, textarea {
  width: 100%;
  border: 1px solid #ccc;
  padding: 12px;
  border-radius: 4px;
  resize: vertical;
}

label {
  display: inline-block;
  padding: 12px 12px 12px 0;
}

input[type=submit] {
  background-color: #4caf50;
  color: white;
  border: none;
  padding: 12px 20px;
  border-radius: 4px;
  cursor: pointer;
  float: right;
}
input[type=submit]:hover {

```

```

    background-color: #45a049;
}
.container {
    background-color: #f2f2f2;
    padding: 20px;
    border-radius: 5px;
}
.col-25 {
    float: left;
    width: 25%;
    margin-top: 6px;
}
.col-75 {
    float: left;
    width: 75%;
    margin-top: 6px;
}
/* Clear floats after the columns */
.row:after {
    content: "";
    display: table;
    clear: both;
}
/* Responsive layout - when the screen is less than 600px wide, make the two
columns stack on top of each other instead of next to each other */
@media screen and (max-width: 600px) {
    .col-25, .col-75, input[type=submit] {
        width: 100%;
        margin-top: 0;
    }
}
</style>
</head>
<body>

<h2>Responsive Form</h2>
<p>Resize the browser window to see the effect. When the screen is less than
600px wide, make the two columns stack on top of each other instead of next to
each other.</p>

<div class="container">
    <form action="/action_page.php">
        <div class="row">
            <div class="col-25">
                <label for="fname">First Name</label>
            </div>
            <div class="col-75">
                <input type="text" id="fname" name="firstname" placeholder="Your name..">
            </div>
        </div>
        <div class="row">
            <div class="col-25">
                <label for="lname">Last Name</label>
            </div>
            <div class="col-75">
                <input type="text" id="lname" name="lastname" placeholder="Your last
name..">
            </div>
        </div>
    </form>
</div>

```

```

</div>
<div class="row">
  <div class="col-25">
    <label for="country">Country</label>
  </div>
  <div class="col-75">
    <select id="country" name="country">
      <option value="australia">Australia</option>
      <option value="canada">Canada</option>
      <option value="usa">USA</option>
    </select>
  </div>
</div>
<div class="row">
  <div class="col-25">
    <label for="subject">Subject</label>
  </div>
  <div class="col-75">
    <textarea id="subject" name="subject" placeholder="Write something.."
style="height:200px"></textarea>
  </div>
</div>
<div class="row">
  <input type="submit" value="submit">
</div>
</form>
</div>

</body>
</html>

```

## CSS 카운터

CSS 카운터는 CSS 규칙에 의해 값이 증가 될 수있는 CSS에 의해 유지되는 "변수"입니다 (사용 횟수를 추적하기 위해). 카운터를 사용하면 문서에서의 위치에 따라 내용의 모양을 조정할 수 있습니다.

## 카운터 자동 번호 매기기

CSS 카운터는 "변수"와 같습니다. 변수 값은 CSS 규칙에 의해 증가 될 수 있습니다 (이는 사용 횟수를 추적합니다).

CSS 카운터로 작업하기 위해 다음 속성을 사용합니다.

- **counter-reset** -카운터 생성 또는 재설정

- `counter-increment` -카운터 값을 증가시킵니다
- `content` -생성 된 콘텐츠 삽입
- `counter()`또는 `counters()`함수-카운터 값을 요소에 추가

CSS 카운터를 사용하려면 먼저 `counter-reset` 로 카운터를 작성해야 합니다.

다음 예제에서는 본문 선택기에서 페이지에 대한 카운터를 만든 다음 각 `<h2>` 요소의 카운터 값을 증가시키고 각 `<h2>` 요소의 시작 부분에 "섹션 < 카운터의 값 > :"을 추가합니다.

```
<html>
<head>
<style>
body {
  counter-reset: section;
}

h2::before {
  counter-increment: section;
  content: "Section " counter(section) ": ";
}
</style>
</head>
<body>

<h1>Using CSS Counters:</h1>
<h2>HTML Tutorial</h2>
<h2>CSS Tutorial</h2>
<h2>JavaScript Tutorial</h2>

<p><b>Note:<b>IE8 supports these properties only if a !DOCTYPE is specified.</p>

</body>
</html>
```

## 중첩 카운터

다음 예제는 페이지 (섹션)에 대한 카운터 하나와 각 `<h1>` 요소 (하위 섹션)에 대해 하나의 카운터를 만듭니다. "섹션"카운터는 "섹션 카운터 섹션의 섹션 값 >"으로 각 `<h1>` 요소에 대해 계산되며, "< 섹션"카운터는 "< 섹션 카운터 값이 있는 각 `<h2>` 요소에 대해 카운트됩니다. >. < 서브 섹션 카운터 값 > ":

