

HTML

HTML 은 웹 페이지의 표준 마크 업 언어입니다.

HTML 을 사용하면 자신의 웹 사이트를 만들 수 있습니다.

HTML 은 배우기 쉽습니다-즐기십시오!

HTML이란 무엇입니까?

HTML 은 웹 페이지 작성을 위한 표준 마크 업 언어입니다.

- HTML 은 Hyper Text Markup Language 의 약자
- HTML 은 웹 페이지의 구조를 설명합니다
- HTML 은 일련의 요소로 구성
- HTML 요소는 브라우저에 내용을 표시하는 방법을 알려줍니다
- HTML 요소는 태그로 표시됩니다
- HTML 태그는 "제목", "단락", "표"등과 같은 콘텐츠를 레이블링합니다.
- 브라우저는 HTML 태그를 표시하지 않지만 이를 사용하여 페이지의 콘텐츠를 렌더링합니다.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

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

</body>
</html>
```

- `<!DOCTYPE html>` 선언은 HTML5 가 될 이 문서를 정의
- `<html>` 요소는 HTML 페이지의 루트 요소입니다
- `<head>` 요소는 문서에 대한 메타 정보를 포함
- `<title>` 요소는 문서의 제목을 명시
- `<body>` 요소는 눈에 보이는 페이지의 내용을 포함

- `<h1>` 요소는 큰 제목을 정의
- `<p>` 요소는 문단을 정의

HTML 태그

HTML 태그는 꺾쇠 괄호로 묶인 요소 이름입니다.

`<tagname> 콘텐츠가 여기에옵니다 ... </ tagname>`

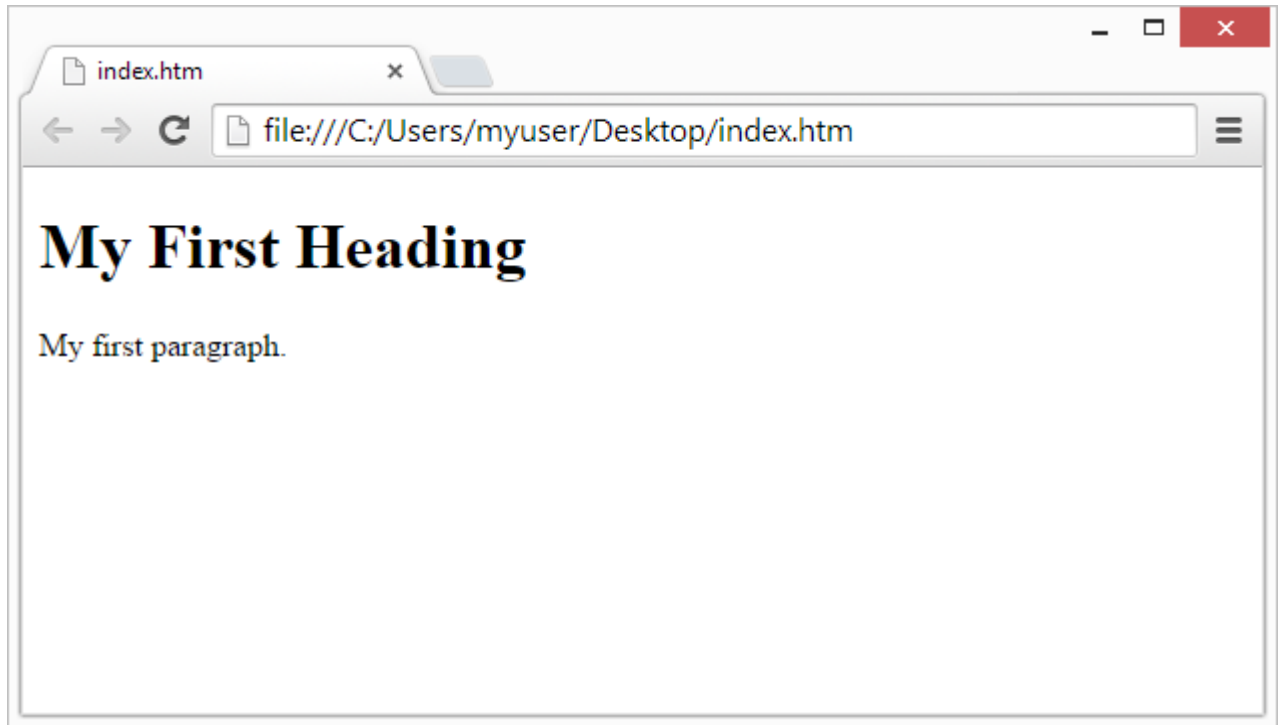
- HTML 태그는 일반적으로 올 쌍으로 같은 `<p>` 과 `</p>`
- 쌍의 첫 번째 태그는 시작 태그이고 두 번째 태그는 종료 태그입니다
- 종료 태그는 시작 태그처럼 작성되지만 태그 이름 앞에 슬래시가 삽입됩니다.

팁 : 시작 태그는 여는 태그 라고 하고 끝 태그는 닫는 태그라고 합니다.

웹 브라우저

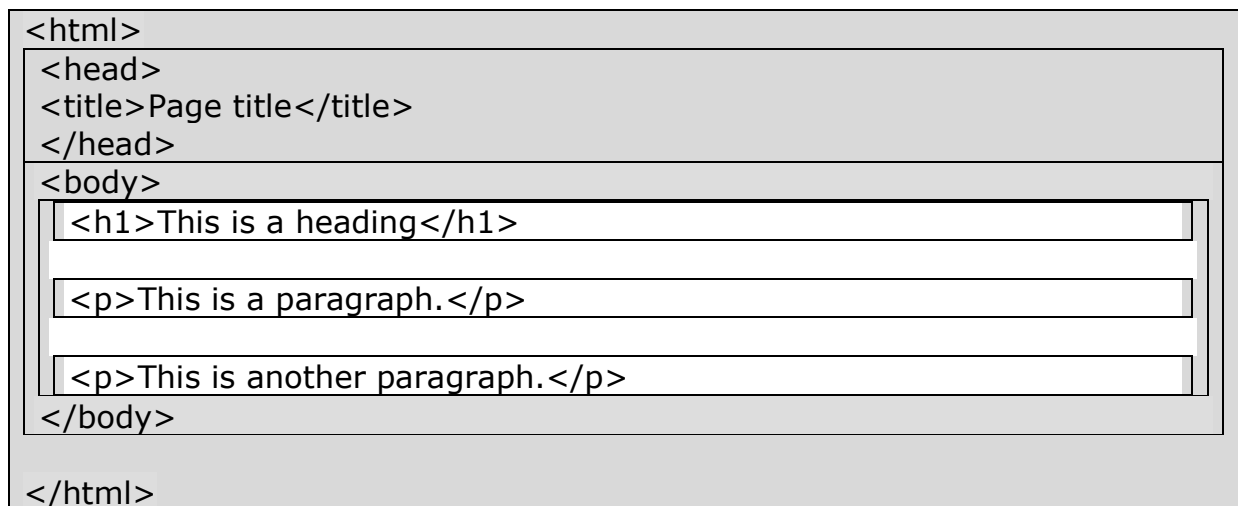
웹 브라우저 (Chrome, Edge, Firefox, Safari)의 목적은 HTML 문서를 읽고 표시하는 것입니다.

브라우저는 HTML 태그를 표시하지 않지만 이를 사용하여 문서를 표시하는 방법을 결정합니다.



HTML 페이지 구조

아래는 HTML 페이지 구조의 시각화입니다.



참고 : <body> 섹션 (위의 흰색 영역) 내의 내용 만 브라우저에 표시됩니다.

<!DOCTYPE> 선언

<!DOCTYPE> 선언은 문서 유형을 나타내며 올바르게 웹 페이지를 표시하는 브라우저를 하는 데 도움이 됩니다.

페이지 상단 (HTML 태그 이전)에 한 번만 나타나야 합니다.

<!DOCTYPE> 선언은 대소 문자를 구분하지 않습니다.

<!DOCTYPE> HTML5 의 선언은 다음과 같습니다.

```
<!DOCTYPE html>
```

HTML 버전

웹 초창기부터 많은 버전의 HTML 이있었습니다.

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

HTML 편집기

메모장 또는 TextEdit을 사용하여 HTML 작성

전문 HTML 편집기를 사용하여 웹 페이지를 작성하고 수정할 수 있습니다.

그러나 HTML 을 배우려면 메모장 (PC) 또는 텍스트 편집기 (Mac)와 같은 간단한 텍스트 편집기를 사용하는 것이 좋습니다.

간단한 텍스트 편집기를 사용하는 것이 HTML 을 배우는 좋은 방법이라고 생각합니다.

아래 단계에 따라 메모장 또는 TextEdit 으로 첫 번째 웹 페이지를 만드십시오.

1 단계 : 메모장 (PC) 열기

2 단계 : HTML 작성

3 단계 : HTML 페이지 저장

파일을 컴퓨터에 저장하십시오. 메모장 메뉴에서 **파일 > 다른 이름으로 저장**을 선택하십시오 .

파일 이름을 **"index.htm"** 으로 지정하고 인코딩을 **UTF-8** (HTML 파일의 기본 인코딩)로 설정하십시오.

.htm 또는 .html을 파일 확장자로 사용할 수 있습니다. 차이는 없습니다, 그것은 당신에게 달려 있습니다.

4 단계 : 브라우저에서 HTML 페이지보기

HTML 기본

HTML 문서

모든 HTML 문서는 `<!DOCTYPE html>` 문서 유형 선언으로 시작해야 합니다.

HTML 문서 자체는 `<html>`로 시작 하고 `</html>`로 끝납니다.

HTML 문서에서 보이는 부분은 `<body>`와 `</body>`사이에 있습니다.

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

HTML 제목

HTML 제목은 `<h1>` to `<h6>` 태그로 정의됩니다.

`<h1>` 가장 중요한 제목을 정의합니다. `<h6>` 가장 중요하지 않은 제목을 정의합니다.

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

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>

</body>
</html>
```

HTML 단락

HTML 단락은 `<p>` 태그로 정의됩니다.

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

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

HTML 링크

HTML 링크는 `<a>` 태그로 정의됩니다.

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

<h2>HTML Links</h2>
<p>HTML links are defined with the a tag:</p>
<a href="https://www.w3schools.com">This is a link</a>

</body>
</html>
```

링크 대상이 `href` 속성에 지정되어 있습니다.

속성은 HTML 요소에 대한 추가 정보를 제공하는 데 사용됩니다.

HTML 이미지

HTML 이미지는 `` 태그로 정의됩니다.

소스 파일 (`src`), 대체 텍스트 (`alt`), `width`, 및 `height` 속성으로 제공됩니다.

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

<h2>HTML Images</h2>
<p>HTML images are defined with the img tag:</p>


</body>
</html>
```

HTML 버튼

HTML 버튼은 다음 `<button>` 태그로 정의됩니다.

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

<h2>HTML Buttons</h2>
<p>HTML buttons are defined with the button tag:</p>
<button>Click me</button>

</body>
</html>
```

HTML 목록

HTML 목록은 ``(정렬되지 않은 / 글 머리 기호 목록) 또는 ``(순서 / 번호가 매겨진 목록) 태그와 `` 태그 (목록 항목)로 정의됩니다.

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

<h2>An Unordered HTML List</h2>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

<h2>An Ordered HTML List</h2>

<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```


HTML 요소

HTML 요소

HTML 요소는 일반적으로 시작 태그와 종료 태그로 구성되며 내용은 다음 사이에 삽입됩니다.

`< tagname >` 내용은 여기에 있습니다 ... `< / tagname >`

HTML 요소 는 시작 태그에서 종료 태그까지의 모든 것입니다.

`< P >` 내 첫 번째 단락. `< / p >`

Start tag	Element content	End tag
<code><h1></code>	My First Heading	<code></h1></code>
<code><p></code>	My first paragraph.	<code></p></code>
<code>
</code>		

내용이 없는 HTML 요소를 빈 요소라고 합니다. 빈 요소에는 `
` 요소 (줄 바꿈을 나타냄)와 같은 종료 태그가 없습니다.

중첩 된 HTML 요소

HTML 요소는 중첩 될 수 있습니다 (요소에는 요소가 포함될 수 있음).

모든 HTML 문서는 중첩 된 HTML 요소로 구성됩니다.

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

<html>요소는 전체 문서를 정의한다. 그것은 시작 태그 **<HTML>** 및 종료 태그 **</HTML>** 가지고 있다.

<html> 요소 안에는 **<body>**요소가 있습니다.

<body>요소는 문서 본문을 정의한다. 그것은 시작 태그 **<body>**와 종료 태그 **</body>** 가지고 있다.

다음 **<BODY>** 내부 요소는 **<h1>**및 **<p>** 두 개의 다른 HTML 요소입니다.

<h1>요소는 제목을 정의한다. 그것은 시작 태그 **<H1>** 및 종료 태그 **</H1>** 가지고 있다. 요소 내용은 My First Heading. 이다.

<p>요소는 단락을 정의한다. 그것은 시작 태그 **<p>** 및 종료 태그 **</p>** 가지고 있다. 요소 내용은 My first paragraph. 이다.

종료 태그를 잊지 마십시오

종료 태그를 잊어 버린 경우에도 일부 HTML 요소가 올바르게 표시됩니다.

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

<p>This is a paragraph
<p>This is a paragraph

</body>
</html>
```

닫는 태그는 선택 사항으로 간주되므로 위의 예는 모든 브라우저에서 작동합니다.

이것에 의존하지 마십시오. 종료 태그를 잊어 버린 경우 예기치 않은 결과 및 / 또는 오류가 발생할 수 있습니다.

빈 HTML 요소

내용이 없는 HTML 요소를 빈 요소라고 합니다.

**
닫는 태그가 없는 빈 요소입니다 (
** 태그는 줄 바꿈을 정의 함).

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

<p>This is a <br> paragraph with a line break.</p>

</body>
</html>
```


와 같이 빈 태그는 여는 태그에서 "닫힐"수 있습니다.

HTML5 는 빈 요소를 닫을 필요가 없습니다. 그러나 보다 엄격한 유효성 검사를 원하거나 XML 파서가 문서를 읽을 수 있게 하려면 모든 HTML 요소를 올바르게 닫아야 합니다.

대소 문자를 구분하지 않는 HTML

HTML 태그는 대소 문자를 구분하지 않습니다. <P>는 <p>와 동일합니다.

HTML5 표준에는 소문자 태그가 필요하지 않지만 W3C 는 소문자 HTML 을 권장하며 XHTML 과 같은 보다 엄격한 문서 유형에는 소문자를 요구 합니다.

HTML 속성

속성은 HTML 요소에 대한 추가 정보를 제공합니다.

HTML 속성

- 모든 HTML 요소는 속성 을 가질 수 있습니다
- 속성 은 요소에 대한 추가 정보 를 제공합니다
- 속성은 항상 시작 태그에 지정됩니다
- 속성은 일반적으로 다음과 같은 이름 / 값 쌍으로 제공됩니다. **name = "value"**

href 속성

HTML 링크는 `<a>` 태그로 정의 됩니다. 링크 주소는 `href` 속성에 지정되어 있습니다.

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

<h2>The href Attribute</h2>
<p>HTML links are defined with the a tag. The link address is specified in the attribute:</p>

<a href="https://www.w3schools.com">This is a link</a>

</body>
</html>
```

src 속성

HTML 이미지는 `` 태그 로 정의됩니다.

이미지 소스의 파일 이름은 `src` 속성에 지정되어 있습니다.

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

<h2>The src Attribute</h2>
<p>HTML images are defined with the img tag. and the filename of the image source is specified in the src attribute:</p>



</body>
</html>
```

너비와 높이 속성

HTML 의 이미지에는 이미지의 너비와 높이를 지정하는 `width` 와 `height` 속성 가지고 있다.

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

<h2>Size Attributes</h2>
<p>Images in HTML have a set of size attributes, which specifies the width and height of the image:</p>



</body>
</html>
```

너비와 높이는 기본적으로 픽셀 단위로 지정됩니다. 따라서 width="500"은 너비가 500 픽셀임을 의미합니다.

대체 속성

이 **alt** 속성은 이미지를 표시 할 수 없는 경우 사용할 대체 텍스트를 지정합니다.

alt 화면 판독기 는 속성 값을 읽을 수 있습니다. 이러한 방식으로, 시각 장애가 있는 사람과 같이 웹 페이지를 "듣는"누군가가 그 요소를 "듣게"할 수 있습니다.

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

<h2>The alt Attribute</h2>
<p>The alt attribute should reflect the image content, so users who cannot see the image gets an understanding of what the image contains:</p>



</body>
</html>
```

이 **alt**속성은 이미지를 표시 할 수 없는 경우 (예 : 존재하지 않는 경우)에도 유용합니다.

스타일 속성

이 **style** 속성은 색상, 글꼴, 크기 등과 같은 요소의 스타일을 지정하는 데 사용됩니다.

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

<h2>The style Attribute</h2>
<p>The style attribute is used to specify the styling of an element, like color:
</p>

<p style="color:red">This is a paragraph.</p>

</body>
</html>
```

lang 속성

문서의 언어는 **<html>** 태그 에서 선언 될 수 있습니다.

언어는 **lang** 속성으로 선언됩니다.

접근성 응용 프로그램 (스크린 리더) 및 검색 엔진에 언어를 선언하는 것이 중요합니다.

```
<!DOCTYPE html>
<html lang="en-US">
<body>

...

</body>
</html>
```

처음 두 글자는 언어 (en)를 지정합니다. 방언이 있으면 두 글자를 더 추가하십시오 (영문).

제목 속성

여기서 **title** 속성이 **<p>** 요소에 추가됩니다. 단락 위로 마우스를 가져 가면 **title** 속성 값이 툴팁으로 표시됩니다.

