

# 221128\_2반\_실습

## HTML 시작하기

### Tools

#### Text Editor

- 코드 작성을 위한 **텍스트 편집기**
- 텍스트 편집기
  - Notepad(Windows)
  - TextEdit(macOS)
  - GNOME Text Editor(Ubuntu 22.04 LTS release)
  - Sublime
  - Brackets
  - Atom
  - Visual Studio Code
  - Notepad++
  - GNU Emacs
  - VIM
- 하이브리드 텍스트 편집기
  - Dreamweaver
  - WebStorm

#### Web Browser

- 코드를 테스트하기 위한 **웹 브라우저**
  - **Linux:** Firefox, Chrome, Opera, Brave.

- **Windows:** Firefox, Chrome, Opera, Microsoft Edge, Brave (Windows 10 comes with Edge by default; otherwise, you should install an alternative browser).
- **macOS:** Firefox, Chrome, Opera, Safari, Brave (macOS and iOS come with Safari by default).

## HTML이란?

### HTML(Hypertext Markup Language,하이퍼텍스트 마크업 언어)

- 프로그래밍 언어는 아니고, 우리가 보는 웹페이지가 어떻게 구조화되어 있는지 브라우저로 하여금 알 수 있도록 하는 **마크업 언어**(**markup language**)입니다.
- markup language : 태그 등을 이용하여 문서 혹은 데이터의 구조를 기록하는 언어
- HTML은 **elements** 로 구성되어 있으며, 이들은 적절한 방법으로 나타내고 실행하기 위해 각 콘텐츠의 여러 부분들을 감싸고 마크업 합니다.
- **tags** 는 웹 상의 다른 페이지로 이동하게 하는 하이퍼링크 내용들을 생성하거나, 단어를 강조하는 등의 역할을 합니다.

### 예시

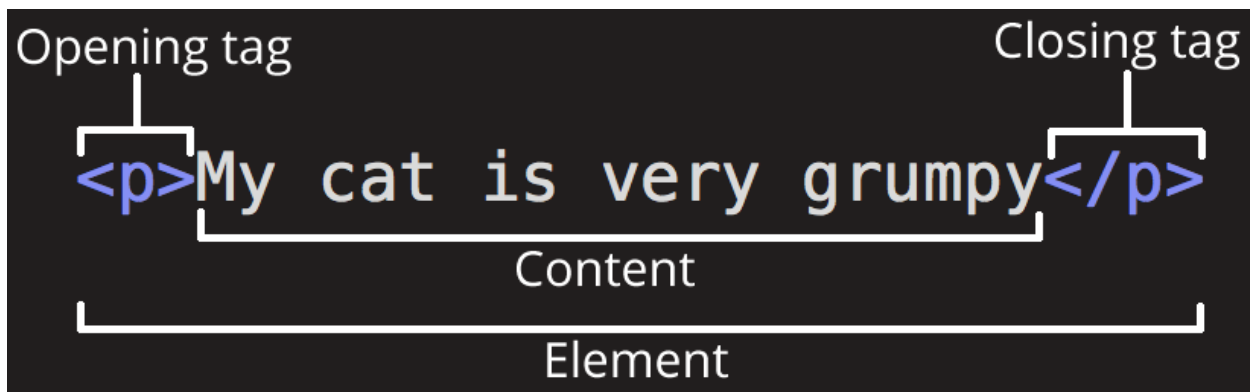
```
My cat is very grumpy
```

- 만약 다음의 문장을 그냥 그 자체로 표시하고 싶다면, 태그 중 (**<p>**)로 감싸 엘리먼트를 문단으로 명시할 수 있습니다.

```
<p>My cat is very grumpy</p>
```

## HTML Element의 구조

### Element 구조



HTML에서 Element의 구조

1. **여는 태그(Opening tag)**: 이것은 요소의 이름과(이 경우 p), 열고 닫는 꺾쇠 괄호로 구성됩니다. 요소가 시작(이 경우 단락의 시작 부분)부터 효과가 적용되기 시작합니다.
2. **닫는 태그(Closing tag)**: 이것은 요소(element)의 이름 앞에 슬래시(/)가 있는것을 제외하면 여는 태그(opening tag)와 같습니다. 이것은 요소의 끝(이 경우 단락의 끝 부분)에 위치합니다. 닫는 태그를 적어주지 않는 것은 흔한 초심자의 오류이며, 이것은 이상한 결과를 낳게됩니다.
3. **내용(Content)**: 요소의 내용이며, 이 경우 단순한 텍스트이다.
4. **요소(Element)**: 여는 태그, 닫는 태그, 내용을 통틀어 요소(element)라고한다.