```
<html>
<head>
<style>
body {
  counter-reset: section;
}
```

```

h1 {
  counter-reset: subsection;
}

h1::before {
  counter-increment: section;
  content: "Section " counter(section) ". ";
}

h2::before {
  counter-increment: subsection;
  content: counter(section) "." counter(subsection) " ";
}
</style>
</head>
<body>

<h1>HTML tutorials:</h1>
<h2>HTML Tutorial</h2>
<h2>CSS Tutorial</h2>

<h1>Scripting tutorials:</h1>
<h2>JavaScript</h2>
<h2>VBScript</h2>

<h1>XML tutorials:</h1>
<h2>XML</h2>
<h2>XSL</h2>

<p><b>Note:<b>IE8 supports these properties only if a !DOCTYPE is specified.</p>

</body>
</html>

```

카운터의 새 인스턴스가 자식 요소에 자동으로 생성되므로 카운터는 개요 목록을 만드는 데 유용 할 수도 있습니다. 여기서는 `counters()` 함수를 사용하여 서로 다른 수준의 중첩 카운터 사이에 문자열을 삽입합니다.

```

<html>
<head>
<style>
ol {
  counter-reset: section;
  list-style-type: none;
}

li::before {
  counter-increment: section;
  content: counters(section, ".") " ";
}
</style>
</head>
<body>

```



```
<ol>
  <li>item</li>
  <li>item
    <ol>
      <li>item</li>
      <li>item</li>
      <li>item
        <ol>
          <li>item</li>
          <li>item</li>
          <li>item</li>
        </ol>
      </li>
    </ol>
  </li>
  <li>item</li>
  <li>item</li>
</ol>

<ol>
  <li>item</li>
  <li>item</li>
</ol>

<p><b>Note:<b>IE8 supports these properties only if a !DOCTYPE is specified.</p>

</body>
</html>
```

## CSS 카운터 속성

Property	Description
<a href="#">content</a>	Used with the ::before and ::after pseudo-elements, to insert generated content
<a href="#">counter-increment</a>	Increments one or more counter values
<a href="#">counter-reset</a>	Creates or resets one or more counters

## CSS 웹 사이트 레이아웃

### 웹 사이트 레이아웃

웹 사이트는 종종 머리글, 메뉴, 내용 및 바닥 글로 나뉩니다.

# 헤더

헤더는 일반적으로 웹 사이트 상단 (또는 상단 탐색 메뉴 바로 아래)에 있습니다. 종종 로고 또는 웹 사이트 이름이 포함됩니다.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  margin: 0;
}
/* Style the header */
.header {
  background-color: #F1F1F1;
  text-align: center;
  padding: 20px;
}
</style>
</head>
<body>

<div class="header">
  <h1>Header</h1>
</div>

</body>
</html>
```

# 네비게이션 바

탐색 막대에는 방문자가 웹 사이트를 탐색하는 데 도움이 되는 링크 목록이 포함되어 있습니다.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}
body {
```

```

    margin: 0;
}

/* Style the header */
.header {
    background-color: #F1F1F1;
    text-align: center;
    padding: 20px;
}

/* Style the top navigation bar */
.topnav {
    overflow: hidden;
    background-color: #333;
}

/* Style the topnav links */
.topnav a {
    float: left;
    display: block;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
    background-color: #ddd;
    color: black;
}
</style>
</head>
<body>

<div class="header">
    <h1>Header</h1>
</div>

<div class="topnav">
    <a href="#">Link</a>
    <a href="#">Link</a>
    <a href="#">Link</a>
</div>