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

<h2 title="I'm a header">The title Attribute</h2>

<p title="I'm a tooltip">
Mouse over this paragraph, to display the title attribute as a tooltip.
</p>

</body>
</html>
```

제안 : 소문자 속성 사용

HTML5 표준에는 소문자 속성 이름이 필요하지 않습니다.

title 속성은 **title** 또는 **TITLE** 과 같이 대문자 또는 소문자로 쓸 수 있습니다.

W3C 는 HTML 에서 소문자를 권장 하고 XHTML 과 같은 보다 엄격한 문서 유형에는 소문자를 요구 합니다.

제안 : 속성 값 인용

HTML5 표준에서는 속성 값 주위에 따옴표가 필요하지 않습니다.

```
<a href=https://www.w3schools.com>
```

W3C 는 HTML 로 따옴표를 권장 하고 XHTML 과 같은 보다 엄격한 문서 유형에 대해서는 따옴표를 요구 합니다.

```
<a href="https://www.w3schools.com">
```

때때로는 필요에 따옴표를 사용하십시오. 이 예는 제목 속성에 공백이 포함되어 있으므로 올바르게 표시되지 않습니다.

```
<p title>About W3Schools>
```

따옴표를 사용하는 것이 가장 일반적입니다. 따옴표를 생략하면 오류가 발생할 수 있습니다.

작은 따옴표 또는 큰 따옴표?

속성 값 주위에 큰 따옴표가 HTML에서 가장 일반적이지만 작은 따옴표도 사용할 수 있습니다.

경우에 따라 속성 값 자체에 큰 따옴표가 포함 된 경우 작은 따옴표를 사용해야 합니다.

```
<p title='John "ShotGun" Nelson'>
```

혹은 그 반대로도:

```
<p title="John 'ShotGun' Nelson">
```

단원 요약

- 모든 HTML 요소는 속성 을 가질 수 있습니다
- 이 **title** 속성은 추가 "툴팁"정보를 제공합니다
- 이 **href** 속성은 링크의 주소 정보를 제공합니다
- **width** 및 **height** 속성은 이미지의 크기 정보를 제공
- 이 **alt** 속성은 스크린 리더를 위한 텍스트를 제공합니다

HTML 속성

Attribute	Description
alt	Specifies an alternative text for an image, when the image cannot be displayed

disabled	Specifies that an input element should be disabled
href	Specifies the URL (web address) for a link
id	Specifies a unique id for an element
src	Specifies the URL (web address) for an image
style	Specifies an inline CSS style for an element
title	Specifies extra information about an element (displayed as a tool tip)

HTML 제목

HTML 제목

제목은 `<h1>` to `<h6>` 태그로 정의됩니다.

`<h1>` 가장 중요한 제목을 정의합니다. `<h6>` 가장 덜 중요한 제목을 정의합니다.

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

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>

</body>
</html>
```

참고 : 브라우저는 제목 앞뒤에 공백 (여백)을 자동으로 추가합니다.

제목이 중요하다

검색 엔진은 표제를 사용하여 웹 페이지의 구조 및 콘텐츠를 색인화합니다.

사용자는 종종 제목으로 페이지를 훑어 봅니다. 문서 구조를 표시하려면 제목을 사용하는 것이 중요합니다.

`<h1>`제목은 주 제목에 사용되어야 하고 그 뒤에 `<h2>`제목이 사용되며 그 다음에는 덜 중요 `<h3>`합니다.

참고 : 제목에만 HTML 제목을 사용하십시오. **BIG** 또는 **bold** 텍스트를 만들기 위해 제목을 사용하지 마십시오.

더 큰 제목

각 HTML 제목에는 기본 크기가 있습니다. 그러나 `style` CSS `font-size` 속성을 사용하여 속성이 있는 제목의 크기를 지정할 수 있습니다.

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

<h1 style="font-size:60px;">Heading 1</h1>

<p>You can change the size of a heading with the style attribute, using the font-size property.</p>

</body>
</html>
```

HTML 수평 규칙

`<hr>`태그는 HTML 페이지에서 주제 휴식을 정의하고, 가장 자주 수평선으로 표시됩니다.

이 `<hr>`요소는 HTML 페이지에서 콘텐츠를 분리하거나 변경을 정의하는 데 사용됩니다.

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

<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
```

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

HTML <head> 요소

HTML <head> 요소는 메타 데이터를 위한 컨테이너입니다. HTML 메타 데이터는 HTML 문서에 대한 데이터입니다. 메타 데이터는 표시되지 않습니다.

<head> 요소는 <html> 태그와 <body> 태그 사이에 배치된다.

```
<!DOCTYPE html>
<html>

<head>
  <title>My First HTML</title>
  <meta charset="UTF-8">
</head>

<body>
  .
  .
  .
```

참고 : 메타 데이터는 일반적으로 문서 제목, 문자 세트, 스타일, 스크립트 및 기타 메타 정보를 정의합니다.

HTML 태그 참조

Tag	Description
<html>	Defines the root of an HTML document
<body>	Defines the document's body
<head>	A container for all the head elements (title, scripts, styles, meta information, and more)
<h1> to <h6>	Defines HTML headings
<hr>	Defines a thematic change in the content

HTML 단락

HTML 단락

HTML `<p>` 요소는 단락을 정의합니다.

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

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

참고 : 브라우저는 단락 앞뒤에 공백 (여백)을 자동으로 추가합니다.

HTML 디스플레이

HTML 이 어떻게 표시 될지 확신 할 수 없습니다.

크거나 작은 화면과 크기 조정 된 창은 다른 결과를 생성합니다.

HTML 을 사용하면 HTML 코드에 공백이나 줄을 추가하여 출력을 변경할 수 없습니다.

페이지가 표시되면 브라우저는 여분의 공백과 여분의 행을 제거합니다.

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

<p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
</p>

<p>
This paragraph
```

```
contains      a lot of spaces
in the source      code,
but the      browser
ignores it.
</p>

<p>The number of lines in a paragraph depends on the size of the browser window.
If you resize the browser window, the number of lines in this paragraph will
change.</p>

</body>
</html>
```

종료 태그를 잊지 마십시오

종료 태그를 잊어 버린 경우에도 대부분의 브라우저는 HTML 을 올바르게 표시합니다.

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

<p>This is a paragraph.
<p>This is another paragraph.
<p>This is another paragraph.

<p>Don't forget to close your HTML tags!</p>

</body>
</html>
```

위의 예는 대부분의 브라우저에서 작동하지만 의존하지는 않습니다.

참고 : 종료 태그를 삭제하면 예기치 않은 결과 또는 오류가 발생할 수 있습니다.

HTML 줄 바꿈

HTML `
` 요소는 줄 바꿈을 정의합니다.

`
` 새 단락을 시작하지 않고 줄 바꿈 (새 줄)을 원할 경우 사용하십시오.

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

```
<p>This is<br>a paragraph<br>with line breaks.</p>
</body>
</html>
```

**
**태그는 끝 태그가 없음을 의미 빈 태그입니다.

시 문제

이시는 한 줄로 표시됩니다.

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

<p>In HTML, spaces and new lines are ignored:</p>
<p>

    My Bonnie lies over the ocean.

    My Bonnie lies over the sea.

    My Bonnie lies over the ocean.

    Oh, bring back my Bonnie to me.

</p>

</body>
</html>
```

HTML <pre> 요소

HTML **<pre>**요소는 사전 형식화 된 텍스트를 정의합니다.

<pre>요소 내부의 텍스트는 고정 너비 글꼴로 표시되며 공백과 줄 바꿈을 모두 유지합니다.

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

<p>The pre tag preserves both spaces and line breaks:</p>
<pre>
```

```
My Bonnie lies over the ocean.  
My Bonnie lies over the sea.  
My Bonnie lies over the ocean.  
Oh, bring back my Bonnie to me.  
  
</pre>  
  
</body>  
</html>
```

HTML 태그 참조

Tag	Description
<p>	Defines a paragraph

	Inserts a single line break
<pre>	Defines pre-formatted text

HTML 스타일

```
나는 레드  
  
나는 블루  
  
나는 크다
```

HTML 스타일 속성

HTML 요소의 스타일 설정은 **style** 속성 으로 수행 할 수 있습니다.

HTML **style** 속성의 구문 은 다음과 같습니다.

```
<tagname style="property:value;">
```

이 속성 은 CSS 속성입니다. 값은 CSS의 값입니다.

배경색

CSS `background-color` 속성은 HTML 요소의 배경색을 정의합니다.

이 예제는 페이지의 배경색을 `powderblue` 로 설정합니다.

```
<!DOCTYPE html>
<html>
<body style="background-color:powderblue;">

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

</body>
</html>
```

글자 색

CSS `color` 속성은 HTML 요소의 텍스트 색상을 정의합니다.

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

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

</body>
</html>
```

폰트

CSS `font-family` 속성은 HTML 요소에 사용될 글꼴을 정의합니다.

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

<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```



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

문자 크기

CSS `font-size` 속성은 HTML 요소의 텍스트 크기를 정의합니다.

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

<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>

</body>
</html>
```

텍스트 정렬

CSS `text-align` 속성은 HTML 요소의 가로 텍스트 정렬을 정의합니다.

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

<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>

</body>
</html>
```

단원 요약

- HTML 요소 스타일 지정에 `style` 속성 사용
- 배경 색상은 `background-color` 속성 사용
- 텍스트 색상은 `color` 속성 사용
- 텍스트 글꼴은 `font-family` 속성 사용
- 텍스트 크기는 `font-size` 속성 사용
- 텍스트 정렬은 `text-align` 속성 사용

HTML 텍스트 형식

This text is bold

This text is italic

This is _{subscript} and ^{superscript}

HTML 서식 요소

HTML은 또한 특별한 의미로 텍스트를 정의하기 위한 특수 요소를 정의합니다.

굵게 또는 기울임꼴 텍스트와 같은 출력 서식은 ``과 `<i>`같은 HTML 요소를 사용합니다.

서식 요소는 특수한 유형의 텍스트를 표시하도록 설계되었습니다.

- `` - 굵은 텍스트
- `` - 중요한 텍스트
- `<i>` - 이탤릭체
- `` - 강조된 텍스트
- `<mark>` - 마크된 텍스트
- `<small>` - 작은 텍스트
- `` - 삭제된 텍스트
- `<ins>` - 삽입된 텍스트
- `<sub>` - 아래첨자 텍스트
- `<sup>` - 위첨자 텍스트

HTML `` 및 `` 요소

HTML `` 요소는 특별한 중요성 없이 굵은 체 텍스트를 정의합니다.

```
<!DOCTYPE html>
```

```
<html>
<body>

<p>This text is normal.</p>
<p><b>This text is bold</b></p>

</body>
</html>
```

HTML **** 요소는 의미론적 "강한" 중요성이 추가 **강력한** 텍스트를 정의합니다.

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

<p>This text is normal.</p>
<p><strong>This text is strong</strong></p>

</body>
</html>
```

HTML <i> 및 요소

HTML **<i>** 요소는 특별한 중요성 없이 *이탤릭체* 텍스트를 정의합니다.

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

<p>This text is normal.</p>
<p><i>This text is italic</i></p>

</body>
</html>
```

HTML **** 요소는 의미적 중요성이 추가 된 *강조된* 텍스트를 정의합니다.

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

<p>This text is normal.</p>
<p><em>This text is emphasized</em></p>

</body>
</html>
```

참고 : ``과 ``, ``과 `<i>` 브라우저 표시는 같으나, 이들 태그의 의미 차이는 ``와 `<i>`는 굵은 이탤릭체 텍스트를 정의하지만, ``및 ``텍스트가 "중요"을 의미합니다.

HTML `<small>` 요소

HTML `<small>`요소는 더 작은 텍스트를 정의합니다.

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

<h2>HTML <small>Small</small> Formatting</h2>

</body>
</html>
```

HTML `<mark>` 요소

HTML `<mark>`요소는 표시 / 강조 표시된 텍스트를 정의합니다.

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

<h2>HTML <mark>Marked</mark> Formatting</h2>

</body>
</html>
```

HTML `` 요소

HTML ``요소는 삭제 / 제거 된 텍스트를 정의합니다.

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

<p>The del element represents deleted (removed) text.</p>
```

```
<p>My favorite color is <del>blue</del> red.</p>
</body>
</html>
```

HTML <ins> 요소

HTML <ins> 요소는 삽입 / 추가 된 텍스트를 정의합니다.

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

<p>The ins element represent inserted (added) text.</p>
<p>My favorite <ins>color</ins> is red.</p>

</body>
</html>
```

HTML <sub> 요소

HTML <sub> 요소는 첨자 텍스트를 정의합니다.

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

<p>This is <sub>subscripted</sub> text.</p>

</body>
</html>
```

HTML <sup> 요소

HTML <sup> 요소는 위첨자 텍스트를 정의합니다.

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

<p>This is <sup>superscripted</sup> text.</p>

</body>
```

```
</html>
```

HTML 텍스트 서식 요소

Tag	Description
<code></code>	Defines bold text
<code></code>	Defines emphasized text
<code><i></code>	Defines italic text
<code><small></code>	Defines smaller text
<code></code>	Defines important text
<code><sub></code>	Defines subscripted text
<code><sup></code>	Defines superscripted text
<code><ins></code>	Defines inserted text
<code></code>	Defines deleted text
<code><mark></code>	Defines marked/highlighted text

HTML 인용 및 인용 요소

Here is a quote from WWF's website:

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

짧은 인용문 용 HTML `<q>`

HTML `<q>` 요소는 짧은 인용을 정의합니다.

브라우저는 일반적으로 `<q>` 요소 주위에 따옴표를 삽입합니다.

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

<p>Browsers usually insert quotation marks around the q element.</p>

<p>WWF's goal is to: <q>Build a future where people live in harmony with
nature.</q></p>
```

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

인용문 용 HTML <blockquote>

HTML <blockquote>요소는 다른 소스에서 인용 된 섹션을 정의합니다.

브라우저는 일반적으로 <blockquote>요소를 들여 씁니다.

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

<p>Browsers usually indent blockquote elements.</p>

<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature.
The world's leading conservation organization,
WWF works in 100 countries and is supported by
1.2 million members in the United States and
close to 5 million globally.
</blockquote>

</body>
</html>
```

약어의 HTML <abbr>

HTML <abbr>요소는 약어 또는 머리 글자를 정의합니다.

마킹 약어는 브라우저, 번역 시스템 및 검색 엔진에 유용한 정보를 제공 할 수 있습니다.

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

<p>The <abbr title="World Health Organization">WHO</abbr> was founded in
1948.</p>

<p>Marking up abbreviations can give useful information to browsers, translation
systems and search-engines.</p>

</body>
</html>
```

연락처 정보 용 HTML <address>

HTML <address> 요소는 문서 또는 기사의 연락처 정보 (저자 / 소유자)를 정의합니다.

<address> 요소는 일반적으로 이탤릭체로 표시됩니다. 대부분의 브라우저는 요소 앞뒤에 줄 바꿈을 추가합니다.

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

<p>The HTML address element defines contact information (author/owner) of a
document or article.</p>

<address>
Written by John Doe.<br>
Visit us at:<br>
Example.com<br>
Box 564, Disneyland<br>
USA
</address>

</body>
</html>
```

작품 제목 HTML <cite>

HTML <cite> 요소는 작품의 제목을 정의합니다.

브라우저는 일반적으로 <cite> 이탤릭체로 요소를 표시합니다.

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

<p>The HTML cite element defines the title of a work.</p>
<p>Browsers usually display cite elements in italic.</p>


<p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>

</body>
</html>
```


양방향 재정의의를 위한 HTML <bdo>

HTML <bdo> 요소는 양방향 재정의의를 정의합니다.

<bdo> 요소는 현재의 텍스트 방향을 무시하는 데 사용됩니다.

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

<p>If your browser supports bi-directional override (bdo), the next line will be
written from right to left (rtl):</p>

<bdo dir="rtl">This text will be written from right to left</bdo>

</body>
</html>
```

HTML 인용 및 인용 요소

Tag	Description
<abbr>	Defines an abbreviation or acronym
<address>	Defines contact information for the author/owner of a document
<bdo>	Defines the text direction
<blockquote>	Defines a section that is quoted from another source
<cite>	Defines the title of a work
<q>	Defines a short inline quotation

HTML 주석

주석 태그는 HTML 소스 코드에 주석을 삽입하는 데 사용됩니다.

HTML 주석 태그

다음 구문을 사용하여 HTML 소스에 주석을 추가 할 수 있습니다.

```
<!-- Write your comments here -->
```

여는 태그에는 느낌표 (!)가 있지만 닫기 태그에는 없습니다.

참고 : 주석은 브라우저에 표시되지 않지만 HTML 소스 코드를 문서화하는 데 도움이 될 수 있습니다.

주석을 사용하면 HTML 에 알림 및 미리 알림을 배치 할 수 있습니다.

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

<!-- This is a comment -->

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

<!-- Remember to add more information here -->

</body>
</html>
```

주석은 HTML 디버깅에 유용합니다. 한 번에 하나씩 HTML 코드 행을 주석 처리하여 오류를 검색 할 수 있기 때문입니다.

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

<!-- Do not display this image at the moment

-->

</body>
</html>
```

HTML 색상

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

색 이름

HTML 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](#).