HTML `element` 에 관한 참고 사이트

<https://developer.mozilla.org/ko/docs/Web/HTML/Element/a>

## <html>: HTML 문서 / 루트 요소

HTML `<html>` 요소는 HTML 문서의 루트(최상단 요소)를 나타내며, "루트 요소"라고도 부릅니다. 모든 다른 요소는 `<html>` 요소의 후손이어야 합니다.

## <head>: 문서 메타데이터 (헤더) 요소

HTML `<head>` 요소는 기계가 식별할 수 있는 문서 정보(메타데이터)를 담습니다. 정보로는 문서가 사용할 제목, 스크립트, 스타일 시트 등이 있습니다.

## <body>: 문서 본문 요소

HTML `<body>` 요소는 HTML 문서의 내용을 나타냅니다. 한 문서에 하나의 `<body>` 요소만 존재할 수 있습니다.

## <h1>–<h6>: HTML 구획 제목 요소

HTML `<h1>` – `<h6>` 요소는 6단계의 구획 제목을 나타냅니다. 구획 단계는 `<h1>` 이 가장 높고 `<h6>` 은 가장 낮습니다.

- 특성
  - `<h1>` - `<h6>` 요소는 전역 특성만 포함합니다.
  - **전역 특성(Global attributes)** : 모든 HTML에서 공통으로 사용할 수 있는 특성입니다. 그러나 일부 요소에는 아무런 효과도 없을 수 있습니다.

### 예제

#### 1. 모든 제목 단계

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
<h3>Heading level 3</h3>
<h4>Heading level 4</h4>
<h5>Heading level 5</h5>
<h6>Heading level 6</h6>
```

#### 2. 예제 페이지

```
<h1>Heading elements</h1>
<h2>Summary</h2>
<p>Some text here...</p>

<h2>Examples</h2>
<h3>Example 1</h3>
<p>Some text here...</p>

<h3>Example 2</h3>
<p>Some text here...</p>

<h2>See also</h2>
<p>Some text here...</p>
```

#### 3. 탐색

- 스크린 리더 사용자가 흔히 사용하는 탐색 기법은 제목에서 제목으로 뛰어넘으며 페이지 콘텐츠를 빠르게 파악하는 것입니다. 따라서 제목 단계를 뛰어넘으면 중간에 빠진 제목이 어디 있는 건지 모르므로 사용자의 혼란을 야기할 수 있습니다.
- 나쁜 예

```
<h1>Heading level 1</h1>
<h3>Heading level 3</h3>
<h4>Heading level 4</h4>
```

- 좋은 예

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
<h3>Heading level 3</h3>
```

#### 4. 중첩

- 하위 구획으로 제목을 중첩해 콘텐츠의 구조를 나타낼 수 있습니다. 대부분의 스크린 리더는 페이지의 제목을 정렬한 목록도 생성해 제공하므로, 사용자가 빠르게 콘텐츠 구조를 파악할 수 있도록 도와줍니다.

## <a>

**HTML** **<a>** 요소(앵커 요소)는 **href** 특성을 통해 다른 페이지나 같은 페이지의 어느 위치, 파일, 이메일 주소와 그 외 다른 URL로 연결할 수 있는 하이퍼링크를 만듭니다. **<a>** 안의 콘텐츠는 링크 목적지의 설명을 나타내야 합니다.

## 예제

### 1. URL로 연결

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

### 2. 이메일 주소로 연결

- 사용자의 이메일 프로그램을 통해 새로운 메일을 보낼 수 있는 링크를 생성하려면 **mailto:**

스킴을 사용하세요.

```
<a href="mailto:nowhere@mozilla.org">Send email to nowhere</a>
```

### 3. 전화번호로 연결

```
<a href="tel:+49.157.0156">+49 157 0156</a>  
<a href="tel:+1(555)5309">(555) 5309</a>
```

- **tel:** 링크는 장치의 능력에 따라 행동이 바뀝니다.
  - 휴대전화에서는 번호를 자동으로 입력합니다.
  - 대부분의 운영체제에는 Skype, FaceTime처럼 전화를 걸 수 있는 프로그램이 있습니다.
  - 웹사이트는 `Navigator.registerProtocolHandler()`를 사용해 전화를 걸 수 있습니다. `web.skype.com`을 참고하세요.
  - 다른 행동으로는 연락처에 저장하기와 다른 장치로 전송 등이 있습니다.