</body>
</html>

```

## 컨텐츠

이 섹션의 레이아웃은 종종 대상 사용자에 따라 다릅니다. 가장 일반적인 레이아웃은 다음 중 하나입니다 (또는 조합).

- **1 열** (종종 모바일 브라우저에 사용됨)
- **2 열** (종종 태블릿 및 노트북에 사용)
- **3 열 레이아웃** (데스크톱에만 사용)

3 열 레이아웃을 만들고 작은 화면에서 1 열 레이아웃으로 변경합니다.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 0;
}

/* Style the header */
.header {
  background-color: #F1F1F1;
  text-align: center;
  padding: 20px;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

/* Create three equal columns that floats next to each other */
.column {
  float: left;
  width: 33.33%;
  padding: 15px;
}
```

```

}

/* Clear floats after the columns */
.row:after {
    content: "";
    display: table;
    clear: both;
}

/* Responsive layout - makes the three columns stack on top of each other instead
of next to each other on smaller screens (600px wide or less) */
@media screen and (max-width: 600px) {
    .column {
        width: 100%;
    }
}
</style>
</head>
<body>

<div class="header">
    <h1>Header</h1>
    <p>Resize the browser window to see the responsive effect.</p>
</div>

<div class="topnav">
    <a href="#">Link</a>
    <a href="#">Link</a>
    <a href="#">Link</a>
</div>

<div class="row">
    <div class="column">
        <h2>Column</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet
pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna
tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt
eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan
convallis.</p>
    </div>
    <div class="column">
        <h2>Column</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet
pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna
tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt
eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan
convallis.</p>
    </div>

    <div class="column">
        <h2>Column</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet
pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna
tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt
eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan
convallis.</p>
    </div>
</div>

```

```
</body>
</html>
```

**팁 :** 2 열 레이아웃을 만들려면 너비를 50 %로 변경하십시오. 4 열 레이아웃을 만들려면 25 % 등을 사용하십시오.

**팁 :** @media 규칙이 어떻게 작동하는지 궁금하십니까? [CSS 미디어 쿼리 장에서 자세한 내용을 읽어보십시오](#) .

**팁 :** 열 레이아웃을 만드는 보다 현대적인 방법은 CSS Flexbox 를 사용하는 것입니다. 그러나 Internet Explorer 10 및 이전 버전에서는 지원되지 않습니다. IE6-10 지원이 필요한 경우, 위에 표시된 대로 float 를 사용하십시오.

플렉시블 박스 레이아웃 모듈에 대한 자세한 내용은 [CSS Flexbox 장을 참조하십시오](#) .

## 동일하지 않은 열

주요 콘텐츠는 사이트에서 가장 크고 가장 중요한 부분입니다.

열 너비 가 **같지 않은** 경우가 많으므로 대부분의 공간이 기본 콘텐츠 용으로 예약되어 있습니다. 사이드 콘텐츠 (있는 경우)는 대체 탐색으로 사용되거나 기본 콘텐츠와 관련된 정보를 지정하는 데 종종 사용됩니다. 너비를 원하는 대로 변경하십시오. 총 100 %까지 추가해야 합니다.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 0;
}

/* Style the header */
.header {
  background-color: #F1F1F1;
  text-align: center;
  padding: 20px;
}
```

```

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

/* Clear floats after the columns */
.row:after {
  content: "";
  display: table;
  clear: both;
}

/* Create three unequal columns that floats next to each other */
.column {
  float: left;
  padding: 10px;
}

/* Left and right column */
.column.side {
  width: 25%;
}

/* Middle column */
.column.middle {
  width: 50%;
}

/* Responsive layout - makes the three columns stack on top of each other instead
of next to each other */
@media screen and (max-width: 600px) {
  .column.side, .column.middle {
    width: 100%;
  }
}
</style>
</head>
<body>

<div class="header">
  <h1>Header</h1>

```

```

    <p>Resize the browser window to see the responsive effect.</p>
</div>

<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
</div>

<div class="row">
  <div class="column side">
    <h2>Side</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
  <div class="column middle">
    <h2>Main Content</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan convallis.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan convallis.</p>
  </div>
  <div class="column side">
    <h2>Side</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
</div>

</body>
</html>

```

## 바닥글

바닥 글은 페이지 하단에 있습니다. 종종 저작권 및 연락처 정보와 같은 정보가 포함됩니다.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