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>
```

글자 색

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>
```

테두리 색

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>
```

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>
```

RGB 값

HTML에서 다음 공식을 사용하여 색상을 RGB 값으로 지정할 수 있습니다.

rgb(red, green, blue)

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

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

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

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

16 진수 가치

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 가치

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 백분을 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 설명할 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 가치

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>
```

HTML 스타일-CSS

CSS로 HTML 스타일링

CSS 는 의미 **C**ascading **S**tyle **S**heets.

CSS 는 **HTML** 요소가 화면, 종이 또는 기타 매체에 표시되는 방법을 설명 합니다.

CSS 는 많은 작업을 저장합니다. 여러 웹 페이지의 레이아웃을 한 번에 제어 할 수 있습니다.

CSS 는 3 가지 방법으로 HTML 요소에 추가 될 수 있습니다.

- **인라인** -HTML 요소에서 스타일 속성 사용
- **내부** - 섹션 의 `<style>`요소를 사용하여 `<head>`
- **외부** -외부 CSS 파일을 사용하여

CSS 를 추가하는 가장 일반적인 방법은 스타일을 별도의 CSS 파일로 유지하는 것입니다. 그러나 여기에서는 인라인 및 내부 스타일링을 사용합니다. 시연하기 쉽고 직접 사용해 볼 수 있기 때문입니다.

인라인 CSS

인라인 CSS 는 단일 HTML 요소에 고유한 스타일을 적용하는 데 사용됩니다.

인라인 CSS 는 HTML 요소의 스타일 속성을 사용합니다.

이 예제는 `<h1>` 요소의 텍스트 색상을 파란색으로 설정합니다.

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

<h1 style="color:blue;">This is a Blue Heading</h1>

</body>
</html>
```

내부 CSS

내부 CSS 는 단일 HTML 페이지의 스타일을 정의하는 데 사용됩니다.

내부 CSS 는 HTML 페이지 `<head>` 섹션의 `<style>` 요소 내에 정의되어 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

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

</body>
</html>
```

외부 CSS

외부 스타일 시트는 많은 HTML 페이지의 스타일을 정의하는 데 사용됩니다.

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

외부 스타일 시트를 사용하려면 HTML 페이지 `<head>` 섹션에 해당 스타일 시트에 대한 링크를 추가 하십시오.

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

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

</body>
</html>
```

외부 스타일 시트는 모든 텍스트 편집기에서 쓸 수 있습니다. 파일에는 HTML 코드가 포함되어서는 안되며 확장자는 `.css` 로 저장해야 합니다.

"styles.css"의 모양은 다음과 같습니다.

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

CSS 글꼴

CSS `color` 속성은 사용할 텍스트 색상을 정의합니다.

CSS `font-family` 속성은 사용할 글꼴을 정의합니다.

CSS **font-size** 속성은 사용할 텍스트 크기를 정의합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
</head>
<body>

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

</body>
</html>
```

CSS 테두리

CSS **border** 속성은 HTML 요소 주위에 테두리를 정의합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border: 1px solid powderblue;
}
</style>
</head>
<body>

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

</body>
</html>
```

CSS 패딩

CSS **padding** 속성은 텍스트와 테두리 사이의 패딩 (공백)을 정의합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border: 1px solid powderblue;
  padding: 30px;
}
</style>
</head>
<body>

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

</body>
</html>
```

CSS 여백

CSS **margin** 속성은 테두리 외부에 여백 (공백)을 정의합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border: 1px solid powderblue;
  margin: 50px;
}
</style>
</head>
<body>

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

</body>
</html>
```

id 속성

하나의 특수 요소에 특정 스타일을 정의하려면 요소에 **id** 속성을 추가하십시오.

```
<p id="p01">I am different</p>
```

그런 다음 특정 ID로 요소의 스타일을 정의하십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
#p01 {
  color: blue;
}
</style>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p id="p01">I am different.</p>

</body>
</html>
```

참고 : 요소의 id는 페이지 내에서 고유해야 하므로 id 선택기는 하나의 고유한 요소를 선택하는 데 사용됩니다!

클래스 속성

특수한 유형의 요소에 대한 스타일을 정의하려면 요소에 **class** 속성을 추가하십시오.

```
<p class="error">I am different</p>
```

그런 다음 특정 클래스가 있는 요소의 스타일을 정의하십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.error {
  color: red;
}
```

```

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

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p class="error">I am different.</p>
<p>This is a paragraph.</p>
<p class="error">I am different too.</p>

</body>
</html>

```

외부 참조

외부 스타일 시트는 전체 URL 또는 현재 웹 페이지에 상대적인 경로를 사용하여 참조 할 수 있습니다.

이 예제는 전체 URL 을 사용하여 스타일 시트에 링크합니다.

```

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://www.w3schools.com/html/styles.css">
</head>
<body>

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

</body>
</html>

```

이 예제는 현재 웹 사이트의 html 폴더에 있는 스타일 시트에 링크됩니다.

```

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

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

</body>
</html>

```


이 예제는 현재 페이지와 동일한 폴더에있는 스타일 시트에 링크됩니다.

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

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

</body>
</html>
```

단원 요약

- 인라인 스타일링에 HTML `style` 속성 사용
- HTML `<style>` 요소를 사용하여 내부 CSS 정의
- HTML `<link>` 요소를 사용하여 외부 CSS 파일을 참조
- HTML `<head>` 요소를 사용하여 `<style>` 및 `<link>` 요소를 저장
- 텍스트 색상에 CSS `color` 속성 사용
- 텍스트 글꼴에 CSS `font-family` 속성 사용
- 텍스트 크기에 CSS `font-size` 속성 사용
- 테두리에 CSS `border` 속성 사용
- 테두리 내부의 공간에 CSS `padding` 속성을 사용
- 테두리 밖의 공간에 CSS `margin` 속성을 사용

HTML 스타일 태그

Tag	Description
<style>	Defines style information for an HTML document
<link>	Defines a link between a document and an external resource

HTML 링크

링크는 거의 모든 웹 페이지에 있습니다. 링크를 사용하면 사용자가 페이지 간을 클릭 할 수 있습니다.

HTML 링크-하이퍼 링크

HTML 링크는 하이퍼 링크입니다.

링크를 클릭하고 다른 문서로 이동할 수 있습니다.

링크 위로 마우스를 움직이면 마우스 화살표가 작은 손 모양으로 바뀝니다.

참고 : 링크는 텍스트 일 필요는 없습니다. 이미지 또는 다른 HTML 요소 일 수 있습니다.

HTML 링크-구문

하이퍼 링크는 HTML `<a>` 태그로 정의됩니다.

```
<a href="url">Link text</a>
```

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

<h2>HTML Links</h2>
<p><a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a></p>

</body>
</html>
```

이 `href` 속성은 링크의 대상 주소 (`https://www.w3schools.com/html/`)를 지정합니다.

링크 텍스트가 보이는 부분 (Visit our HTML tutorial)입니다.

링크 텍스트를 클릭하면 지정된 주소로 보내집니다.

참고 : 하위 폴더 주소 끝에 슬래시가 없으면 서버에 두 개의 요청을 생성 할 수 있습니다. 많은 서버가 주소 끝에 슬래시를 자동으로 추가 한 다음 새 요청을 만듭니다.

로컬 링크

위의 예에서는 절대 URL (전체 웹 주소)을 사용했습니다.

로컬 링크 (같은 웹 사이트에 대한 링크)는 상대 URL (https : // www)없이 지정됩니다.

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

<h2>HTML Links</h2>
<p><a href="html_images.asp">HTML Images</a>is a link to a page on this website.
</p>
<p><a href="https://www.w3.org/">W3C</a>is a link to a website on the World Wide
Web.</p>

</body>
</html>
```

HTML 링크 색상

기본적으로 링크는 다음과 같이 나타납니다 (모든 브라우저에서).

- 방문하지 않은 링크는 밑줄과 파란색으로 표시됩니다
- 방문한 링크는 밑줄이 그어져 있으며 자주색
- 활성화 된 링크는 밑줄과 빨간색으로 표시됩니다

CSS 를 사용하여 기본 색상을 변경할 수 있습니다.

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

```

<head>
<style>
a:link {
    color: green;
    background-color: transparent;
    text-decoration: none;
}

a:visited {
    color: pink;
    background-color: transparent;
    text-decoration: none;
}

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

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

<h2>Link Colors</h2>
<p>You can change the default colors of links</p>
<a href="html_images.asp" target="_blank">HTML Images</a>
</body>
</html>

```

링크는 종종 CSS를 사용하여 버튼으로 스타일이 지정됩니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
a:link, a:visited {
    background-color: #f44336;
    color: white;
    padding: 15px 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="default.asp" target="_blank">This is a link</a>
</body>
</html>
```

HTML 링크-대상 속성

이 **target** 속성은 링크된 문서를 열 위치를 지정합니다.

target 속성은 다음 값 중 하나를 사용할 수 있습니다:

- **_blank** - 링크 된 문서를 새 창이나 탭에서 엽니다
- **_self** - 클릭 한 것과 동일한 창 / 탭에서 링크 된 문서를 엽니다 (기본값).
- **_parent** - 부모 프레임에서 링크 된 문서를 엽니다
- **_top** - 링크 된 문서를 창 전체에서 엽니다
- **filename** - 이름이 지정된 프레임으로 링크 된 문서를 엽니다

이 예제는 링크 된 문서를 새 브라우저 창 / 탭에서 엽니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 15px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

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

<h2>The target Attribute</h2>
<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a>
<p>If you set the target attribute to "_blank", the link will open in a new
browser window or tab.</p>
</body>
</html>
```

팁 : 웹 페이지가 프레임에 잠겨 있으면 프레임 `target="_top"`을 벗어나는 데 사용할 수 있습니다 .

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>The target Attribute</h2>
<p>Locked in a frame?
<a href="https://www.w3schools.com/html/" target="_top">Click here!</a></p>
</body>
</html>
```

HTML 링크-링크 이미지

이미지를 링크로 사용하는 것이 일반적입니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Image Links</h2>
<p>The image is a link. You can click on it.</p>
<a href="default.asp">
  
</a>
<p>We have added "border:0" to prevent IE9 (and earlier) from displaying a border
around the image.</p>
</body>
</html>
```

참고 : `border:0;` IE9 (및 이전 버전)가 이미지 주위에 테두리를 표시하지 않도록 추가되었습니다 (이미지가 링크 인 경우).

링크 제목

이 **title** 속성은 요소에 대한 추가 정보를 지정합니다. 정보는 마우스가 요소 위로 움직일 때 가장 자주 툴팁 텍스트로 표시됩니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Link Titles</h2>
<p>The title attribute specifies extra information about an element. The
information is most often shown as a tooltip text when the mouse moves over the
element. </p>
<a href="https://www.w3schools.com/html/" title="Go to W3Schools HTML
section">Visit our HTML Tutorial</a>
</body>
</html>
```

HTML 링크-북마크 생성

HTML 책갈피는 독자가 웹 페이지의 특정 부분으로 이동할 수 있도록 하는데 사용됩니다.

웹 페이지가 너무 길면 북마크가 유용 할 수 있습니다.

북마크를 만들려면 먼저 북마크를 만든 다음 링크를 추가해야 합니다.

링크를 클릭하면 페이지가 책갈피가 있는 위치로 스크롤됩니다.

먼저 다음 **id** 속성을 사용하여 책갈피를 작성하십시오.

```
<h2 id="C4">Chapter 4</h2>
```

그런 다음 같은 페이지에서 책갈피에 대한 링크 ("4 장으로 이동")를 추가하십시오.

```
<a href="#C4">Jump to Chapter 4</a>
```

또는 다른 페이지에서 책갈피에 대한 링크 ("4 장으로 이동")를 추가하십시오.

```
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<p><a href="#C4">Jump to Chapter 4</a></p>
<p><a href="#C10">Jump to Chapter 10</a></p>

<h2>Chapter 1</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 2</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 3</h2>
<p>This chapter explains ba bla bla</p>

<h2 id="C4">Chapter 4</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 5</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 6</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 7</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 8</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 9</h2>
<p>This chapter explains ba bla bla</p>

<h2 id="C10">Chapter 10</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 11</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 12</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 13</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 14</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 15</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 16</h2>
```



```
<p>This chapter explains ba bla bla</p>

<h2>Chapter 17</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 18</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 19</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 20</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 21</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 22</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 23</h2>
<p>This chapter explains ba bla bla</p>

</body>
</html>
```

외부 경로

외부 페이지는 전체 URL 또는 현재 웹 페이지에 상대적인 경로를 사용하여 참조 할 수 있습니다.

이 예는 전체 URL 을 사용하여 웹 페이지에 연결합니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>External Paths</h2>
<p>This example uses a full URL to link to a web page:</p>
<p><a href="https://www.w3schools.com/html/default.asp">HTML tutorial</a></p>
</body>
</html>
```

이 예제는 현재 웹 사이트의 html 폴더에있는 페이지로 연결됩니다.

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

```
</head>
<body>

<h2>External Paths</h2>
<p>This example links to a page located in the html folder on the current web
site:</p>
<p><a href="/html/default.asp">HTML tutorial</a></p>
</body>
</html>
```

이 예는 현재 페이지와 같은 폴더에 있는 페이지에 연결됩니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>External Paths</h2>
<p>This example links to a page located in the same folder as the current
page:</p>
<p><a href="default.asp">HTML tutorial</a></p>
</body>
</html>
```

단원 요약

- 링크 정의는 `<a>` 요소를 사용
- 링크 주소 정의는 `href` 속성을 사용
- `target` 속성을 사용하여 링크 된 문서를 열 위치를 정의
- 이미지를 링크로 사용하려면 `` 요소 (`<a>` 내부)를 사용
- 페이지에 책갈피 정의는 `id` 속성 (`ID = "값"`)을 사용
- 북마크에 링크는 `href` 속성 (`HREF = "#값"`)을 사용

HTML 링크 태그

Tag	Description
<code><a></code>	Defines a hyperlink

HTML 이미지

이미지는 웹 페이지의 디자인과 모양을 향상시킬 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>HTML Image</h2>




</body>
</html>
```

HTML 이미지 구문

HTML 에서 이미지는 `` 태그 로 정의됩니다.

`` 태그는 단지 속성을 포함, 비어있는, 그리고 닫는 태그가 없습니다.

이 `src` 속성은 이미지의 URL (웹 주소)을 지정합니다.

```

```

대체 속성

`alt` 어떤 이유로 사용자가 볼 수 없는 경우 속성은, 이미지에 대한 대체 텍스트를 제공한다 (때문에 느린 연결로, `src` 속성의 오류, 또는 사용자가 화면 판독기를 사용하는 경우).

`alt` 속성 값은 이미지를 설명해야 합니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
```

```
<h2>Alternative text</h2>
<p>The alt attribute should reflect the image content, so users who cannot see
the image gets an understanding of what the image contains:</p>


</body>
</html>
```

브라우저가 이미지를 찾을 수 없으면 **alt** 속성 값이 표시됩니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Alternative text</h2>
<p>If a browser cannot find the image, it will display the alternate text:</p>


</body>
</html>
```

참고 : **alt** 속성이 필요합니다. 웹 페이지가 없으면 웹 페이지의 유효성이 올바르게 검사되지 않습니다.

이미지 크기-너비와 높이

이 **style** 속성을 사용하여 이미지의 너비와 높이를 지정할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Image Size</h2>
<p>Use the style attribute to specify the width and height of an image:</p>


</body>
</html>
```

또는 **width** 및 **height** 속성을 사용할 수 있습니다.

```
<!DOCTYPE html>
```

```

<html>
<head>
</head>
<body>

<h2>Image Size</h2>
<p>In this example, we specify the width and height of an image with the width
and height attributes:</p>


</body>
</html>

```

참고 : 항상 이미지의 너비와 높이를 지정하십시오. 너비와 높이를 지정하지 않으면 이미지가 로드 되는 동안 페이지가 깜박일 수 있습니다.

너비와 높이 또는 스타일?

`width`, `height` 및 `style` 속성은 HTML 에서 유효합니다.

그러나 `style` 속성을 사용하는 것이 좋습니다. 스타일 시트가 이미지의 크기를 변경하지 못하게 합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
/* This stylesheet sets the width of all images to 100%: */
img {
  width: 100%;
}
</style>
</head>
<body>

<h2>Styling Images</h2>
<p>The image below has the width attribute set to 128 pixels, but the stylesheet
overrides it, and sets the width to 100%.</p>


<p>The image below uses the style attribute, where the width is set to 128 pixels
which overrides the stylesheet:</p>


</body>
</html>

```

다른 폴더의 이미지

지정하지 않으면 브라우저는 웹 페이지와 동일한 폴더에서 이미지를 찾습니다.

그러나 하위 폴더에 이미지를 저장하는 것이 일반적입니다. 그런 다음 **src** 속성에 폴더 이름을 포함시켜야 합니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Images in Another Folder</h2>
<p>It is common to store images in a sub-folder. You must then include the folder
name in the src attribute:</p>


</body>
</html>
```

다른 서버의 이미지

일부 웹 사이트는 이미지 서버에 이미지를 저장합니다.

실제로 전 세계 모든 웹 주소에서 이미지에 액세스 할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Images on Another Server</h2>


</body>
</html>
```

애니메이션 이미지

HTML 은 애니메이션 GIF 를 허용합니다:

```

<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Animated Images</h2>
<p>The GIF standard allows moving images.</p>


</body>
</html>

```

링크로서의 이미지

이미지를 링크로 사용하려면 `<a>` 태그 안에 `` 태그를 넣으십시오.

```

<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Image as a Link</h2>
<p>The image is a link. You can click on it.</p>
<a href="default.asp">
  
</a>
<p>Add "border:0;" to prevent IE9 (and earlier) from displaying a border around
the image.</p>

</body>
</html>

```

참고 : `border:0;` IE9 (및 이전 버전)가 이미지 주위에 테두리를 표시하지 않도록 추가되었습니다 (이미지가 링크 인 경우).

이미지 플로팅

CSS `float` 속성을 사용하여 이미지를 텍스트의 오른쪽 또는 왼쪽으로 띄우십시오.