## <br>: 줄바꿈 요소

**HTML <br> 요소**는 텍스트 안에 줄바꿈(캐리지 리턴)을 생성합니다. 주소나 시조 등 줄의 구분이 중요한 내용을 작성할 때 유용합니다.

### 예제

```
Mozilla Foundation<br>  
1981 Landings Drive<br>  
Building K<br>  
Mountain View, CA 94043-0801<br>  
USA
```

## <table>

**HTML <table> 요소**는 행과 열로 이루어진 표를 나타냅니다.

### 예제

#### 1. 연습

```

<table>
  <thead>
    <tr>
      <th colspan="2">The table header</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>The table body</td>
      <td>with two columns</td>
    </tr>
  </tbody>
</table>

```

```

table,
td {
  border: 1px solid #333;
}

thead,
tfoot {
  background-color: #333;
  color: #fff;
}

```

## 2. 간단한 예

```

<table>
  <tr>
    <td>John</td>
    <td>Doe</td>
  </tr>
  <tr>
    <td>Jane</td>
    <td>Doe</td>
  </tr>
</table>

```

## 3. 추가 예제

```

<p>Simple table with header</p>
<table>
  <tr>
    <th>First name</th>

```

```

        <th>Last name</th>
    </tr>
    <tr>
        <td>John</td>
        <td>Doe</td>
    </tr>
    <tr>
        <td>Jane</td>
        <td>Doe</td>
    </tr>
</table>

<p>Table with thead, tfoot, and tbody</p>
<table>
    <thead>
        <tr>
            <th>Header content 1</th>
            <th>Header content 2</th>
        </tr>
    </thead>
    <tfoot>
        <tr>
            <td>Footer content 1</td>
            <td>Footer content 2</td>
        </tr>
    </tfoot>
    <tbody>
        <tr>
            <td>Body content 1</td>
            <td>Body content 2</td>
        </tr>
    </tbody>
</table>

<p>Table with colgroup</p>
<table>
    <colgroup span="4" class="columns"></colgroup>
    <tr>
        <th>Countries</th>
        <th>Capitals</th>
        <th>Population</th>
        <th>Language</th>
    </tr>
    <tr>
        <td>USA</td>
        <td>Washington D.C.</td>
        <td>309 million</td>
        <td>English</td>
    </tr>
    <tr>
        <td>Sweden</td>
        <td>Stockholm</td>
        <td>9 million</td>
        <td>Swedish</td>
    </tr>

```



```

    </tr>
</table>

<p>Table with colgroup and col</p>
<table>
  <colgroup>
    <col class="column1">
    <col class="columns2plus3" span="2">
  </colgroup>
  <tr>
    <th>Lime</th>
    <th>Lemon</th>
    <th>Orange</th>
  </tr>
  <tr>
    <td>Green</td>
    <td>Yellow</td>
    <td>Orange</td>
  </tr>
</table>

<p>Simple table with caption</p>
<table>
  <caption>Awesome caption</caption>
  <tr>
    <td>Awesome data</td>
  </tr>
</table>

```

## <col>

**HTML** `<col>` 요소는 표의 열을 나타내며, 열에 속하는 칸에 공통된 의미를 부여할 때 사용합니다.

### 예제

```

<table>
  <caption>Superheros and sidekicks</caption>
  <colgroup>
    <col>
    <col span="2" class="batman">
    <col span="2" class="flash">
  </colgroup>
  <tr>
    <td> </td>
    <th scope="col">Batman</th>
    <th scope="col">Robin</th>
    <th scope="col">The Flash</th>
    <th scope="col">Kid Flash</th>
  </tr>
</table>

```

```

    </tr>
    <tr>
      <th scope="row">Skill</th>
      <td>Smarts</td>
      <td>Dex, acrobat</td>
      <td>Super speed</td>
      <td>Super speed</td>
    </tr>
  </table>

```

```

.batman {
  background-color: #d7d9f2;
}

.flash {
  background-color: #ffe8d4;
}

caption {
  padding: 8px;
  caption-side: bottom;
}

table {
  border-collapse: collapse;
  border: 2px solid rgb(100, 100, 100);
  letter-spacing: 1px;
  font-family: sans-serif;
  font-size: .7rem;
}

td,
th {
  border: 1px solid rgb(100, 100, 100);
  padding: 10px 10px;
}

td {
  text-align: center;
}