```



```
body {
    margin: 0;
}

/* Style the header */
.header {
    background-color: #F1F1F1;
    text-align: center;
    padding: 20px;
}

/* Style the top navigation bar */
.topnav {
    overflow: hidden;
    background-color: #333;
}

/* Style the topnav links */
.topnav a {
    float: left;
    display: block;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
    background-color: #ddd;
    color: black;
}

/* Clear floats after the columns */
.row:after {
    content: "";
    display: table;
    clear: both;
}

/* Create three unequal columns that floats next to each other */
.column {
    float: left;
    padding: 10px;
}

/* Left and right column */
.column.side {
    width: 25%;
}

/* Middle column */
.column.middle {
    width: 50%;
}

/* Responsive layout - makes the three columns stack on top of each other instead
```

```

of next to each other */
@media screen and (max-width: 600px) {
  .column.side, .column.middle {
    width: 100%;
  }
}

.footer {
  background-color: #F1F1F1;
  text-align: center;
  padding: 10px;
}
</style>
</head>
<body>

<div class="header">
  <h1>Header</h1>
  <p>Resize the browser window to see the responsive effect.</p>
</div>

<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
</div>

<div class="row">
  <div class="column side">
    <h2>Side</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
  <div class="column middle">
    <h2>Main Content</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan convallis.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan convallis.</p>
  </div>

  <div class="column side">
    <h2>Side</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
</div>

<div class="footer">
  <p>Footer</p>
</div>

</body>
</html>

```

# 반응형 웹 사이트 레이아웃

위의 CSS 코드 중 일부를 사용하여 반응형 웹 사이트 레이아웃을 만들었습니다.  
이 레이아웃은 화면 너비에 따라 두 열과 전체 열 사이에서 다릅니다.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  font-family: Arial;
  padding: 10px;
  background: #f1f1f1;
}

/* Header/Blog Title */
.header {
  background: white;
  text-align: center;
  padding: 30px;
}
.header h1 {
  font-size: 50px;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
```

```

    color: black;
}

/* Clear floats after the columns */
.row:after {
    content: "";
    display: table;
    clear: both;
}

/* Create two unequal columns that floats next to each other */
/* Left column */
.leftcolumn {
    float: left;
    width: 75%;
}

/* Right column */
.rightcolumn {
    float: left;
    width: 25%;
    background-color: #f1f1f1;
    padding-left: 20px;
}

/* Fake image */
.fakeimg {
    background-color: #aaa;
    width: 100%;
    padding: 20px;
}

/* Add a card effect for articles */
.card {
    background-color: white;
    margin-top: 20px;
    padding: 20px;
}

/* Footer */
.footer {
    background: #ddd;
    text-align: center;
    padding: 20px;
    margin-top: 20px;
}

/* Responsive layout - when the screen is less than 800px wide, make the two
columns stack on top of each other instead of next to each other */
@media screen and (max-width: 800px) {
    .leftcolumn, .rightcolumn {
        width: 100%;
        padding: 0;
    }
}

/* Responsive layout - when the screen is less than 400px wide, make the
navigation links stack on top of each other instead of next to each other */

```

```

@media screen and (max-width: 400px) {
  .leftcolumn, .rightcolumn {
    width: 100%;
    float: none;
  }
}

</style>
</head>
<body>

<div class="header">
  <h1>My Website</h1>
  <p>Resize the browser window to see the effect.</p>
</div>

<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#" style="float:right">Link</a>
</div>

<div class="row">
  <div class="leftcolumn">
    <div class="card">
      <h2>TITLE HEADING</h2>
      <h5>Title description, Dec 7, 2017</h5>
      <div class="fakeimg" style="height:200px;">Image</div>
      <p>Some text..</p>
      <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur
      adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
      aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>
    </div>
  </div>
  <div class="rightcolumn">
    <div class="card">
      <h2>About Me</h2>
      <div class="fakeimg" style="height:100px;">Image</div>
      <p>Some text about me in culpa qui officia deserunt mollit anim..</p>
    </div>
    <div class="card">
      <h3>Popular Post</h3>
      <div class="fakeimg"><p>Image</p></div>
      <div class="fakeimg"><p>Image</p></div>
      <div class="fakeimg"><p>Image</p></div>
    </div>
    <div class="card">
      <h3>Follow Me</h3>
      <p>Some text</p>
    </div>
  </div>