```

<!DOCTYPE html>
<html>
<head>
</head>

```

```
<body>

<h2>Floating Images</h2>
<p><strong>Float the image to the right:</strong></p>
<p>
A paragraph with a floating image. A paragraph with a floating image. A paragraph
with a floating image.</p>

<p><strong>Float the image to the left:</strong></p>
<p>
A paragraph with a floating image. A paragraph with a floating image. A paragraph
with a floating image.</p>

</body>
</html>
```

HTML 스크린 리더

스크린 리더는 HTML 코드를 읽고 텍스트를 변환하며 사용자가 내용을 "듣게"할 수 있는 소프트웨어 프로그램입니다. 스크린 리더는 시각 장애가 있거나 학습 장애가 있는 사람들에게 유용합니다.

단원 요약

- 이미지 정의는 HTML `` 요소를 사용
- 이미지의 URL 을 정의는 HTML `src` 속성을 사용
- 이미지를 표시 할 수 없는 경우 대체 텍스트 정의는 HTML `alt` 속성을 사용
- 이미지 크기를 정의는 HTML `width` 및 `height` 속성을 사용
- (또는) CSS `width` 와 `height` 속성을 사용하여 이미지의 크기를 정의
- CSS `float` 속성을 사용하여 이미지를 띄웁니다
- HTML `<map>` 요소를 사용하여 이미지 맵을 정의 하십시오.
- HTML `<area>` 요소를 사용하여 이미지 맵에서 클릭 가능한 영역을 정의하십시오.
- HTML `` 의 element `usemap` 속성을 사용하여 이미지 맵을 가리 킵니다.
- HTML `<picture>` 요소를 사용하여 장치마다 다른 이미지를 표시

참고 : 이미지를 로드 하는 데 시간이 걸립니다. 큰 이미지는 페이지 속도를 늦출 수 있습니다. 이미지를 주의해서 사용하십시오.

HTML 이미지 맵

이미지 맵을 사용하면 이미지에서 클릭 가능한 영역을 추가 할 수 있습니다.

이미지 맵

<map> 태그는 이미지 맵을 정의합니다. 이미지 맵은 클릭 가능한 영역이 있는 이미지입니다.

아래 이미지에서 컴퓨터, 전화 또는 커피 잔을 클릭하십시오.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Image Maps</h2>
<p>Click on the computer, the phone, or the cup of coffee to go to a new page and
read more about the topic:</p>


<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">
</map>

</body>
</html>
```

어떻게 작동합니까?

이미지 맵의 기본 개념은 클릭 한 이미지의 종류에 따라 다른 작업을 수행 할 수 있다는 것입니다.

이미지 맵을 만들려면 이미지와 클릭 가능한 영역을 설명하는 규칙이 포함 된 맵이 필요합니다.

이미지

이미지는 `` 태그를 사용하여 삽입됩니다. 다른 이미지와의 유일한 차이점은 `usemap` 속성을 추가해야 한다는 것입니다.

```

```

`usemap` 값은 해시 태그 `#`로 시작 이미지 맵의 이름 다음에, 이미지 및 이미지 맵의 관계를 생성하는데 사용된다.

참고 : 모든 이미지를 이미지 맵으로 사용할 수 있습니다.

지도

그런 다음 `<map>` 요소를 추가하십시오 .

`<map>` 요소는 이미지 맵을 생성하는데 사용되며, `name` 속성을 사용하여 이미지에 연결된다.

```
<map name="workmap">
```

`name` 속성은 `usemap` 속성과 같은 값이 있어야 한다.

참고 : 원하는 위치에 `<map>` 요소를 삽입 할 수 있으며 이미지 바로 뒤에 삽입 할 필요는 없습니다.

지역

그런 다음 클릭 가능한 영역을 추가하십시오.

클릭 가능한 영역은 `<area>` 요소를 사용하여 정의됩니다.

모양

영역의 모양을 정의해야 하며 다음 값 중 하나를 선택할 수 있습니다.

- **rect** -직사각형 영역을 정의
- **circle** -원형 영역을 정의
- **poly** -다각형 영역을 정의
- **default** -전체 지역을 정의

좌표

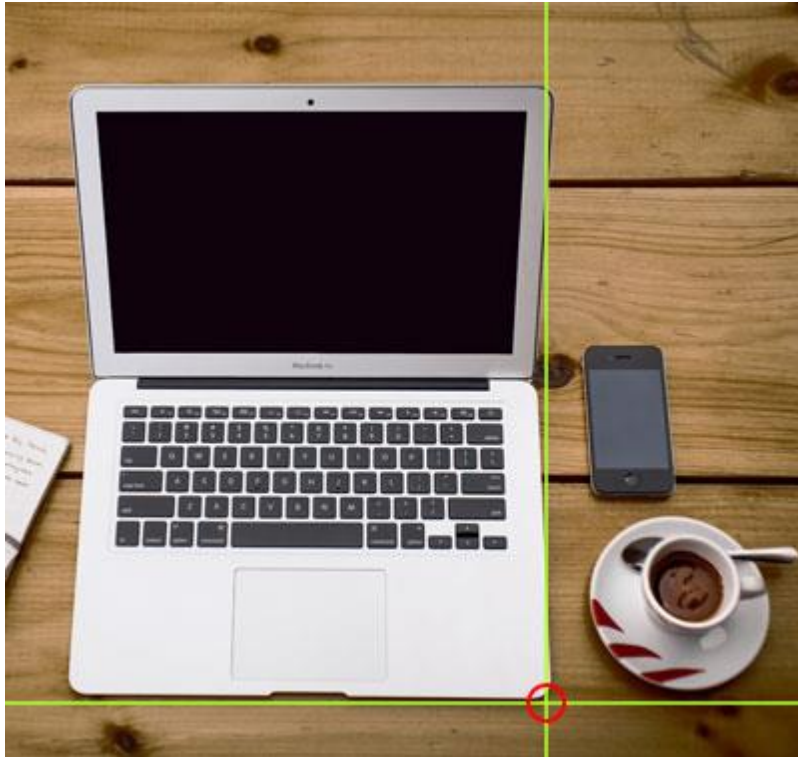
클릭 가능한 영역을 이미지에 배치 할 수 있도록 일부 좌표를 정의해야 합니다.

좌표는 x 축과 y 축에 대해 하나씩 쌍을 이룹니다.

좌표 **34, 44** 는 왼쪽 여백에서 34 픽셀, 위쪽에서 44 픽셀에 있습니다.



좌표 **270, 350** 는 왼쪽 여백에서 270 픽셀, 위쪽에서 350 픽셀에 있습니다.



이제 클릭 가능한 직사각형 영역을 생성하기에 충분한 데이터가 있습니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Image Maps</h2>
<p>Click on the computer, to go to a new page and read more about the topic:</p>


<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
</map>

</body>
</html>
```

클릭 할 수 있는 영역으로 사용자를 페이지 `computer.htm` 으로 보냅니다.

원

원 영역을 추가하려면 먼저 원 중심의 좌표를 찾으십시오.

337, 300



그런 다음 원의 반경을 지정하십시오.

44 픽셀



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

```

<head>
</head>
<body>

<h2>Image Maps</h2>
<p>Click on the computer, to go to a new page and read more about the topic:</p>


<map name="workmap">
  <area shape="circle" coords="337, 300, 44" href="coffee.htm">
</map>

</body>
</html>

```

클릭 할 수 있는 영역으로 사용자를 coffee.htm 페이지로 보냅니다.

이미지 맵 및 JavaScript

클릭 가능한 영역은 다른 페이지에 대한 링크 일 필요는 없으며 JavaScript 기능을 트리거할 수도 있습니다.

JavaScript 함수를 실행하려면 `<area>` 요소에 `click` 이벤트를 추가하십시오.

```

<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Image Maps</h2>
<p> Click on the cup of coffee to execute a JavaScript function:</p>


<map name="workmap">
  <area shape="circle" coords="337,300,44" onclick="myFunction()">
</map>

<script>
  function myFunction() {
    alert("You clicked the coffee cup!");
  }
</script>

</body>
</html>

```

`onclick` 영역을 클릭 할 때 속성을 사용하여 JavaScript 함수를 실행할 수 있습니다.

HTML 배경 이미지

배경 이미지

거의 모든 HTML 요소에 배경 이미지를 지정할 수 있습니다.

HTML로 배경 이미지를 추가하려면 CSS `background-image` 속성을 사용하십시오.

HTML 요소의 배경 이미지

HTML 요소에 배경 이미지를 추가하려면 다음 `style` 속성을 사용할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Background Image</h2>

<div style="background-image: url('img_girl.jpg');">
You can specify background images<br>
for any visible HTML element.<br>
In this example, the background image<br>
is specified for a div element.<br>
By default, the background-image<br>
will repeat itself in the direction(s)<br>
where it is smaller than the element<br>
where it is specified. (Try resizing the<br>
browser window to see how the<br>
background image behaves.
</div>

</body>
</html>
```

`<style>` 요소에 배경 이미지를 지정할 수도 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-image: url('img_girl.jpg');
}
```

```

</style>
</head>
<body>

<h2>Background Image</h2>

<div>
You can specify background images<br>
for any visible HTML element.<br>
In this example, the background image<br>
is specified for a div element.<br>
By default, the background-image<br>
will repeat itself in the direction(s)<br>
where it is smaller than the element<br>
where it is specified. (Try resizing the<br>
browser window to see how the<br>
background image behaves.
</div>

</body>
</html>

```

페이지의 배경 이미지

전체 페이지에 배경 이미지가 포함되도록 하려면 `<body>` 요소에 배경 이미지를 지정해야 합니다.

```

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

<h2>Background Image</h2>

<p>By default the background image will repeat itself if it is smaller than the
element where it is specified, in this case the BODY element.</p>

</body>
</html>

```


백그라운드 반복

배경 이미지가 요소보다 작으면 이미지는 요소의 끝에 도달 할 때까지 수평 및 수직으로 반복됩니다.

설명하기 위해 큰 **div** 요소에서 작은 이미지를 배경으로 사용하면 어떻게 되는지 확인하십시오.

이 **background-image** 속성은 div 요소가 끝에 도달 할 때까지 이미지로 채우기를 시도합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url('example_img_girl.jpg');
}
</style>
</head>
<body>

<h2>Background Image</h2>

<p>By default the background image will repeat itself if it is smaller than the
element where it is specified, in this case the BODY element.</p>

</body>
</html>
```

배경 이미지 자체가 반복되지 않도록 하려면 **background-repeat** 속성을 사용하십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url('example_img_girl.jpg');
  background-repeat: no-repeat;
}
</style>
</head>
<body>

<h2>Background No Repeat</h2>

<p>You can avoid the image from being repeated by setting the background-repeat
```

```
property to "no-repeat".</p>
```

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

배경 표지

배경 이미지가 전체 요소를 커버하도록 하려면 `background-size` 속성을 `cover`.

또한 전체 요소를 항상 덮으려면 `background-attachment` 속성을 `fixed` 로 설정하십시오.

보시다시피 이미지는 스트레칭 없이 전체 요소를 덮으며 이미지는 원래 비율을 유지합니다.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
  background-image: url('img_girl.jpg');  
  background-repeat: no-repeat;  
  background-attachment: fixed;  
  background-size: cover;  
}  
</style>  
</head>  
<body>  
  
<h2>Background Cover</h2>  
  
<p>Set the background-size property to "cover" and the background image will  
cover the entire element, in this case the BODY element.</p>  
  
</body>  
</html>
```

배경 스트레치

배경 이미지 스트레치를 요소의 전체 이미지에 맞추려면 `background-size` 속성을 `100% 100%` 다음과 같이 설정할 수 있습니다.

브라우저 창의 크기를 조정하면 이미지가 늘어나지 만 항상 전체 요소를 덮는 것을 볼 수 있습니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url('img_girl.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: 100% 100%;
}
</style>
</head>
<body>

<h2>Background Stretch</h2>

<p>Set the background-size property to "100% 100%" and the background image will
be stretched to cover the entire element, in this case the BODY element.</p>

</body>
</html>

```

HTML 그림 요소

그림 요소를 사용하면 장치 또는 화면 크기에 따라 다른 그림을 표시 할 수 있습니다.

HTML <picture> 요소

HTML5 는 이미지 리소스를 지정할 때 유연성을 높이기 위해 <picture> 요소를 도입했습니다.

<picture> 요소는 몇 개의 <source> 요소를 포함하고, 각각 상이한 이미지 소스를 참조한다. 이렇게 하면 브라우저가 현재보기 및 / 또는 장치에 가장 적합한 이미지를 선택할 수 있습니다.

각 <source> 요소에는 이미지가 가장 적합한 시기를 설명하는 속성이 있습니다.

다른 화면 크기에서 다른 이미지 표시 :

```

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

<h2>The picture Element</h2>

```

```
<picture>
  <source media="(min-width: 650px)" srcset="img_food.jpg">
  <source media="(min-width: 465px)" srcset="img_car.jpg">
  
</picture>
```

<p>Resize the browser to see different versions of the picture loading at different viewport sizes.

The browser looks for the first source element where the media query matches the user's current viewport width, and fetches the image specified in the srcset attribute.**</p>**

<p>The img element is required as the last child tag of the picture declaration block.

The img element is used to provide backward compatibility for browsers that do not support the picture element, or if none of the source tags matched.**</p>**

<p>Note:**** The picture element is not supported in IE12 and earlier or Safari 9.0 and earlier.**</p>**

```
</body>
```

```
</html>
```

참고 : 항상 ****요소를 **<picture>**요소의 마지막 자식 요소로 지정 하십시오 . **<picture>** 요소를 지원하지 않는 브라우저 또는 **<source>**태그가 일치하지 않을 경우 ****요소를 사용한다.

그림 요소를 사용하는 경우

<picture>요소의 주요 목적은 두 가지입니다.

1. 대역폭

작은 화면이나 장치가 있는 경우 큰 이미지 파일을 로드 할 필요가 없습니다. 브라우저는 **<source>** 속성 값과 함께 일치하는 첫 번째 요소를 사용 하고 다음 요소를 무시합니다.

2. 포맷 지원

일부 브라우저 또는 장치는 모든 이미지 형식을 지원하지 않을 수 있습니다. `<picture>` 요소를 사용하면 모든 형식의 이미지를 추가할 수 있으며 브라우저는 인식하는 첫 번째 형식을 사용하고 다음 중 하나를 무시합니다.

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

<h2>The picture Element</h2>

<picture>
  <source srcset="img_avatar.png">
  <source srcset="img_girl.jpg">
  
</picture>

<p>The picture element can be used when the image format is not supported by all devices.</p>

<p>The device will use the first image format it supports, and ignore the rest of the images. </p>

</body>
</html>
```

참고 : 브라우저는 `<source>` 속성 값이 일치하는 첫 번째 요소를 사용하고 다음 `<source>` 요소는 무시합니다 .

HTML 이미지 태그

Tag	Description
	Defines an image
<map>	Defines an image-map
<area>	Defines a clickable area inside an image-map
<picture>	Defines a container for multiple image resources

HTML 테이블

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
table {
  font-family: arial, sans-serif;
  border-collapse: collapse;
  width: 100%;
}
th, td {
  border: 1px solid #dddddd;
  text-align: left;
  padding: 8px;
}
tr:nth-child(even) {
  background-color: #dddddd;
}
</style>
</head>
<body>

<h2>HTML Table</h2>

<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</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>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>
</table>

</body>
</html>
```

HTML 테이블 정의

HTML 테이블은 `<table>` 태그로 정의됩니다.

각 테이블 행은 `<tr>` 태그로 정의됩니다. 테이블 헤더는 `<th>` 태그로 정의됩니다. 기본적으로 테이블 제목은 굵게 표시되고 가운데에 표시됩니다. 테이블 데이터 / 셀은 `<td>` 태그로 정의됩니다.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<h2>Basic HTML Table</h2>

<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>

</body>
```

```
</html>
```

참고 `<td>` 요소는 테이블 데이터 컨테이너.

여기에는 모든 종류의 HTML 요소가 포함될 수 있습니다. 텍스트, 이미지, 목록, 기타 테이블 등

HTML 테이블-테두리 추가

테이블의 테두리를 지정하지 않으면 테두리 없이 표시됩니다.

CSS `border` 속성을 사용하여 테두리가 설정 됩니다.

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

<h2>Bordered Table</h2>
<p>Use the CSS border property to add a border to the table.</p>

<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>

</body>
```



```
</html>
```

테이블 및 테이블 셀 모두에 대한 경계를 정의해야 합니다.

HTML 테이블-접힌 테두리

테두리를 하나의 테두리로 축소하려면 CSS `border-collapse` 속성을 추가하십시오.

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

<h2>Bordered Table</h2>
<p>Use the CSS border property to add a border to the table.</p>

<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>

</body>
</html>
```

HTML 테이블-셀 패딩 추가

셀 패딩은 셀 내용과 테두리 사이의 간격을 지정합니다.

패딩을 지정하지 않으면 테이블 셀이 패딩 없이 표시됩니다.

패딩을 설정하려면 CSS `padding` 속성을 사용하십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 15px;
  border-collapse: collapse;
}
</style>
</head>
<body>

<h2>Cellpadding</h2>
<p>Cell padding specifies the space between the cell content and its borders.</p>

<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>

<p>Try to change the padding to 5px.</p>

</body>
```

```
</html>
```

HTML 테이블-왼쪽 정렬 제목

기본적으로 테이블 제목은 굵게 표시되고 가운데에 표시됩니다.

테이블 머리글을 왼쪽 정렬하려면 CSS `text-align` 속성을 사용하십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 5px;
}
th {
  text-align: left;
}
</style>
</head>
<body>

<h2>Left-align Headings</h2>
<p>To left-align the table headings, use the CSS text-align property.</p>

<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>
```

```
<p>Try to change the padding to 5px.</p>
</body>
</html>
```

참고 : 표가 테두리를 축소 한 경우 **border-spacing**에는 효과가 없습니다.

HTML 테이블-많은 열로 확장되는 셀

셀이 둘 이상의 열에 걸쳐 있도록 하려면 **colspan** 속성을 사용하십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 5px;
}
th {
  text-align: left;
}
</style>
</head>
<body>

<h2>Cell that spans two columns</h2>
<p>To make a cell span more than one column, use the colspan attribute.</p>

<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>

<p>Try to change the padding to 5px.</p>

</body>
</html>
```

참고 : 표가 테두리를 축소 한 경우 **border-spacing**에는 효과가 없습니다.

HTML 테이블-여러 행으로 확장되는 셀

셀이 둘 이상의 행에 걸쳐 있도록 하려면 `rowspan` 속성을 사용하십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 5px;
  text-align: left;
}
</style>
</head>
<body>

<h2>Cell that spans two rows</h2>
<p>To make a cell span more than one column, use the colspan attribute.</p>

<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>

<p>Try to change the padding to 5px.</p>

</body>
</html>
```

HTML 테이블-캡션 추가

테이블에 캡션을 추가하려면 `<caption>` 태그를 사용하십시오.