```

## 속성(Attributes)

속성은 요소에 실제로 나타내고 싶지 않지만 추가적인 내용을 담고 싶을 때 사용합니다. 위에는 나중에 스타일에 관련된 내용이나 기타 내용을 위해 해당 목표를 구분할 수 있는 `class` 속성을 부여했습니다.

```
<p class="editor-note">My cat is very grumpy</p>
```

속성을 사용할 때에는 아래 내용을 지켜야 합니다:

1. 요소 이름 다음에 바로 오는 속성은 요소 이름과 속성 사이에 공백이 있어야 되고, 하나 이상의 속성들이 있는 경우엔 속성 사이에 공백이 있어야 합니다.
2. 속성 이름 다음엔 등호(=)가 붙습니다.
3. 속성 값은 열고 닫는 따옴표로 감싸야 합니다.

## class

### class

전역 특성은 공백으로 구분한 요소 클래스의 목록으로, 대소문자를 구분하지 않습니다. 클래스는 CSS나 JavaScript에서 클래스 선택자나 DOM 메서드의 `document.getElementsByClassName().(en-US)`과 같은 메서드를 통해 요소에 접근할 수 있는 방법입니다.

### 예제

```
<p>Narrator: This is the beginning of the play.</p>

<p class="note editorial">Above point sounds a bit obvious. Remove/rewrite?</p>

<p>Narrator: I must warn you now folks that this beginning is very exciting.</p>

<p class="note">[Lights go up and wind blows; Caspian enters stage right]</p>
```

```
.note {
  font-style: italic;
  font-weight: bold;
}

.editorial {
  background: rgb(255, 0, 0, .25);
  padding: 10px;
}

.editorial:before {
  content: 'Editor: ';
}
```

## id

### id

전역 특성은 문서 전체에서 유일한 고유식별자(ID)를 정의합니다. 고유식별자의 목적은 프래그먼트 식별자를 사용해 요소를 가리킬 때와 스크립트 및 스타일 적용 시 특정 요소를 식별하기 위함입니다.

### 예제

```
<p>A normal, boring paragraph. Try not to fall asleep.</p>

<p id="exciting">The most exciting paragraph on the page. One of a kind!</p>
```

```
#exciting {
  background: linear-gradient(to bottom, #ffe8d4, #f69d3c);
  border: 1px solid #696969;
  padding: 10px;
  border-radius: 10px;
  box-shadow: 2px 2px 1px black;
}

#exciting:before {
  content: "i";
  margin-right: 5px;
}
```

## style

style 전역 특성은 요소에 적용할 CSS 스타일 선언을 담습니다. 스타일은 별도의 파일에 정의하는 것이 권장된다는 점을 참고하세요. 이 특성과 <style> 요소는 주로 테스트 등 빠른 스타일링을 위한 목적으로 사용됩니다.

### 예제

```
<div style="background: #ffe7e8; border: 2px solid #e66465;">
  <p style="margin: 15px; line-height: 1.5; text-align: center;">
    Well, I am the slime from your video<br>
    Oozin' along on your livin' room floor.</p>
</div>
```

## title

**title** 전역 특성은 요소와 관련된 추가 정보를 제공하는 텍스트를 나타냅니다.

## 예제

```
<p>Use the <code>title</code> attribute on an <code>iframe</code> to clearly identify the content of the <code>iframe</code> to screen readers.</p>
```

```
<iframe title="Wikipedia page for the HTML language"
  src="https://en.m.wikipedia.org/wiki/HTML"></iframe>
<iframe title="Wikipedia page for the CSS language"
  src="https://en.m.wikipedia.org/wiki/CSS"></iframe>
```

```
iframe {
  height: 200px;
  margin-bottom: 24px;
  width: 100%;
}
```

# CSS

## CSS기초

### css란?

- CSS (Cascading Style Sheets)는 웹페이지를 꾸미려고 작성하는 코드입니다.
- HTML과 같이 CSS는 실제로 프로그래밍 언어는 아닙니다. *마크업(markup) 언어*도 아닙니다. **Style sheet 언어**입니다.

```
p {
  color: red;
  width: 500px;
  border: 1px solid black;
}
```

## CSS의 ruleset 해부

전체 구조는 **rule set** 라 불립니다 (하지만 종종 줄여서 "rule"이라고 합니다). 각 부분의 이름에도 주목하세요:

## 1. 선택자(selector)

rule set의 맨 앞에 있는 HTML 요소 이름. 이것은 꾸밀 요소(들)을 선택합니다 (이 예에서는 p 요소). 다른 요소를 꾸미기 위해서는 선택자만 바꿔주세요.