<div class="footer">
  <p>Footer</p>
</div>

</body>

```

</html>

# CSS 단위

## CSS 단위

CSS에는 길이를 표현하기 위한 여러 가지 단위가 있습니다.

대부분의 CSS 속성은 다음과 같은 "길이"값을 가지고 **width**, **margin**, **padding**, **font-size**, 등

길이는 숫자 뒤에 **10px**, **2em** 등과 같은 길이 단위입니다.

숫자와 장치 사이에는 공백이 나타나지 않습니다. 그러나 값이 **0** 이면 단위를 생략할 수 있습니다.

일부 CSS 속성의 경우 음수 길이가 허용됩니다.

길이 단위에는 절대 및 상대의 두 가지 유형이 있습니다.

## 절대 길이

절대 길이 단위는 고정되어 있으며 이들 중 하나로 표시된 길이는 정확히 그 크기로 나타납니다.

화면 크기는 매우 다양하므로 절대 길이 단위는 화면에서 사용하지 않는 것이 좋습니다. 그러나 인쇄 매체와 같이 출력 매체를 알고 있는 경우 사용할 수 있습니다.

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

\* 픽셀 (px)은 보기 장치를 기준으로 합니다. DPI가 낮은 장치의 경우 1px는 디스플레이의 장치 픽셀 (도트)입니다. 프린터 및 고해상도 화면의 경우 1px는 여러 장치 픽셀을 의미합니다.

# px

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  font-size: 50px;
}

h2 {
  font-size: 30px;
}

p {
  font-size: 15px;
  line-height: 20px;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

## 상대 길이

상대 길이 단위는 다른 길이 속성에 상대적인 길이를 지정합니다. 상대 길이 단위는 다른 렌더링 매체 간에 더 잘 확장됩니다.

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

**팁 :** em 및 rem 단위는 완벽하게 확장 가능한 레이아웃을 만드는 데 실용적입니다!

\* 뷰포트 = 브라우저 창 크기. 뷰포트의 너비가 50cm 인 경우 1vw = 0.5cm입니다.

## em

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  font-size: 16px;
  line-height: 2em;
}

div {
  font-size: 30px;
  border: 1px solid black;
}

span {
  font-size: 0.5em;
}
</style>
</head>
<body>

<p>These paragraphs have a calculated line-height of: 2x16px = 32px.</p>
<p>These paragraphs have a calculated line-height of: 2x16px = 32px.</p>
<p>These paragraphs have a calculated line-height of: 2x16px = 32px.</p>
<p>This is another paragraph.</p>
<div>The font-size of the div element is set to 30px. <span>The span element
inside the div element has a font-size of 0.5em, which equals to 0.5x30 =
15px</span>.</div>

</body>
</html>
```

## rem

```
<!DOCTYPE html>
<html>
<head>
<style>
html {
  font-size: 16px;
}

div {
```



```

    font-size: 3rem;
    border: 1px solid black;
}

#top-div {
    font-size: 2rem;
    border: 1px solid red;
}
</style>
</head>
<body>

<p>The font-size of this document is 16px.</p>
<div id="top-div"> The font-size of this div element is 2rem, which translates to
2 x the browser's font size.
<div>The font-size of this div element is 3rem. It also shows that it does not
inherit from its parent element font size.</div></div>
<p>The rem unit sets the font-size relative to the browsers base font-size, and
will not inherit from its parents.</p>

</body>
</html>

```

## VW

```

<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    font-size: 20vw;
}
</style>
</head>
<body>

<h1>Hello</h1>
<p>Resize the width of the browser window to see how the font-size of h1
changes.</p>
<p>1vw = 1% of viewport width.</p>
<p>The vw unit is not supported in IE8 and earlier.</p>

</body>
</html>

```

## vh

```

<!DOCTYPE html>
<html>
<head>
<style>

```

```
h1 {  
  font-size: 20vh;  
}  
</style>  
</head>  
<body>  
  
<h1>Hello</h1>  
<p>Resize the height of the browser window to see how the font-size of h1  
changes.</p>  
<p>1vw = 1% of viewport height.</p>  
<p>The vw unit is not supported in IE8 and earlier.</p>  
  
</body>  
</html>
```

## vmin

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {  
  font-size: 15vmin;  
}  
</style>  
</head>  
<body>  
  
<h1>Hello</h1>  
<p>Resize the browser window (both width and height) to see how the font-size of  
h1 changes.</p>  
<p>1vmin = 1vw or 1vh, whichever is smaller.</p>  
  
</body>  
</html>
```

## vmax

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {  
  font-size: 15vmax;  
}  
</style>  
</head>  
<body>  
  
<h1>Hello</h1>
```

```






<p>Resize the browser window (both width and height) to see how the font-size of
h1 changes.</p>
<p>1vmax = 1vw or 1vh, whichever is larger.</p>
<p>The vmax unit is not supported in Edge 15 and earlier, nor in Safari 6.1 and
earlier.</p>

</body>
</html>

```

## 브라우저 지원

표의 숫자는 길이 단위를 완전히 지원하는 첫 번째 브라우저 버전을 지정합니다.