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

```

<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 5px;
  text-align: left;
}
</style>
</head>
<body>

<h2>Table Caption</h2>
<p>To add a caption to a table, use the caption tag.</p>

<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>

</body>
</html>

```

참고 : <caption> 태그는 <table> 태그 바로 뒤에 삽입해야 합니다.

하나의 테이블을 위한 특별한 스타일

특수 테이블의 특수 스타일을 정의하려면 테이블에 **id** 속성을 추가 하십시오.

```

<table id="t01">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>

```

```
</table>
```

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
th, td {
    padding: 15px;
    text-align: left;
}
table#t01 {
    width: 100%;
    background-color: #f1f1c1;
}
</style>
</head>
<body>

<h2>Styling Tables</h2>

<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>
<br>

<table id="t01">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
```

```

        <td>Smith</td>
        <td>50</td>
    </tr>
    <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
    </tr>
    <tr>
        <td>John</td>
        <td>Doe</td>
        <td>80</td>
    </tr>
</table>

</body>
</html>

```

```

<!DOCTYPE html>
<html>
<head>
<style>
table {
    width: 100%;
}
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
th, td {
    padding: 15px;
    text-align: left;
}
table#t01 tr:nth-child(even) {
    background-color: #eee;
}
table#t01 tr:nth-child(odd) {
    background-color: #fff;
}
table#t01 th {
    color: white;
    background-color: black;
}
</style>
</head>
<body>

<h2>Styling Tables</h2>

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

```



```

        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
    </tr>
    <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
    </tr>
    <tr>
        <td>John</td>
        <td>Doe</td>
        <td>80</td>
    </tr>
</table>
<br>

<table id="t01">
    <tr>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Age</th>
    </tr>
    <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
    </tr>
    <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
    </tr>
    <tr>
        <td>John</td>
        <td>Doe</td>
        <td>80</td>
    </tr>
</table>

</body>
</html>

```

단원 요약

- HTML `<table>` 요소를 사용하여 테이블 정의
- HTML `<tr>` 요소를 사용하여 테이블 행 정의
- HTML `<td>` 요소를 사용하여 테이블 데이터 정의
- HTML `<th>` 요소를 사용하여 테이블 제목을 정의 하십시오.
- HTML `<caption>` 요소를 사용하여 테이블 캡션 정의

- CSS **border** 속성을 사용하여 테두리 정의
- CSS **border-collapse** 속성을 사용하여 셀 테두리 축소
- CSS **padding** 속성을 사용하여 셀에 패딩 추가
- CSS **text-align** 속성을 사용하여 셀 텍스트 정렬
- CSS **border-spacing** 속성을 사용하여 셀 사이의 간격을 설정하십시오
- 셀을 여러 열로 확장 하려면 **colspan** 속성을 사용하십시오.
- 셀을 여러 행으로 확장 하려면 **rowspan** 속성을 사용하십시오.
- **id** 속성을 사용하여 하나의 테이블을 고유하게 정의하십시오.

HTML 테이블 태그

Tag	Description
<u><table></u>	Defines a table
<u><th></u>	Defines a header cell in a table
<u><tr></u>	Defines a row in a table
<u><td></u>	Defines a cell in a table
<u><caption></u>	Defines a table caption
<u><colgroup></u>	Specifies a group of one or more columns in a table for formatting
<u><col></u>	Specifies column properties for each column within a <code><colgroup></code> element
<u><thead></u>	Groups the header content in a table
<u><tbody></u>	Groups the body content in a table
<u><tfoot></u>	Groups the footer content in a table

HTML 목록

정렬되지 않은 HTML 목록

순서가 없는 목록은 태그로 시작합니다. 각 목록 항목은 `` 태그로 시작합니다.

목록 항목은 기본적으로 글 머리 기호 (작은 검은 색 원)로 표시됩니다.

```
<!DOCTYPE html>
```

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

<h2>An unordered HTML list</h2>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

</body>
</html>
```

정렬되지 않은 HTML 목록-목록 항목 마커 선택

CSS `list-style-type` 속성은 목록 항목 마커의 스타일을 정의하는 데 사용됩니다.

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

예-디스크

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

<h2>Unordered List with Disc Bullets</h2>

<ul style="list-style-type:disc;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

</body>
</html>
```

예-서클

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

<h2>Unordered List with Circle Bullets</h2>

<ul style="list-style-type:circle;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

</body>
</html>
```

예-정사각형

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

<h2>Unordered List with Square Bullets</h2>

<ul style="list-style-type:square;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

</body>
</html>
```

예-없음

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

<h2>Unordered List without Bullets</h2>

<ul style="list-style-type:none;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

</body>
</html>
```

정렬 된 HTML 목록

정렬 된 목록은 태그로 시작합니다. 각 목록 항목은 `` `` 태그로 시작합니다.

목록 항목은 기본적으로 숫자로 표시됩니다.

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

<h2>An ordered HTML list</h2>

<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```

정렬 된 HTML 목록-유형 속성

`` 태그의 `type` 속성은 목록 항목 마커의 유형을 정의합니다.

Type	Description
<code>type="1"</code>	The list items will be numbered with numbers (default)
<code>type="A"</code>	The list items will be numbered with uppercase letters
<code>type="a"</code>	The list items will be numbered with lowercase letters
<code>type="I"</code>	The list items will be numbered with uppercase roman numbers
<code>type="i"</code>	The list items will be numbered with lowercase roman numbers

번호:

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

```
<h2>Ordered List with Numbers</h2>
```

```
<ol type="1">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

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

대문자 :

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

```
<h2>Ordered List with Letters</h2>
```

```
<ol type="A">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

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

소문자 :

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

```
<h2>Ordered List with Lowercase Letters</h2>
```

```
<ol type="a">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

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

대문자 로마 숫자 :

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

<h2> Ordered List with Roman Numbers</h2>

<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```

소문자 로마 숫자 :

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

<h2>Ordered List with Lowercase Roman Numbers</h2>

<ol type="i">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```

HTML 설명 목록

HTML 은 설명 목록도 지원합니다.

설명 목록은 각 용어에 대한 설명이 포함 된 용어 목록입니다.

[<dl>](#)태그는 상기 설명 목록을 정의 [<dt>](#)태그 용어 (이름)를 정의하고, [<dd>](#)태그는 각각의 용어 설명한다.

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

```
<body>

<h2>A Description List</h2>

<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>

</body>
</html>
```

중첩 된 HTML 목록

리스트는 중첩 될 수 있습니다 (리스트 내부의 리스트) :

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

<h2>A Nested List</h2>
<p>List can be nested (lists inside lists):</p>

<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>

</body>
</html>
```

참고 : 목록 항목에는 새 목록과 이미지 및 링크 등의 다른 HTML 요소가 포함될 수 있습니다.

컨트롤리스트 카운팅

기본적으로 정렬 된 목록은 1 부터 계산을 시작합니다. 지정된 숫자부터 계산을 시작하려면 다음 **start** 속성을 사용할 수 있습니다.


```

<!DOCTYPE html>
<html>
<body>

<h2>The start attribute </h2>
<p>By default, an ordered list will start counting from 1. Use the start
attribute to start counting from a specified number: </p>

<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

<ol type="I" start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>

```

CSS가 포함 된 가로 목록

HTML 목록은 CSS 를 사용하여 다양한 방식으로 스타일을 지정할 수 있습니다.

인기있는 방법 중 하나는 목록을 가로로 스타일을 지정하여 탐색 메뉴를 만드는 것입니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 16px;
  text-decoration: none;
}

```

```

}

li a:hover {
    background-color: #111111;
}
</style>
</head>
<body>

<h2>Navigation Menu</h2>
<p>In this example, we use CSS to style the list horizontally, to create a
navigation menu:</p>

<ul>
    <li><a href="#home">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>

```

단원 요약

- HTML `` 요소를 사용하여 순서가 없는 목록을 정의하십시오.
- CSS `list-style-type` 속성을 사용하여 목록 항목 마커를 정의하십시오.
- HTML `` 요소를 사용하여 정렬 된 목록을 정의하십시오.
- HTML `type` 속성을 사용하여 번호 매기기 유형을 정의하십시오.
- HTML `` 요소를 사용하여 목록 항목 정의
- HTML `<dl>` 요소를 사용하여 설명 목록을 정의하십시오.
- HTML `<dt>` 요소를 사용하여 설명 용어를 정의하십시오.
- 설명 요소에서 용어를 설명 하려면 HTML `<dd>` 요소를 사용하십시오.
- 리스트는 리스트 안에 중첩 될 수 있습니다
- 목록 항목은 다른 HTML 요소를 포함 할 수 있습니다
- CSS `float:left` 하거나 `display:inline` 속성을 사용하여 목록을 가로로 표시

HTML 목록 태그

Tag	Description
	Defines an unordered list
	Defines an ordered list
	Defines a list item
<dl>	Defines a description list
<dt>	Defines a term in a description list
<dd>	Describes the term in a description list

HTML 블록 및 인라인 요소

모든 HTML 요소에는 요소 유형에 따라 기본 표시 값이 있습니다.

표시 값은 블록과 인라인입니다.

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.

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

<div style="border: 1px solid black">Hello World</div>
<p>The DIV element is a block element, and will always start on a new line and
take up the full width available (stretches out to the left and right as far as
it can).</p>

</body>
</html>
```

HTML의 블록 레벨 요소 :

```
<address>
<article>
```

```
<aside>
<blockquote>
<canvas>
<dd>
<div>
<dl>
<dt>
<fieldset>
<figcaption>
<figure>
<footer>
<form>
<h1>_<h6>
<header>
<hr>
<li>
<main>
<nav>
<noscript>
<ol>
<p>
<pre>
<section>
<table>
<tfoot>
<ul>
<video>
```

Inline Elements

인라인 요소는 새 줄에서 시작되지 않으며 필요한 만큼만 폭을 차지합니다.

이것은 단락 안의 인라인 `` 요소입니다.

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

<div style="border: 1px solid black">Hello World</div>
<p>This is an inline span <span style="border: 1px solid black">Hello
World</span> element inside a paragraph.</p>

<p>The SPAN element is an inline element, and will not start on a new line and
only takes up as much width as necessary.</p>

</body>
</html>
```

HTML의 인라인 요소 :

```
<a>
<abbr>
<acronym>
<b>
<bdo>
<big>
<br>
<button>
<cite>
<code>
<dfn>
<em>
<i>
<img>
<input>
<kbd>
<label>
<map>
<object>
<output>
<q>
<samp>
<script>
<select>
<small>
<span>
<strong>
<sub>
<sup>
<textarea>
<time>
<tt>
<var>
```

<div> 요소

이 <div> 요소는 종종 다른 HTML 요소의 컨테이너로 사용됩니다.

<div> 요소에 속성이 필요 없지만, **style**, **class** 그리고 **id** 일반적으로 있다.

CSS 와 함께 사용하면 <div> 요소를 사용하여 콘텐츠 블록의 스타일을 지정할 수 있습니다.

```
<!DOCTYPE html>
```

```
<html>
<body>

<div style="background-color:black;color:white;padding:20px;">
  <h2>London</h2>
  <p>London is the capital city of England. It is the most populous city in the
United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
  <p>Standing on the River Thames, London has been a major settlement for two
millennia, its history going back to its founding by the Romans, who named it
Londinium.</p>
</div>

</body>
</html>
```

 요소

이 `` 요소는 종종 일부 텍스트의 컨테이너로 사용됩니다.

`` 요소에 속성이 필요 없지만, `style`, `class` 그리고 `id` 일반적으로 있다.

CSS 와 함께 사용하면 `` 요소를 사용하여 텍스트 부분의 스타일을 지정할 수 있습니다.

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

<h1>My <span style="color:red">Important</span> Heading</h1>

</body>
</html>
```

HTML 그룹화 태그

Tag	Description
<code><div></code>	Defines a section in a document (block-level)
<code></code>	Defines a section in a document (inline)

HTML 클래스 속성

클래스 속성 사용

HTML `class` 속성은 클래스 이름이 동일한 요소에 동일한 스타일을 정의하는 데 사용됩니다.

따라서 동일한 `class` 속성을 가진 모든 HTML 요소는 동일한 스타일을 갖습니다.

여기 `<div>`에 같은 클래스 이름을 가리키는 세 가지 요소가 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.cities {
  background-color: black;
  color: white;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="cities">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>

<div class="cities">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>

<div class="cities">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```

인라인 요소에서 클래스 속성 사용

HTML `class` 속성은 인라인 요소에서도 사용할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
span.note {
  font-size: 120%;
  color: red;
}
</style>
</head>
<body>

<h1>My <span class="note">Important</span> Heading</h1>
<p>This is some <span class="note">important</span> text.</p>

</body>
</html>
```

팁 : 이 `class` 속성은 모든 HTML 요소 에서 사용할 수 있습니다 .

참고 : 클래스 이름은 대소 문자를 구분합니다!

특정 클래스를 가진 요소 선택

CSS 에서 특정 클래스를 가진 요소를 선택하려면 마침표 (.) 문자와 클래스 이름을 작성하십시오.

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
</head>
<body>

<h2>The class Attribute</h2>
<p>Use CSS to style elements with the class name "city":</p>
```



```
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>

</body>
</html>
```

CSS를 사용하여 클래스 이름이 "city"인 모든 요소의 스타일을 지정하십시오.

여러 클래스

HTML 요소는 둘 이상의 클래스 이름을 가질 수 있으며 각 클래스 이름은 공백으로 구분해야 합니다.

클래스 이름이 "city"인 스타일 요소와 클래스 이름이 "main"인 스타일 요소 :

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
.main {
  text-align: center;
}
</style>
</head>
<body>

<h2>Multiple Classes </h2>
<p>All three headers have the class name "city". In addition, London also have
the class name "main", which center-aligns the text. </p>

<h2 class="city main">London</h2>
<h2 class="city">Paris</h2>
<h2 class="city">Tokyo</h2>

</body>
</html>
```

위의 예에서 첫 번째 `<h2>` 요소는 "city" 클래스와 "main" 클래스 모두에 속합니다.

다른 태그는 동일한 클래스를 공유 할 수 있습니다

`<h2>`과 `<p>` 같이 다른 태그도 같은 클래스 이름을 함으로써 같은 스타일을 공유 할 수 있습니다 :

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
</head>
<body>

<h2>Same Class, Different Tag</h2>
<p>Even if the two elements do not have the same tag name, they can have the same
class name, and get the same styling:</p>

<h2 class="city">Paris</h2>
<p class="city">Paris is the capital of France</p>

</body>
</html>
```

JavaScript에서 클래스 속성 사용

JavaScript 는 클래스 이름을 사용하여 지정된 클래스 이름을 가진 요소에 대한 특정 작업을 수행 할 수도 있습니다.

JavaScript 는 다음 `getElementsByClassName()` 메소드를 사용하여 지정된 클래스 이름을 가진 요소에 액세스 할 수 있습니다.

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

<h2>Using The class Attribute in JavaScript</h2>
<p>Click the button, to hide all elements with the class name "city", with
JavaScript:</p>
```

```

<button onclick="myFunction()">Hide elements</button>

<h2 class="city">London</h2>
<p class="city">London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p class="city">Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p class="city">Tokyo is the capital of Japan.</p>

<script>
function myFunction() {
  var x = document.getElementsByClassName("city");
  for (var i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
}
</script>

</body>
</html>

```

HTML id 속성

id 속성 사용

이 **id** 속성은 HTML 요소의 고유 ID 를 지정합니다 (같은 HTML 문서 내에서 고유해야 함).

CSS 및 JavaScript 는 id 값을 사용하여 특정 id 값을 가진 요소에 대한 특정 작업을 수행 할 수 있습니다.

CSS 에서 특정 ID 를 가진 요소를 선택하려면 해시 (#) 문자를 쓰고 요소의 ID 를 입력하십시오.

```

<!DOCTYPE html>
<html>
<head>
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
</head>

```

```
<body>

<h2>The id Attribute</h2>
<p>Use CSS to style an element with the id "myHeader":</p>

<h1 id="myHeader">My Header</h1>

</body>
</html>
```

팁 : id 속성은 모든 HTML 요소 에서 사용할 수 있습니다 .

참고 : id 값은 대소 문자를 구분합니다.

참고 : id 값은 하나 이상의 문자를 포함해야 하며 공백 (공백, 탭 등)을 포함 해서는 안됩니다 .

클래스와 ID의 차이점

HTML 요소는 해당 단일 요소에 속하는 하나의 고유 한 ID 만 가질 수 있지만 클래스 이름은 여러 요소에서 사용할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
/* Style the element with the id "myHeader" */
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}

/* Style all elements with the class name "city" */
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
</head>
<body>

<h2>Difference Between Class and ID</h2>
<p>An HTML page can only have one unique id applied to one specific element,
while a class name can be applied to multiple elements.</p>

<!-- A unique element -->
<h1 id="myHeader">My Cities</h1>
```

```
<!-- Multiple similar elements -->
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>

</body>
</html>
```

ID와 링크가있는 책갈피

HTML 책갈피는 독자가 웹 페이지의 특정 부분으로 이동할 수 있도록 하는 데 사용됩니다.

웹 페이지가 너무 길면 북마크가 유용 할 수 있습니다.

북마크를 만들려면 먼저 북마크를 만든 다음 링크를 추가해야 합니다.

링크를 클릭하면 페이지가 책갈피가 있는 위치로 스크롤됩니다.