## 2. 선언

`color: red` 와 같은 단일 규칙; 여러분이 꾸미기 원하는 요소의 속성을 명시합니다.

## 3. 속성(property)

주어진 HTML 요소를 꾸밀 수 있는 방법입니다. (이 예에서, `color` 는 p 요소의 속성입니다.) CSS에서, rule 내에서 영향을 줄 속성을 선택합니다.

## 4. 속성 값

속성의 오른쪽에, 콜론 뒤에, 주어진 속성을 위한 많은 가능한 결과중 하나를 선택하기 위해 속성 값을 갖습니다 (`color` 의 값에는 `red` 외에 많은 것이 있습니다).

구문의 다른 중요한 부분들도 주목하세요:

- 각각의 rule set (셀렉터로 구분) 은 반드시 (`{ }`) 로 감싸져야 합니다.
- 각각의 선언 안에, 각 속성을 해당 값과 구분하기 위해 콜론 (`:`)을 사용해야만 합니다.
- 각각의 rule set 안에, 각 선언을 그 다음 선언으로부터 구분하기 위해 세미콜론 (`;`)을 사용해야만 합니다.

```
p {  
  color: red;  
  width: 500px;  
  border: 1px solid black;  
}
```

## 여러 요소 선택하기

여러분은 요소의 여러 타입을 선택하고 모두에게 하나의 rule set 을 적용할 수도 있습니다. 여러 선택자는 콤마로 구분합니다.

```
p, li, h1 {  
  color: red;  
}
```

# 선택자의 여러 종류

## 요소선택자

선택자 이름	선택하는 것	예시
요소 선택자 (때때로 태그 또는 타입 선택자라 불림)	특정 타입의 모든 HTML 요소.	<code>p</code> <code>&lt;p&gt;</code> 를 선택
아이디 선택자	특정 아이디를 가진 페이지의 요소 (주어진 HTML 페이지에서, 아이디당 딱 하나의 요소만 허용됩니다).	<code>#my-id</code> <code>&lt;p id="my-id"&gt;</code> 또는 <code>&lt;a id="my-id"&gt;</code> 를 선택
클래스 선택자	특정 클래스를 가진 페이지의 요소 (한 페이지에 클래스가 여러번 나타날 수 있습니다).	<code>.my-class</code> <code>&lt;p class="my-class"&gt;</code> 와 <code>&lt;a class="my-class"&gt;</code> 를 선택
속성 선택자	특정 속성을 갖는 페이지의 요소.	<code>img[src]</code> <code>&lt;img src="myimage.png"&gt;</code> 를 선택하지만 <code>&lt;img&gt;</code> 는 선택 안함
수도(Pseudo) 클래스 선택자	특정 요소이지만 특정 상태에 있을 때만, 예를 들면, hover over 상태일 때.	<code>a:hover</code> <code>&lt;a&gt;</code> 를 선택하지만, 마우스 포인터가 링크위에 있을 때만 선택함

# HTML 에 CSS 적용하기

## 외부 스타일 시트

외부 스타일 시트는 CSS 확장자가 `.css` 인 별도의 파일로 작성되고, HTML `<link>` 요소에서 참조하는 경우입니다:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>나의 CSS 실험</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>헬로우 월드!</h1>
    <p>이것은 나의 첫 번째 CSS 예제입니다</p>
```

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

```
h1 {
  color: blue;
  background-color: yellow;
  border: 1px solid black;
}

p {
  color: red;
}
```

`<link>` 요소의 `href` 속성은 여러분의 파일 시스템 파일을 참조해야 합니다.

위의 예에서, CSS 파일은 HTML 문서와 동일한 폴더에 있지만, 다른 곳에 저장 한다면, 지정된 경로를 다음과 같이 조정할 수 있습니다:

```
<!-- 현재 폴더의 styles 라는 하위 폴더 안에 -->
<link rel="stylesheet" href="styles/style.css">