Length Unit					
em, ex, %, px, cm, mm, in, pt, pc	1.0	3.0	1.0	1.0	3.5
ch	27.0	9.0	1.0	7.0	20.0
rem	4.0	9.0	3.6	4.1	11.6
vh, vw	20.0	9.0	19.0	6.0	20.0
vmin	20.0	9.0*	19.0	6.0	20.0
vmax	26.0	X	19.0	7.0	20.0

참고 : \* Internet Explorer 9는 비표준 이름 인 vm으로 vmin을 지원합니다.

## CSS 특이성

### 특이성이란 무엇입니까?

동일한 요소를 가리키는 충돌하는 CSS 규칙이 두 개 이상 있는 경우 브라우저는 몇 가지 규칙에 따라 가장 구체적인 규칙을 결정하여 승리합니다.

특이성을 어떤 스타일 선언이 요소에 적용되는지를 결정하는 점수 / 순위로 생각하십시오.

범용 선택기 (\*)는 특이도가 낮고 ID 선택기는 매우 구체적입니다!

참고 : 특수성은 CSS 규칙이 일부 요소에는 적용되지 않는 일반적인 이유입니다.

# 특이성 계층

모든 선택기는 특이성 계층 구조에 위치합니다. 선택기의 특정 수준을 정의하는 네 가지 범주가 있습니다.

**인라인 스타일** -인라인 스타일은 스타일을 지정할 요소에 직접 첨부됩니다. 예 :  
<h1 style = "color : #ffffff;">.

**ID** -ID 는 #navbar 와 같은 페이지 요소의 고유 식별자입니다.

**클래스, 속성 및 유사 클래스** -이 범주에는 .classes, [속성] 및 유사 클래스 (예 : : hover, : focus 등)가 포함됩니다.

**요소 및 의사 요소** -이 범주에는 h1, div, : before 및 : after 와 같은 요소 이름 및 의사 요소가 포함됩니다.

## 특이성을 계산하는 방법?

특이성을 계산하는 방법을 기억하십시오!

0 에서 시작하여 스타일 속성에 1000 을 추가하고 각 ID 에 100 을 추가하고 각 속성, 클래스 또는 유사 클래스에 10 을 추가하고 각 요소 이름 또는 유사 요소에 1 을 추가하십시오.

다음 세 가지 코드 조각을 고려하십시오.

A: h1 B: #content h1 C: <div id="content"><h1 style="color: #ffffff">Heading</h1></div>
---

A 의 특이도는 1 (하나의 요소)  
B 의 특이도는 101 (하나의 ID 참조와 하나의 요소)  
C 의 특이도는 1000 (인라인 스타일링)

1 <101 <1000 이후, 세 번째 규칙 (C)은 더 높은 수준의 특이성을 가지므로 적용됩니다.

# 특이성 규칙

**동일성 :** 최신 규칙 수 -동일한 규칙이 외부 스타일 시트에 두 번 작성되면 스타일 시트의 하위 규칙이 스타일을 지정할 요소에 더 가깝게 적용되므로 다음이 적용됩니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  background-color: yellow;
}
h1 {
  background-color: red;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>

</body>
</html>
```

후자의 규칙은 항상 적용됩니다.

**ID 선택기는 속성 선택기보다 고유성이 높습니다.** 다음 세 가지 코드 행을 보십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
div#a {background-color: green;}
#a {background-color: yellow;}
div[id=a] {background-color: blue;}
</style>
</head>
<body>

<div id="a">This is a div</div>

</body>
</html>
```

첫 번째 규칙은 다른 두 규칙보다 더 구체적이며 적용됩니다.

**상황 별 선택기는 단일 요소 선택기보다 더 구체적입니다.** 포함 된 스타일 시트는 스타일을 지정할 요소에 더 가깝습니다. 따라서 다음 상황에서

```
From external CSS file:
#content h1 {background-color: red;}
```

```
In HTML file:
<style>
#content h1 {
  background-color: yellow;
}
</style>
```

후자의 규칙이 적용됩니다.

클래스 선택기는 .intro 비트 h1, p, div 등과 같은 클래스 선택기 보다 많은 요소 선택기를 이깁니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.intro {background-color: yellow;}
h1 {background-color: red;}
</style>
</head>
<body>

<h1 class="intro">This is a heading</div>

</body>
</html>
```

범용 선택기 및 상속 된 값의 고유도는 0- \*이고 body \*와 유사하며 고유성은 없습니다. 상속 된 값의 특이도는 0입니다.