예

먼저 다음 `id` 속성을 사용하여 책갈피를 작성하십시오.

```
<h2 id="C4">Chapter 4</h2>
```

그런 다음 같은 페이지에서 책갈피에 대한 링크 ("4 장으로 이동")를 추가하십시오.

```
<a href="#C4">Jump to Chapter 4</a>
```

또는 다른 페이지에서 책갈피에 대한 링크 ("4 장으로 이동")를 추가하십시오.

```
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```

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

<p><a href="#C4">Jump to Chapter 4</a></p>
<p><a href="#C10">Jump to Chapter 10</a></p>
```

```
<h2>Chapter 1</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 2</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 3</h2>
<p>This chapter explains ba bla bla</p>

<h2 id="C4">Chapter 4</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 5</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 6</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 7</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 8</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 9</h2>
<p>This chapter explains ba bla bla</p>

<h2 id="C10">Chapter 10</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 11</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 12</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 13</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 14</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 15</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 16</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 17</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 18</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 19</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 20</h2>
```

```
<p>This chapter explains ba bla bla</p>

<h2>Chapter 21</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 22</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 23</h2>
<p>This chapter explains ba bla bla</p>

</body>
</html>
```

JavaScript에서 id 속성 사용

JavaScript 는 다음 `getElementById()` 메소드를 사용하여 지정된 ID 를 가진 요소에 액세스 할 수 있습니다.

JavaScript로 텍스트를 조작하려면 id 속성을 사용하십시오.

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

<h2>Using The id Attribute in JavaScript</h2>
<p>JavaScript can access an element with a specified id by using the
getElementById() method:</p>

<h1 id="myHeader">Hello World!</h1>
<button onclick="displayResult()">Change text</button>

<script>
function displayResult() {
    document.getElementById("myHeader").innerHTML = "Have a nice day!";
}
</script>

</body>
</html>
```

HTML Iframe

iframe은 웹 페이지 내에 웹 페이지를 표시하는 데 사용됩니다.

Iframe 구문

HTML iframe 은 다음 `<iframe>` 태그로 정의됩니다.

```
<iframe src="URL"></iframe>
```

이 `src` 속성은 인라인 프레임 페이지의 URL (웹 주소)을 지정합니다.

Iframe-높이 및 너비 설정

iframe 이의 크기를 지정하는 속성은 `height` 하고 `width` 를 사용.

높이와 너비는 기본적으로 픽셀 단위로 지정됩니다.

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

<h2>HTML Iframes</h2>
<p>You can use the height and width attributes to specify the size of the
iframe:</p>

<iframe src="demo_iframe.htm" height="200" width="300"></iframe>

</body>
</html>
```

또는 CSS를 사용하여 iframe의 높이와 너비를 설정할 수 있습니다.

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

<h2>HTML Iframes</h2>
<p>You can also use the CSS height and width properties to specify the size of
the iframe:</p>

<iframe src="demo_iframe.htm" style="height:200px;width:300px;"></iframe>

</body>
</html>
```


Iframe-테두리 제거

기본적으로 iframe 주위에는 테두리가 있습니다.

테두리를 제거하려면 `style` 속성을 추가 하고 CSS `border` 속성을 사용하십시오.

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

<h2>Remove the Iframe Border</h2>
<p>To remove the default border of the iframe, use CSS:</p>

<iframe src="demo_iframe.htm" style="border:none;"></iframe>

</body>
</html>
```

CSS를 사용하면 iframe 테두리의 크기, 스타일 및 색상을 변경할 수도 있습니다.

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

<h2>Custom Iframe Border</h2>
<p>With CSS, you can also change the size, style and color of the iframe's border:</p>

<iframe src="demo_iframe.htm" style="border:2px solid red;"></iframe>

</body>
</html>
```

Iframe-링크 대상

링크의 대상 프레임으로 iframe 을 사용할 수 있습니다.

링크의 `target` 속성은 iframe 의 `name` 속성을 참조해야 합니다 :

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

<h2>Iframe - Target for a Link</h2>

<iframe src="demo_iframe.htm" name="iframe_a" height="300px">
```

```
width="100%"></iframe>

<p><a href="https://www.w3schools.com" target="iframe_a">W3Schools.com</a></p>

<p>When the target of a link matches the name of an iframe, the link will open in
the iframe.</p>

</body>
</html>
```

HTML iframe 태그

Tag	Description
<u><iframe></u>	Defines an inline frame

HTML 자바 스크립트

JavaScript는 HTML 페이지를 보다 역동적이고 대화식으로 만듭니다.

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

<h1>My First JavaScript</h1>

<button type="button" onclick="document.getElementById('demo').innerHTML
Date()">Click me to display Date and Time.</button>

<p id="demo"></p>

</body>
</html>
```

HTML <script> 태그

이 **<script>** 태그는 클라이언트 측 스크립트 (JavaScript)를 정의하는 데 사용됩니다.

<script> 요소 중 하나 스크립트 문을 포함, 또는 그것을 **src** 속성을 통해 외부 스크립트 파일을 가리킨다.

JavaScript 의 일반적인 용도는 이미지 조작, 양식 유효성 검사 및 동적 콘텐츠 변경입니다.

HTML 요소를 선택하기 위해 JavaScript 는 대부분이 `document.getElementById()` 방법을 사용합니다.

이 JavaScript 예제는 HTML 요소의 `id = "demo"`를 사용하여 "Hello JavaScript!"를 작성합니다.

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

<h2>Use JavaScript to Change Text</h2>

<p>This example writes "Hello JavaScript!" into an HTML element with
id="demo":</p>
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>

</body>
</html>
```

자바 스크립트 맛

다음은 JavaScript 로 수행 할 수 있는 작업의 예입니다.

JavaScript는 HTML 내용을 변경할 수 있습니다

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

<h1>My First JavaScript</h1>

<p>JavaScript can change the content of an HTML element:</p>

<button type="button" onclick="myFunction()">Click Me!</button>

<p id="demo">This is a demonstration.</p>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
```

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

JavaScript는 HTML 스타일을 변경할 수 있습니다

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

<h1>My First JavaScript</h1>

<p id="demo">JavaScript can change the style of an HTML element.</p>

<script>
function myFunction() {
    document.getElementById("demo").style.fontSize = "25px";
    document.getElementById("demo").style.color = "red";
    document.getElementById("demo").style.backgroundColor = "yellow";
}
</script>

<button type="button" onclick="myFunction()">Click Me!</button>

</body>
</html>
```

JavaScript는 HTML 속성을 변경할 수 있습니다

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

<h1>My First JavaScript</h1>

<p id="demo">JavaScript can change the style of an HTML element.</p>

<script>
function light(sw) {
    var pic;
    if p(sw == 0) {
        pic = "pic_bulboff.gif";
    } else {
        pic = "pic_bulbon.gif";
    }
    document.getElementById("myImage").src = pic;
}
</script>

Click
Me!</button>
```

```
<p>
<button type="button" onclick="light(1)">Light On</button>
<button type="button" onclick="light(0)">Light Off</button>
</p>

</body>
</html>
```

HTML <noscript> 태그

이 **<noscript>** 태그는 브라우저에서 스크립트를 비활성화하거나 클라이언트 쪽 스크립트를 지원하지 않는 브라우저를 가진 사용자에게 대체 콘텐츠를 제공하는 데 사용됩니다.

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

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>

<noscript>Sorry, your browser does not support JavaScript!</noscript>

<p>A browser without support for JavaScript will show the text written inside the
noscript element.</p>

</body>
</html>
```

HTML 스크립트 태그

Tag	Description
<script>	Defines a client-side script
<noscript>	Defines an alternate content for users that do not support client-side scripts

HTML 파일 경로

Path	Description
<code></code>	picture.jpg는 현재 페이지와 동일한 폴더에 있습니다
<code></code>	picture.jpg는 현재 폴더의 이미지 폴더에 있습니다.
<code></code>	picture.jpg는 현재 웹 루트의 이미지 폴더에 있습니다.
<code></code>	picture.jpg는 현재 폴더에서 한 단계 위의 폴더에 있습니다.

HTML 파일 경로

파일 경로는 웹 사이트의 폴더 구조에서 파일의 위치를 설명합니다.

파일 경로는 다음과 같은 외부 파일에 연결할 때 사용됩니다.

- 웹 페이지
- 이미지
- 스타일 시트
- 자바 스크립트

절대 파일 경로

절대 파일 경로는 인터넷 파일의 전체 URL입니다.

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

<h2>Using a Full URL File Path</h2>


```

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

상대 파일 경로

상대 파일 경로는 현재 페이지를 기준으로 파일을 가리 킵니다.

이 예제에서 파일 경로는 현재 웹의 루트에 있는 이미지 폴더의 파일을 가리 킵니다.

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h2>Using a Relative File Path</h2>  
  
  
  
</body>  
</html>
```

이 예에서 파일 경로는 현재 폴더에 있는 images 폴더의 파일을 가리 킵니다.

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h2>Using a Relative File Path</h2>  
  
  
  
</body>  
</html>
```

이 예에서 파일 경로는 현재 폴더보다 한 수준 위에 있는 폴더에 있는 images 폴더의 파일을 가리 킵니다.

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h2>Using a Relative File Path</h2>  
  

```

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

모범 사례

상대 파일 경로를 사용하는 것이 가장 좋습니다 (가능한 경우).

상대 파일 경로를 사용하면 웹 페이지가 현재 기본 URL에 바인딩 되지 않습니다. 모든 링크는 자신의 컴퓨터 (localhost)와 현재 공개 도메인 및 향후 공개 도메인에서 작동합니다.

HTML 헤드

HTML <head> 요소

<head>요소는 <html>태그와 <body>태그 사이에 위치한 메타 데이터(데이터에 대한 데이터) 컨테이너이다.

HTML 메타 데이터는 HTML 문서에 대한 데이터입니다. 메타 데이터가 표시되지 않습니다.

메타 데이터는 일반적으로 문서 제목, 문자 세트, 스타일, 스크립트 및 기타 메타 정보를 정의합니다.

다음 태그는 메타 데이터를

설명 : <title>, <style>, <meta>, <link>, <script>,와 <base>.

HTML <title> 요소

<title>요소는 문서의 제목을 정의하고, 모든 HTML 문서가 필요합니다.

<title>요소 :

- 브라우저 탭에서 제목을 정의합니다
- 즐겨 찾기에 추가 될 때 페이지 제목을 제공합니다.
- 검색 엔진 결과에 페이지 제목을 표시합니다

간단한 HTML 문서 :

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
</head>

<body>

<p>The content of the body element is displayed in the browser window.</p>
<p>The content of the title element is displayed in the browser tab, in favorites
and in search engine results.</p>

</body>

</html>
```

HTML <style> 요소

이 <style> 요소는 단일 HTML 페이지에 대한 스타일 정보를 정의하는 데 사용됩니다.

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
  <style>
    body {background-color: powderblue;}
    h1 {color: red;}
    p {color: blue;}
  </style>
</head>

<body>

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

</body>

</html>
```

HTML <link> 요소

이 `<link>` 요소는 외부 스타일 시트에 연결하는 데 사용됩니다.

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
  <link rel="stylesheet" href="mystyle.css">
</head>

<body>

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

</body>

</html>
```

HTML <meta> 요소

이 `<meta>` 요소는 사용되는 문자 세트, 페이지 설명, 키워드, 작성자 및 기타 메타 데이터를 지정하는 데 사용됩니다.

메타 데이터는 브라우저 (콘텐츠 표시 방법), 검색 엔진 (키워드) 및 기타 웹 서비스에서 사용됩니다.

사용된 문자 세트를 정의하십시오.

```
<meta charset="UTF-8">
```

웹 페이지에 대한 설명을 정의하십시오.

```
<meta name="description" content="Free Web tutorials">
```

검색 엔진에 대한 키워드를 정의하십시오.

```
<meta name="keywords" content="HTML, CSS, XML, JavaScript">
```

페이지 작성자를 정의하십시오.

```
<meta name="author" content="John Doe">
```

30 초마다 문서를 새로 고칩니다.

```
<meta http-equiv="refresh" content="30">
```

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="John Doe">
</head>

<body>

<p>All meta information goes before the body.</p>

</body>

</html>
```

뷰포트 설정

HTML5 는 웹 디자이너가 **<meta>**태그를 통해 뷰포트를 제어 할 수 있는 방법을 도입했습니다.

뷰포트는 웹 페이지에서 사용자가 볼 수 있는 영역입니다. 장치에 따라 다르며 컴퓨터 화면보다 휴대 전화에서 더 작습니다.

<meta> 모든 웹 페이지에 다음 뷰포트 요소를 포함해야 합니다.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

<meta> 뷰포트 요소는 페이지의 크기와 스케일링을 제어하는 방법에 브라우저 지침을 제공합니다.

width = device-width 부분은 장치의 화면 너비에 따라 페이지 너비를 설정합니다 (장치에 따라 다름).

initial-scale = 1.0 부분은 페이지가 브라우저에 의해 처음으로 로드 될 때 초기 줌 레벨을 설정합니다.

다음은 뷰포트 메타 태그가 *없는* 웹 페이지와 뷰포트 **<meta>**태그가 있는 동일한 웹 페이지의 예 *입니다*.

팁 : 휴대 전화 나 태블릿으로 이 페이지를 탐색하는 경우 아래 두 링크를 클릭하여 차이점을 확인할 수 있습니다.



[뷰포트 메타 태그없이](#)



[뷰포트 메타 태그 사용](#)

HTML <script> 요소

이 `<script>` 요소는 클라이언트 측 JavaScript 를 정의하는 데 사용됩니다.

이 JavaScript 는 HTML 요소로 id = "demo"를 사용하여 "Hello JavaScript!" 입니다.

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
  <script>
    function myFunction() {
      document.getElementById("demo").innerHTML = "Hello JavaScript!";
    }
  </script>
</head>

<body>

<h1>My Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

</body>

</html>
```

HTML <base> 요소

이 `<base>` 요소는 페이지의 모든 상대 URL 에 대한 기본 URL 및 기본 대상을 지정합니다.

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
  <base href="https://www.w3schools.com/images/" target="_blank">
</head>

<body>


<p> Since we have specified a base URL, the browser will look for the image
```

```
"html5.gif" at "https://www.w3schools.com/images/html5.gif"</p>
<p><a href="https://www.w3schools.com">W3Schools</a></p>
<p>The link above opens in a new window. This is because the base target is set
to "_blank".</p>

</body>

</html>
```

<html>, <head> 및 <body>를 생략 하시겠습니까?

HTML5 표준에 따르면; <html>는 <body>, 상기 <head>태그는 생략 될 수 있다.

다음 코드는 HTML5 로 유효합니다.

```
<!DOCTYPE html>
<title>Page Title</title>

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

노트 :

W3Schools 는 <html>및 <body> 태그를 생략하지 않는 것이 좋습니다. 이 태그를 생략하면 DOM 또는 XML 소프트웨어가 충돌하여 이전 브라우저 (IE9)에서 오류가 발생할 수 있습니다.

그러나 <head>태그를 생략하는 것은 꽤 오랫동안 일반적인 관행이었습니다.

HTML 헤드 요소

Tag	Description
<head>	Defines information about the document
<title>	Defines the title of a document
<base>	Defines a default address or a default target for all links on a page
<link>	Defines the relationship between a document and an external resource
<meta>	Defines metadata about an HTML document

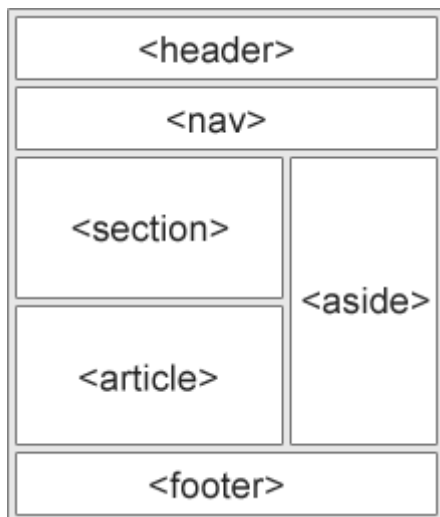
<code><script></code>	Defines a client-side script
<code><style></code>	Defines style information for a document

HTML 레이아웃

HTML 레이아웃 요소

웹 사이트는 종종 잡지 나 신문과 같은 여러 열로 콘텐츠를 표시합니다.

HTML 은 웹 페이지의 다른 부분을 정의하는 몇 가지 의미 요소를 제공합니다.



- `<header>`-문서 또는 섹션의 헤더를 정의합니다
- `<nav>`-탐색 링크를 위한 컨테이너를 정의합니다
- `<section>`-문서에서 섹션을 정의합니다
- `<article>`-독립적 인 기사를 정의합니다
- `<aside>`-사이드 바와 같이 콘텐츠를 제외하고 콘텐츠를 정의합니다.
- `<footer>`-문서 또는 섹션의 바닥 글을 정의합니다
- `<details>`-추가 세부 사항을 정의합니다
- `<summary>`-`<details>` 요소의 제목을 정의합니다

HTML 레이아웃 기법

여러 열 레이아웃을 만드는 5 가지 방법이 있습니다. 각 방법에는 장단점이 있습니다.

- HTML 테이블 (권장되지 않음)
- CSS float 속성
- CSS 플렉스 박스
- CSS 프레임 워크
- CSS 그리드

어느 것을 선택해야 할까요?

HTML 테이블

`<table>` 요소는 레이아웃 도구로 설계되지 않았습니다! `<table>` 요소의 목적은 테이블 형식의 데이터를 표시하는 것입니다. 따라서 페이지 레이아웃에 테이블을 사용하지 마십시오! 그들은 당신의 코드에 혼란을 가져올 것입니다. 몇 달 후에 사이트를 재 설계하는 것이 얼마나 어려운지 상상해보십시오.

팁 : 페이지 레이아웃에 테이블을 사용하지 마십시오!

CSS 프레임 워크

레이아웃을 빠르게 만들려면 [W3.CSS](#) 또는 [Bootstrap](#) 과 같은 프레임 워크를 사용할 수 있습니다 .

CSS 수레

CSS float 속성을 사용하여 전체 웹 레이아웃을 수행하는 것이 일반적입니다. 플로트는 배우기 쉽습니다. 플로트와 클리어 속성이 어떻게 작동하는지 기억하면 됩니다. **단점** : 플로팅 요소가 문서 흐름과 연결되어 있어 유연성이 떨어질 수 있습니다. [CSS Float and Clear](#) 장 에서 float 에 대해 자세히 알아보십시오.

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

<head>
  <title>CSS Template</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    * {
      box-sizing: border-box;
    }

    body {
      font-family: Arial, Helvetica, sans-serif;
    }
    /* Style the header */
    header {
      background-color: #666;
      padding: 30px;
      text-align: center;
      font-size: 35px;
      color: white;
    }

    /* Create two columns/boxes that floats next to each other */
    nav {
      float: left;
      width: 30%;
      height: 300px; /* only for demonstration, should be removed */
      background: #ccc;
      padding: 20px;
    }

    /* Style the list inside the menu */
    nav ul {
      list-style-type: none;
      padding: 0;
    }

    article {
      float: left;
      width: 70%;
      height: 300px; /* only for demonstration, should be removed */
      background-color: #f1f1f1;
      padding: 20px;
```

```

}

/* Clear floats after the columns */
section:after {
    content: "";
    display: table;
    clear: both;
}

/* Style the footer */
footer {
    background-color: #777;
    padding: 10px;
    text-align: center;
    color: white;
}

/* Responsive layout - makes the two columns/boxes stack on top of each other
instead of next to each other, on small screens */
@media (max-width: 600px) {
    nav, article {
        width: 100%;
        height: auto;
    }
}

</style>
</head>

<body>

<h2>CSS Layout Float</h2>
<p>In this example, we have created a header, two columns/boxes and a footer. On
smaller screens, the columns will stack on top of each other.</p>
<p>Resize the browser window to see the responsive effect (you will learn more
about this in our next chapter - HTML Responsive.)</p>

<header>
    <h2>Cities</h2>
</header>

<section>
    <nav>
        <ul>
            <li><a href="#">London</a></li>
            <li><a href="#">Paris</a></li>
            <li><a href="#">Tokyo</a></li>
        </ul>
    </nav>

    <article>
        <h1>London</h1>
        <p>London is the capital city of England. It is the most populous city in
the United Kingdom, with a metropolitan area of over 13 million
inhabitants.</p>
        <p>Standing on the River Thames, London has been a major settlement for two
millennia, its history going back to its founding by the Romans, who named
it Londinium.</p>
    </article>