<!-- 현재 폴더의 styles 라는 하위 폴더에 있는 general 이라는 하위 폴더 안에 -->
<link rel="stylesheet" href="styles/general/style.css">

<!-- 상위 폴더로 올라간 다음, styles 라는 하위 폴더 내로 이동 -->
<link rel="stylesheet" href="../styles/style.css">
```

## 내부 스타일 시트

내부 스타일 시트는 외부 CSS 파일이 없는 대신, HTML `<head>` 안에 포함된 `<style>` 요소 내부에 CSS 를 배치합니다.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>나의 CSS 실험</title>
    <style>
      h1 {
        color: blue;
        background-color: yellow;
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <h1>안녕하세요</h1>
  </body>
</html>
```



```

    p {
      color: red;
    }
  </style>
</head>
<body>
  <h1>헬로우 월드</h1>
  <p>이것은 나의 첫 번째 CSS 예제입니다</p>
</body>
</html>

```

이는 일부 상황 (CSS 파일을 직접 수정할 수 없는 콘텐츠 관리 시스템을 사용하는 경우도 있지만) 에서 유용할 수 있지만, CSS 가 필요한 외부 스타일 시트 만큼 효율적이지 않습니다 — 웹 사이트에서, CSS 가 모든 페이지에서 반복되고 변경이 필요한 경우 여러 위치에서 업데이트 됩니다.

## 인라인 스타일인라인 스타일

인라인 스타일은 `style` 속성 내에 포함된 한 요소에만 영향을 주는 CSS 선언입니다:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>나의 CSS 실험</title>
  </head>
  <body>
    <h1 style="color: blue;background-color: yellow;border: 1px solid black;">헬로우 월드!</h1>
    <p style="color:red;">이것은 나의 첫 번째 CSS 예제입니다</p>
  </body>
</html>

```

**당신이 정말로 필요하지 않는 한, 이것을 사용하지 마십시오!** 유지 관리가 실제로 좋지 않으며 (문서당 동일한 정보를 여러번 업데이트 해야할 수도 있음), 프리젠테이션 CSS 정보와 HTML 구조 정보를 혼합하여 코드를 읽고 이해하기 어렵게 만듭니다. 다른 유형의 코드를 분리하여 유지하면 코드 작업을 하는 모든 사람이 훨씬 쉽게 작업할 수 있습니다.

인라인 스타일이 더 일반적이거나 권장되는 곳이 몇 군데 있습니다. 작업 환경이 실제로 제한적인 경우 (CMS 로 HTML 본문만 편집할 수 있음), 이를 사용하는 것이 좋습니다. 또한 가능한 많은 전자 메일 클라이언트와 호환되도록 HTML 전자 메일에 많이 사용된 것을 볼 수 있습니다.

## float

## floats의 배경

float 속성은 웹 개발자가 텍스트 열 내부에 float하는 이미지를 포함하고, 아울러 해당 이미지의 좌측 우측 주변으로 텍스트를 둘러싸는 간단한 레이아웃을 구현할 수 있도록 도입되었습니다. 이런 것은 신문 레이아웃에서 볼 수 있는 종류입니다.

그러나 웹 개발자들은 이미지뿐만 아니라 무엇이든 float할 수 있음을 빠르게 깨달았고, 그래서 floats 사용이 확대되었습니다. 앞서 살펴본 고급 단락 예제는 재미있는 드롭캡 효과를 생성하는데 floats를 어떻게 사용할 수 있는지를 보여줍니다.

floats는 일반적으로 상대 요소와 나란히 놓이도록 float(浮動)하는 다단 정보를 갖춘 웹 사이트의 전체 레이아웃을 만들는데 널리 사용되어 왔다(기본 행동은 다단 무리가 소스에서 보이는 순서와 같은 순서대로 상대 요소 아래에 자리잡기하는 것이다). 더 새롭고 더 나은 레이아웃 기술이 나와있으므로 이러한 방식으로 floats를 사용하는 것은 낡은 기술로 간주되어야 합니다.

## 간단한 float 예제

```
<h1>간단한 float 예제</h1>
```

```
<div class="box">float</div>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. </p>
```

```
<p>Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.</p>
```

```
<p>Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.</p>
```

```
body {  
  width: 90%;  
  max-width: 900px;  
  margin: 0 auto;  
  font: .9em/1.2 Arial, Helvetica, sans-serif;  
}
```

```
.box {
  width: 150px;
  height: 100px;
  border-radius: 5px;
  background-color: rgb(207,232,220);
  padding: 1em;
}
```

지금 저장하고 새로 고침하면, 여러분이 기대한 것과 비슷한 것을 보게 될 것입니다. 다시말해 일반 대열에 속한 상자는 텍스트 위에 위치를 잡고 있습니다. 텍스트가 상자 주변에 float하려면 아래에서 보듯 `.box` 규칙에 두 가지 속성을 추가하십시오.