```

```
</article>
</section>

<footer>
  <p>Footer</p>
</footer>

</body>

</html>
```

CSS Flexbox

Flexbox 는 CSS3 의 새로운 레이아웃 모드입니다.

flexbox 를 사용하면 페이지 레이아웃이 다른 화면 크기와 다른 디스플레이 장치를 수용해야 할 때 요소가 예측 가능하게 작동합니다. **단점 : IE10** 및 이전 버전에서는 작동하지 않습니다.

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

<head>
  <title>CSS Template</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    * {
      box-sizing: border-box;
    }

    body {
      font-family: Arial, Helvetica, sans-serif;
    }
    /* Style the header */
    header {
      background-color: #666;
      padding: 30px;
      text-align: center;
      font-size: 35px;
      color: white;
    }

    /* Container for flexboxes */
    section {
      display: -webkit-flex;
      display: flex;
    }

    /* Create two columns/boxes that floats next to each other */
    nav {
      -webkit-flex: 1;
      -ms-flex: 1;
```

```

    flex: 1;
    background: #ccc;
    padding: 20px;
}

/* Style the list inside the menu */
nav ul {
    list-style-type: none;
    padding: 0;
}

/* Style the content */
article {
    -webkit-flex: 3;
    -ms-flex: 3;
    flex: 3;
    background-color: #f1f1f1;
    padding: 10px;
}

/* Style the footer */
footer {
    background-color: #777;
    padding: 10px;
    text-align: center;
    color: white;
}

/* Responsive layout - makes the menu and the content (inside the section) sit
on top of each other instead of next to each other */
@media (max-width: 600px) {
    section {
        -webkit-flex-direction: column;
        flex-direction: column;
    }
}
</style>
</head>

<body>

<h2>CSS Layout Flexbox</h2>
<p>In this example, we have created a header, two columns/boxes and a footer. On
smaller screens, the columns will stack on top of each other.</p>
<p>Resize the browser window to see the responsive effect.</p>
<p><strong>Note:</strong> Flexbox is not supported in Internet Explorer 10 and
earlier versions.</p>

<header>
  <h2>Cities</h2>
</header>

<section>
  <nav>
    <ul>
      <li><a href="#">London</a></li>
      <li><a href="#">Paris</a></li>
    </ul>
  </nav>

```

```

    <li><a href="#">Tokyo</a></li>
  </ul>
</nav>

<article>
  <h1>London</h1>
  <p>London is the capital city of England. It is the most populous city in
  the United Kingdom, with a metropolitan area of over 13 million
  inhabitants.</p>
  <p>Standing on the River Thames, London has been a major settlement for two
  millennia, its history going back to its founding by the Romans, who named
  it Londinium.</p>
</article>
</section>

<footer>
  <p>Footer</p>
</footer>

</body>

</html>

```

CSS 그리드 뷰

CSS 그리드 레이아웃 모듈은 행과 열이있는 그리드 기반 레이아웃 시스템을 제공하므로 플롯과 위치 지정없이 웹 페이지를보다 쉽게 디자인 할 수 있습니다.

단점 : IE 나 Edge 15 이하에서는 작동하지 않습니다.

HTML 반응형 웹 디자인

반응 형 웹 디자인이란 무엇입니까?

반응형 웹 디자인은 HTML 및 CSS 를 사용하여 웹 사이트의 크기를 자동으로 조정, 숨기기, 축소 또는 확대하여 모든 장치 (데스크톱, 태블릿 및 전화)에서 보기 좋게 보이도록 합니다.

```

<!DOCTYPE html>
<html>
<head>

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
  * {
    box-sizing: border-box;
  }

  .menu {
    float: left;
    width: 20%;
  }
  .menuitem {
    padding: 8px;
    margin-top: 7px;
    border-bottom: 1px solid #f1f1f1;
  }
  .main {
    float: left;
    width: 60%;
    padding: 0 20px;
    overflow: hidden;
  }
  .right {
    background-color: lightblue;
    float: left;
    width: 20%;
    margin-top: 7px;
    padding: 10px 15px;
  }

  @media only screen and (max-width: 800px) {
    /* For tablets */
    .main {
      width: 80%;
      padding: 0;
    }
    .right {
      width: 100%;
    }
  }

  @media only screen and (max-width: 500px) {
    /* For mobile phones */
    .menu, .main, .right {
      width: 100%;
    }
  }
}

</style>
</head>

<body style="overflow:auto">

<div style="background-color:#f1f1f1;padding:15px;">
  <h1>Cinque Terre</h1>
  <h3>Resize the browser window</h3>
</div>

<div style="overflow:auto">

```

```

<div class="menu">
  <div class="menuitem">London</div>
  <div class="menuitem">Transport</div>
  <div class="menuitem">History</div>
  <div class="menuitem">Gallery</div>
</div>

<div class="main">
  <h2>The Walk</h2>
  <p>The walk from Monterosso to Riomaggiore will take you approximately two hours, give or take an hour depending on the weather conditions and your physical shape.</p>
  
</div>

<div class="right">
  <h2>What?</h2>
  <p>Cinque Terre comprises five villages: Monterosso, Vernazza, Corniglia, Manarola, and Riomaggiore.</p>
  <h2>Where?</h2>
  <p>On the northwest cost of the Italian Riviera, north of the city La Spezia.</p>
  <h2>Price?</h2>
  <p>The Walk is free!</p>
</div>
</div>
<div style="background-color:#f1f1f1;text-align:center;padding:10px;margin-top:7px;font-size:12px;">This web page is a part of a demonstration of fluid web design made by w3schools.com. Resize the browser window to see the content respond to the resizing.</div>

</body>

</html>

```

뷰포트 설정

반응형 웹 페이지를 만들 때는 `<meta>` 모든 웹 페이지에 다음 요소를 추가하십시오.

```

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

<body>

<h2>Setting the Viewport</h2>
<p>This example does not really do anything, other than showing you how to add the viewport meta element.</p>

</body>

```

</html>

그러면 페이지의 뷰포트가 설정되어 페이지의 크기와 크기를 제어하는 방법에 대한 브라우저 지침이 제공됩니다.

다음은 뷰포트 메타 태그가 *없는* 웹 페이지 *와* 뷰포트 메타 태그가 있는 동일한 웹 페이지의 예 *입니다*.

뷰포트 메타 태그가 없는 경우 :



뷰포트 메타 태그를 사용하는 경우 :



팁 : 휴대 전화 나 태블릿에서이 페이지를 탐색하는 경우 위의 두 링크를 클릭하여 차이점을 확인할 수 있습니다.

반응형 이미지

반응 형 이미지는 모든 브라우저 크기에 맞게 확장 할 수 있는 이미지입니다.

너비 속성 사용

CSS **width** 속성을 100 %로 설정하면 이미지가 반응형이고 크기가 확대 / 축소됩니다.

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

<body>
```

```
<h2>Responsive Image</h2>
<p>When the CSS width property is set in a percentage value, the image will scale up and down when resizing the browser window. Resize the browser window to see the effect.</p>



</body>

</html>
```

위의 예에서 이미지를 원래 크기보다 크게 확대 할 수 있습니다. 많은 경우에 더 좋은 해결책은 `max-width` 속성을 대신 사용하는 것입니다.

최대 너비 속성 사용

경우 `max-width` 속성을 100 %로 설정하면 이미지는 가있는 경우 축소, 하지만 원래 크기보다 크게 확장하지 않습니다 :

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

<body>

<h2>Responsive Image</h2>
<p>"max-width:100%" prevents the image from getting bigger than its original size. However, if you make the browser window smaller, the image will still scale down.</p>
<p>Resize the browser window to see the effect.</p>



</body>

</html>
```

브라우저 너비에 따라 다른 이미지 표시

HTML `<picture>` 요소를 사용하면 브라우저 창 크기마다 다른 이미지를 정의 할 수 있습니다.

너비에 따라 아래 이미지가 어떻게 변하는 지 보려면 브라우저 창의 크기를 조정하십시오.

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

<body>

<h2>Show Different Images Depending on Browser Width</h2>
<p>Resize the browser width and the image will change at 600px and 1500px.</p>

<picture>
  <source srcset="img_smallflower.jpg" media="(max-width: 600px)">
  <source srcset="img_flowers.jpg" media="(max-width: 1500px)">
  <source srcset="flowers.jpg">
  
</picture>

</body>

</html>
```

반응형 텍스트 크기

텍스트 크기는 "뷰포트 너비"를 의미하는 "vw"단위로 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 1vw는 0.5cm입니다.

미디어 쿼리

텍스트와 이미지의 크기를 조정하는 것 외에도 반응형 웹 페이지에서 미디어 쿼리를 사용하는 것이 일반적입니다.

미디어 쿼리를 사용하면 다양한 브라우저 크기에 대해 완전히 다른 스타일을 정의할 수 있습니다.

예 : 아래 3 개의 **div** 요소가 큰 화면에 가로로 표시되고 작은 화면에 세로로 쌓이도록 브라우저 창의 크기를 조정하십시오.

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
* {
  box-sizing: border-box;
}

.left {
  background-color: #2196F3;
  padding: 20px;
  float: left;
  width: 20%; /* The width is 20%, by default */
}

.main {
  background-color: #f1f1f1;
  padding: 20px;
  float: left;
  width: 60%; /* The width is 60%, by default */
}

.right {
  background-color: #4CAF50;
  padding: 20px;
  float: left;
  width: 20%; /* The width is 20%, by default */
}

/* Use a media query to add a breakpoint at 800px: */
@media screen and (max-width: 800px) {
  .left, .main, .right {
    width: 100%; /* The width is 100%, when the viewport is 800px or smaller */
  }
}
```

```

</style>
</head>

<body>

<h2>Media Queries</h2>
<p>Resize the browser window.</p>
<p>Make sure you reach the breakpoint at 800px when resizing this frame.</p>

<div class="left">
  <p>Left Menu</p>
</div>

<div class="main">
  <p>Main Content</p>
</div>

<div class="right">
  <p>Right Content</p>
</div>

</body>

</html>

```

반응형 웹 페이지-전체 예

반응형 웹 페이지는 큰 데스크탑 화면과 작은 휴대폰에서 잘 보입니다.

```

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
* {
  box-sizing: border-box;
}

.menu {
  text-align: center;
  float: left;
  width: 20%;
}

.menu a {
  background-color: #e5e5e5;
  padding: 8px;
  margin-top: 7px;
  display: block;
  width: 100%;
  color: black;
}

```

```

.main {
  padding: 0 20px;
  float: left;
  width: 60%;
}

.right {
  background-color: #e5e5e5;
  padding: 15px;
  float: left;
  width: 20%;
  margin-top: 7px;
  text-align: center;
}

@media only screen and (max-width: 620px) {
  /* For mobile phones */
  .menu, .main, .right {
    width: 100%;
  }
}
</style>
</head>

<body style="font-family:verdana;color:#aaaaaa;">

<div style="background-color:#e5e5e5;padding:15px;text-align:center;">
  <h1>Hello World</h1>
</div>

<div style="overflow:auto;">
  <div class="menu">
    <a href="#">Link 1</a>
    <a href="#">Link 2</a>
    <a href="#">Link 3</a>
    <a href="#">Link 4</a>
  </div>

  <div class="main">
    <h2>Lorum Ipsum</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
  </div>

  <div class="right">
    <h2>About</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
</div>

<div style="background-color:#e5e5e5;padding:10px;text-align:center;margin-top:7px">© copyright w3schools.com</div>

</body>

</html>

```

반응형 웹 디자인-프레임 워크

반응형 디자인을 제공하는 기존 CSS 프레임 워크가 많이 있습니다.

그들은 무료이며 사용하기 쉽습니다.

W3.CSS 사용

반응형 디자인을 만드는 가장 좋은 방법은 [W3.CSS](https://www.w3schools.com/w3css/4/w3.css) 와 같은 반응형 스타일 시트를 사용하는 것 [입니다](#).

W3.CSS 를 사용하면 모든 규모의 멋진 사이트를 쉽게 개발할 수 있습니다. 데스크톱, 노트북, 태블릿 또는 전화 :

```
<!DOCTYPE html>
<html>
<title>W3.CSS</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<body>

<div class="w3-container w3-green">
  <h1>W3Schools Demo</h1>
  <p>Resize this responsive page!</p>
</div>

<div class="w3-row-padding">
  <div class="w3-third">
    <h2>London</h2>
    <p>London is the capital city of England.</p>
    <p>It is the most populous city in the United Kingdom,
    with a metropolitan area of over 13 million inhabitants.</p>
  </div>

  <div class="w3-third">
    <h2>Paris</h2>
    <p>Paris is the capital of France.</p>
    <p>The Paris area is one of the largest population centers in Europe,
    with more than 12 million inhabitants.</p>
  </div>

  <div class="w3-third">
    <h2>Tokyo</h2>
    <p>Tokyo is the capital of Japan.</p>
    <p>It is the center of the Greater Tokyo Area,
    and the most populous metropolitan area in the world.</p>
  </div>
</div>
```

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

부트 스트랩

인기있는 또 다른 프레임 워크는 Bootstrap 이며 HTML, CSS 및 jQuery 를 사용하여 반응 형 웹 페이지를 만듭니다.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>
</head>
<body>

<div class="container">
  <div class="jumbotron">
    <h1>My First Bootstrap Page</h1>
    <p>Resize this responsive page to see the effect!</p>
  </div>
  <div class="row">
    <div class="col-sm-4">
      <h3>Column 1</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</p>
      <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...</p>
    </div>
    <div class="col-sm-4">
      <h3>Column 2</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</p>
      <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...</p>
    </div>
    <div class="col-sm-4">
      <h3>Column 3</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</p>
      <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris...</p>
    </div>
  </div>
</div>

</body>
</html>
```


HTML 컴퓨터 코드 요소

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

<h2>Computer Code</h2>
<p>Some programming code:</p>

<code>
x = 5;<br>
y = 6;<br>
z = x + y;
</code>

</body>
</html>
```

키보드 입력 용 HTML <kbd>

HTML <kbd> 요소는 키보드 입력 또는 음성 명령과 같은 사용자 입력을 나타냅니다.

<kbd> 태그로 둘러싸인 텍스트는 일반적으로 모노 스페이스 글꼴로 표시됩니다.

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

<h2>The kbd Element</h2>
<p>The kbd element represents user input:</p>

<p>Save the document by pressing <kbd>Ctrl + S</kbd></p>

</body>
</html>
```

프로그램 출력용 HTML <samp>

HTML <samp> 요소는 프로그램 또는 컴퓨팅 시스템의 출력을 나타냅니다.

<samp> 태그로 둘러싸인 텍스트는 일반적으로 모노 스페이스 글꼴로 표시됩니다.

```
<!DOCTYPE html>
```

```
<html>
<body>

<h2>The samp Element</h2>
<p>The samp element represents output from a program or computing system:</p>

<p>If you input wrong value, the program will return <samp>Error!</samp></p>

</body>
</html>
```

컴퓨터 코드 용 HTML <code>

HTML <code>요소는 컴퓨터 코드 조각을 정의합니다.

<code>태그로 둘러싸인 텍스트는 일반적으로 모노 스페이스 글꼴로 표시됩니다.