```
.box {
  float: left;
  margin-right: 15px;
  width: 150px;
  height: 100px;
  border-radius: 5px;
  background-color: rgb(207,232,220);
  padding: 1em;
}
```

## 예제2

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

<head>
  <meta charset="UTF-8">
  <title>CSS Float</title>
  <style>
    img {
      float: left;
      margin-right: 20px;
    }
  </style>
</head>

<body>

  <h1>float 속성을 이용한 위치 설정</h1>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam ornare sapien suscipit ti
ncidunt ullamcorper. Cras ac sem sed mauris maximus rhoncus vel in metus. Nam pharetra arc
u sit amet dolor interdum, eget scelerisque libero finibus. Phasellus quis vulputate ante.
Fusce sit amet viverra justo. Donec id elementum mauris. Nam id porttitor nisl, et suscipi
t nunc. Vestibulum sit amet volutpat quam. Cum sociis natoque penatibus et magnis dis part
```

```

urient montes, nascetur ridiculus mus. Duis placerat sem eu facilisis ultricies.
    Nulla et elit placerat, malesuada massa sit amet, dictum quam. Duis eu turpis non sapi
    en molestie luctus nec a tortor. Nunc non est eget elit suscipit interdum. Maecenas iaculi
    s lobortis lorem et dignissim. Nullam dapibus, arcu vel malesuada egestas, nisi mi accumsa
    n tellus, at rhoncus dolor nisi in tortor. Donec at aliquam lectus, ac egestas odio. Proin
    convallis sapien id consectetur egestas. Sed at varius libero, ut ultricies eros. Suspendi
    sse potenti. Vivamus non posuere mauris. Maecenas luctus elementum justo, non tempor nisl
    blandit id. Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

</body>

</html>

```

## Clear

- clear 속성은 float 속성이 적용된 이후 나타나는 요소들의 동작을 조절해 줍니다.
- 컨테이너 요소에 float 속성이 적용되면 그 이후에 등장하는 모든 요소들은 정확한 위치를 설정하기가 매우 힘들어집니다.

## 예제1

```

<!DOCTYPE html>
<html lang="ko">

<head>
  <meta charset="UTF-8">
  <title>CSS Float</title>
  <style>
    .left {
      background-color: #FF8C00;
      width: 150px;
      height: 50px;
      float: left;
    }
    .right {
      background-color: #9932CC;
      width: 150px;
      height: 50px;
      float: right;
    }
  </style>
</head>

<body>

  <h1>clear 속성을 이용한 위치 조정</h1>
  <div>

```

```

    <div class="left">왼쪽 끝에 위치하고 싶은 요소</div>
    <div class="right">오른쪽 끝에 위치하고 싶은 요소</div>
  </div>
  <p>이 글자를 아래쪽에 제대로 출력하고 싶어요!</p>

</body>

</html>

```

따라서 float 속성을 적용하고자 하는 요소가 모두 등장한 이후에는 clear 속성을 사용하여, 이후에 등장하는 요소들이 더는 float 속성에 영향을 받지 않도록 설정해줘야 합니다.

## 예제2

```

<!DOCTYPE html>
<html lang="ko">

<head>
  <meta charset="UTF-8">
  <title>CSS Float</title>
  <style>
    .left {
      background-color: #FF8C00;
      width: 150px;
      height: 50px;
      float: left;
    }
    .right {
      background-color: #9932CC;
      width: 150px;
      height: 50px;
      float: right;
    }
    p { clear: both; }
  </style>
</head>

<body>

  <h1>clear 속성을 이용한 위치 조정</h1>
  <div>
    <div class="left">왼쪽 끝에 위치하고 싶은 요소</div>
    <div class="right">오른쪽 끝에 위치하고 싶은 요소</div>
  </div>
  <p>이 글자를 아래쪽에 제대로 출력하고 싶어요!</p>

</body>

</html>

```