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

<h2>The code Element</h2>
<p>Programming code example:</p>

<code>
x = 5;
y = 6;
z = x + y;
</code>

</body>
</html>
```

<code>요소가 여분의 공백과 줄 바꿈을 보존하지 않습니다.

이 문제를 해결하기 위해 <code>요소를 <pre>요소 안에 넣을 수 있습니다.

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

<p>The code element does not preserve whitespace and line-breaks.</p>
<p>To fix this, you can put the code element inside a pre element:</p>

<pre>
<code>
x = 5;
y = 6;
z = x + y;
</code>
</pre>
```

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

변수의 HTML <var>

HTML <var> 요소는 변수를 정의합니다.

변수는 수학 표현식의 변수이거나 프로그래밍 컨텍스트의 변수 일 수 있습니다.

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

<p>The var Element</p>
<p>Einstein wrote: <var>E</var> = <var>mc</var><sup>2</sup>.</p>

</body>
</html>
```

HTML 컴퓨터 코드 요소

Tag	Description
<u><code></u>	Defines programming code
<u><kbd></u>	Defines keyboard input
<u><samp></u>	Defines computer output
<u><var></u>	Defines a variable
<u><pre></u>	Defines preformatted text

HTML 엔티티

HTML 의 예약 문자는 문자 엔티티로 바뀌어야 합니다.

키보드에 없는 문자도 엔티티로 대체 할 수 있습니다.

HTML 엔티티

일부 문자는 HTML 로 예약되어 있습니다.

텍스트에보다 작음 (<) 또는 보다 큼 (>) 기호를 사용하면 브라우저에서 태그와 혼합 할 수 있습니다.

문자 엔티티는 예약 된 문자를 HTML 로 표시하는 데 사용됩니다.

캐릭터 엔티티는 다음과 같습니다 :

`&entity_name;`

OR

`&#entity_number;`

보다 작은 부호 (<)를 표시하려면 다음 과 같이 작성해야 합니다. **<** 또는 **<**;

엔터티 이름 사용의 장점 : 엔터티 이름은 기억하기 쉽습니다.

엔터티 이름 사용의 단점 : 브라우저는 모든 엔터티 이름을 지원하지는 않지만 숫자를 지원하는 것은 좋습니다.

비파괴 공간(Non-breaking Space)

HTML 에 사용되는 일반적인 문자 엔티티는 비 공백 공간입니다. ** **;

비 공백 공간은 줄 바꿈이 되지 않는 공간입니다.

줄 바꿈 하지 않은 공백으로 구분 된 두 단어는 서로 붙어 있습니다 (새 줄로 나뉘지 않음). 단어를 깨뜨리는 것이 방해가 될 수 있는 경우에 편리합니다.

예 :

- § 10
- 10km / h
- 오후 10 시

비 공백 공간의 또 다른 일반적인 용도는 브라우저가 HTML 페이지에서 공백을 자르지 못하게 하는 것입니다.

텍스트에 10 칸을 쓰면 브라우저가 9 칸을 제거합니다. 텍스트에 실제 공백을 추가하려면 ** ** 캐릭터 엔티티.

끊기지 않는 하이픈 ([‑](#))을 사용하면 끊지 않는 하이픈 문자 (-)를 사용할 수 있습니다.

다른 유용한 HTML 문자 엔티티

Result	Description	Entity Name	Entity Number
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark (apostrophe)	'	'
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®

참고 : 엔티티 이름은 대소 문자를 구분합니다.

분음 부호 결합

분음 부호는 문자에 추가 된 "글리프"입니다.

무덤 (̀) 및 급성 (́)과 같은 일부 발음 구별 부호를 악센트라고 합니다.

분음 부호는 글자 위와 아래, 글자 내부 및 두 글자 사이에 나타날 수 있습니다.

분음 부호는 영숫자 문자와 함께 사용되어 페이지에 사용 된 문자 세트 (인코딩)에 없는 문자를 생성 할 수 있습니다.

여기 몇 가지 예가 있어요.

Mark	Character	Construct	Result
`	a	à	à
'	a	á	á
^	a	â	â

~	a	ã	ã
`	o	Ò	ô
'	o	Ó	ó
^	o	Ô	ö
~	o	Õ	õ

HTML 기호

HTML 심볼 엔티티

HTML 엔터티는 이전 장에서 설명했습니다.

많은 수학, 기술 및 통화 기호는 일반 키보드에는 없습니다.

이러한 기호를 HTML 페이지에 추가하기 위해 HTML 엔티티 이름을 사용할 수 있습니다.

엔터티 이름이 없으면 엔터티 번호, 10 진수 또는 16 진수 참조를 사용할 수 있습니다.

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

<p>I will display &euro;</p>
<p>I will display &#8364;</p>
<p>I will display &#x20AC;</p>

</body>
</html>
```

HTML에서 지원하는 일부 수학 기호

Char	Number	Entity	Description
∀	∀	∀	FOR ALL
∂	∂	∂	PARTIAL DIFFERENTIAL
∃	∃	∃	THERE EXISTS
∅	∅	∅	EMPTY SETS
∇	∇	∇	NABLA
∈	∈	∈	ELEMENT OF
∉	∉	∉	NOT AN ELEMENT OF

⊃	∋	∋	CONTAINS AS MEMBER
∏	∏	∏	N-ARY PRODUCT
∑	∑	∑	N-ARY SUMMATION

[전체 수학 참조](#)

HTML이 지원하는 일부 그리스 문자

Char	Number	Entity	Description
Α	Α	Α	GREEK CAPITAL LETTER ALPHA
Β	Β	Β	GREEK CAPITAL LETTER BETA
Γ	Γ	Γ	GREEK CAPITAL LETTER GAMMA
Δ	Δ	Δ	GREEK CAPITAL LETTER DELTA
Ε	Ε	Ε	GREEK CAPITAL LETTER EPSILON
Ζ	Ζ	Ζ	GREEK CAPITAL LETTER ZETA

[완전한 그리스어 참조](#)

HTML이 지원하는 다른 엔티티

Char	Number	Entity	Description
©	©	©	COPYRIGHT SIGN
®	®	®	REGISTERED SIGN
€	€	€	EURO SIGN
™	™	™	TRADEMARK
←	←	←	LEFTWARDS ARROW
↑	↑	↑	UPWARDS ARROW
→	→	→	RIGHTWARDS ARROW
↓	↓	↓	DOWNWARDS ARROW
♠	♠	♠	BLACK SPADE SUIT
♣	♣	♣	BLACK CLUB SUIT
♥	♥	♥	BLACK HEART SUIT
♦	♦	♦	BLACK DIAMOND SUIT

[전체 통화 참조](#)

[전체 화살표 참조](#)

[전체 기호 참조](#)

HTML 인코딩 (문자 세트)

문자 인코딩이란 무엇입니까?

ASCII 는 첫 번째 문자 인코딩 표준 (문자 집합이라고도 함)이었습니다. ASCII 는 인터넷에서 사용할 수 있는 128 개의 영숫자 (숫자 (0-9), 영문자 (AZ) 및 ! \$ +-() @ <> .

HTML 4 의 기본 문자 집합은 ISO-8859-1 입니다. 이 문자 집합은 256 개의 다른 문자 코드를 지원했습니다.

ANSI (Windows-1252)는 원래 Windows 문자 세트입니다. ANSI 에 32 개의 추가 문자가 있다는 점을 제외하면 ANSI 는 ISO-8859-1 과 동일합니다.

ANSI 와 ISO-8859-1 은 매우 제한적이기 때문에 HTML 4 는 UTF-8 도 지원했습니다.

UTF-8 (유니 코드)은 세계의 거의 모든 문자와 기호를 다룹니다.

HTML5 의 기본 문자 집합은 UTF-8 입니다.

HTML 문자셋 속성

HTML 페이지를 올바르게 표시하려면 웹 브라우저가 페이지에 사용 된 문자 세트를 알아야 합니다.

`<meta>` 태그에 지정되어 있습니다 :

```
<meta charset="UTF-8">
```

브라우저가 웹 페이지에서 ISO-8859-1을 감지하면 기본값은 ANSI입니다.

ASCII 문자 세트

ASCII 는 제어 문자에 0 에서 31 (및 127) 사이의 값을 사용합니다.

ASCII 는 문자, 숫자 및 기호에 32 에서 126 사이의 값을 사용합니다.

ASCII 는 128 에서 255 사이의 값을 사용하지 않습니다.

ANSI 문자 집합 (Windows-1252)

ANSI 는 0 에서 127 사이의 값에 대해 ASCII 와 동일합니다.

ANSI 에는 128 에서 159 사이의 값에 대한 독점 문자 세트가 있습니다.

ANSI 는 160 에서 255 사이의 값에 대해 UTF-8 과 동일합니다.

ISO-8859-1 문자 세트

8859-1 은 0 에서 127 사이의 값에 대해 ASCII 와 동일합니다.

8859-1 은 128 에서 159 사이의 값을 사용하지 않습니다.

8859-1 은 160 에서 255 사이의 값에 대해 UTF-8 과 동일합니다.

UTF-8 문자 세트

UTF-8 은 0 에서 127 사이의 값에 대해 ASCII 와 동일합니다.

UTF-8 은 128 에서 159 사이의 값을 사용하지 않습니다.

UTF-8 은 160 에서 255 사이의 값에 대해 ANSI 및 8859-1 과 동일합니다.

UTF-8 은 256 개 이상의 문자로 10,000 개 이상의 문자를 계속합니다.

자세히 살펴 보려면 [완전한 HTML 문자 세트 참조를](#) 연구 [하십시오](#) .

@charset CSS 규칙

CSS **@charset** 규칙을 사용하여 스타일 시트에 사용되는 문자 인코딩을 지정할 수 있습니다.

스타일 시트의 인코딩을 유니 코드 UTF-8로 설정하십시오.

```
@charset "UTF-8";
```

HTML 유니폼 리소스 로케이터

URL 은 웹 주소의 다른 단어입니다.

URL 은 단어 (w3schools.com) 또는 인터넷 프로토콜 (IP) 주소 (192.68.20.50)로 구성 될 수 있습니다.

대부분의 사람들은 이름을 숫자보다 기억하기 쉽기 때문에 서핑 할 때 이름을 입력합니다.

URL-Uniform Resource Locator

웹 브라우저는 URL 을 사용하여 웹 서버에서 페이지를 요청합니다.

URL (Uniform Resource Locator)은 웹에서 문서 (또는 다른 데이터)를 처리하는데 사용됩니다.

<https://www.w3schools.com/html/default.asp> 와 같은 웹 주소는 다음 구문 규칙을 따릅니다.

<code>scheme://prefix.domain:port/path/filename</code>
--

설명:

- **scheme-** 인터넷 서비스 유형 을 정의합니다 (가장 일반적으로 **http** 또는 **https** 입니다)
- **prefix-** 도메인 접두사를 정의 합니다 (http 의 기본값은 **www** 입니다).
- **domain** - 인터넷 정의 도메인 이름 (w3schools.com 등)
- **port-** 호스트 의 포트 번호 를 정의합니다 (http 의 기본값은 **80** 임)
- **path-** 서버 의 경로 를 정의 합니다 (생략 된 경우 : 사이트의 루트 디렉토리)
- **filename-** 문서 또는 리소스의 이름을 정의합니다

일반적인 URL 체계

아래 표에는 몇 가지 일반적인 구성표가 나와 있습니다.

Scheme	Short for	Used for
http	HyperText Transfer Protocol	Common web pages. Not encrypted
https	Secure HyperText Transfer Protocol	Secure web pages. Encrypted
ftp	File Transfer Protocol	Downloading or uploading files
file		A file on your computer

URL 인코딩

URL 은 [ASCII 문자 집합](#) 을 사용하여 인터넷을 통해서만 보낼 수 있습니다. URL 에 ASCII 집합 외부의 문자가 포함되어 있으면 URL 을 변환해야 합니다.

URL 인코딩은 비 ASCII 문자를 인터넷을 통해 전송할 수 있는 형식으로 변환합니다.

URL 인코딩은 ASCII 가 아닌 문자를 "%"로 대체 한 다음 16 진수입니다.

URL 은 공백을 포함 할 수 없습니다. URL 인코딩은 일반적으로 공백을 더하기 (+) 기호 또는 %20 으로 바꿉니다.

직접 해보기

제출

"제출"을 클릭하면 브라우저가 입력을 서버로 보내기 전에 URL 을 인코딩 합니다.

서버의 페이지에 수신 된 입력이 표시됩니다.

다른 입력을 시도하고 제출을 다시 클릭하십시오.

ASCII 인코딩 예

브라우저는 페이지에서 사용 된 문자 집합에 따라 입력을 인코딩 합니다.

HTML5 의 기본 문자 집합은 UTF-8 입니다.

Character	From Windows-1252	From UTF-8
€	%80	%E2%82%AC
£	%A3	%C2%A3
©	%A9	%C2%A9
®	%AE	%C2%AE
À	%C0	%C3%80
Á	%C1	%C3%81
Â	%C2	%C3%82
Ã	%C3	%C3%83
Ä	%C4	%C3%84
Å	%C5	%C3%85

모든 URL 인코딩에 대한 자세한 내용은 [URL 인코딩 참조](#)를 방문하십시오.

HTML 과 XHTML

XHTML은 XML로 작성된 HTML입니다.

XHTML이란 무엇입니까?

- XHTML 은 약자 **EX**tensible **H**yper **TEXT** **M**arkup **L**anguage
- XHTML 은 HTML 과 거의 동일
- XHTML 은 HTML 보다 엄격합니다.
- XHTML 은 XML 애플리케이션으로 정의 된 HTML 입니다.
- XHTML 은 모든 주요 브라우저에서 지원됩니다

왜 XHTML인가?

인터넷의 많은 페이지에는 "잘못된"HTML 이 포함되어 있습니다.

이 HTML 코드는 HTML 규칙을 따르지 않더라도 대부분의 브라우저에서 제대로 작동합니다.

```
<html>
<head>
  <title>This is bad HTML</title>

<body>
  <h1>Bad HTML
  <p>This is a paragraph
</body>
```

오늘날의 시장은 서로 다른 브라우저 기술로 구성됩니다. 일부 브라우저는 컴퓨터에서 실행되고 일부 브라우저는 휴대폰 또는 기타 소형 장치에서 실행됩니다. 작은 장치에는 종종 "나쁜"마크 업을 해석 할 수 있는 리소스 나 능력이 부족합니다.

XML 은 문서를 올바르게 마크 업해야 하는 마크 업 언어입니다 ("잘 구성된"것).

XML 을 공부하고 싶다면 [XML 튜토리얼](#)을 읽으십시오.

XHTML 은 HTML 과 XML 의 장점을 결합하여 개발되었습니다.

XHTML 은 XML 로 재 설계된 HTML 입니다.

HTML과의 가장 중요한 차이점 :

문서 구조

- XHTML DOCTYPE 은 필수입니다
- <html>의 xmlns 속성은 필수입니다
- <html>, <head>, <title> 및 <body>는 필수입니다

XHTML 요소

- XHTML 요소는 올바르게 중첩 되어야 합니다
- XHTML 요소는 항상 닫아야 합니다
- XHTML 요소는 소문자 여야 합니다
- XHTML 문서에는 하나의 루트 요소 가 있어야 합니다

XHTML 속성

- 속성 이름은 소문자 여야 합니다
- 속성 값은 인용 해야 합니다
- 속성 최소화는 금지되어 있습니다

<! DOCTYPE> 필수입니다

XHTML 문서에는 XHTML DOCTYPE 선언이 있어야 합니다.

모든 [XHTML 독 타입](#) 의 전체 목록은 HTML 태그 참조에 있습니다.

<html>, <head>, <title> 및 <body> 요소도 있어야 하며 <html>의 xmlns 속성은 문서의 xml 네임 스페이스를 지정해야 합니다.

이 예제는 최소 필수 태그가 있는 XHTML 문서를 보여줍니다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <title>Title of document</title>
</head>

<body>
  some content
</body>

</html>
```

XHTML 요소는 올바르게 중첩되어야 합니다

HTML에서 다음과 같이 일부 요소가 서로 잘못 중첩 될 수 있습니다.

```
<b><i>This text is bold and italic</b></i>
```

XHTML에서 모든 요소는 다음과 같이 서로 올바르게 중첩되어야합니다.

```
<b><i>This text is bold and italic</i></b>
```

XHTML 요소는 항상 닫아야 합니다

이것은 틀렸다 :

```
<p>This is a paragraph
<p>This is another paragraph
```

이것은 맞습니다

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

빈 요소도 닫아야 합니다

이것은 틀렸다 :

```
A break: <br>
A horizontal rule: <hr>
An image: 
```

이것은 맞습니다 :

```
A break: <br />
A horizontal rule: <hr />
An image: 
```

XHTML 요소는 소문자 여야 합니다

이것은 틀렸다

```
<BODY>
<P>This is a paragraph</P>
</BODY>
```

이것은 맞습니다 :

```
<body>
<p>This is a paragraph</p>
</body>
```

XHTML 속성 이름은 소문자 여야 합니다

이것은 틀렸다 :

```
<table WIDTH="100%">
```

이것은 맞습니다 :

```
<table width="100%">
```


속성 값을 인용해야 합니다

이것은 틀렸다 :

```
<table width=100%>
```

이것은 맞습니다 :

```
<table width="100%">
```

속성 최소화가 금지됩니다

잘못된:

```
<input type="checkbox" name="vehicle" value="car" checked />
```

옳은:

```
<input type="checkbox" name="vehicle" value="car" checked="checked" />
```

잘못된:

```
<input type="text" name="lastname" disabled />
```

옳은:

```
<input type="text" name="lastname" disabled="disabled" />
```

HTML에서 XHTML로 변환하는 방법

1. 모든 페이지의 첫 줄에 XHTML <!DOCTYPE> 추가
2. 모든 페이지의 html 요소에 xmlns 속성 추가
3. 모든 요소 이름을 소문자로 변경
4. 빈 요소를 모두 닫습니다
5. 모든 속성 이름을 소문자로 변경
6. 모든 속성 값 인용

W3C Validator로 HTML 검증

아래 상자에 웹 주소를 입력하십시오.