

CSS3

# 1. CSS의 개요와 기본 사용법

## 1.1 CSS(Cascading Style Sheet)

CSS는 웹 페이지의 내용적인 부분과 분리해서 따로 지정한 서식을 의미하며 스타일시트라고도 부른다. 이 스타일시트는 HTML 태그만을 사용하여 웹 문서를 작성할 때 디자인 적인 한계를 보완하기 위해 개발된 독립적인 마크업 언어이다. HTML은 웹 문서를 빠르고 간단하게 작성하여 인터넷을 통해 정보를 공유할 수 있도록 고안된 마크업 언어이다. HTML은 웹 문서의 내용을 빠르고 간편하게 작성하는데 많은 비중을 두고 개발된 언어이므로 웹 페이지의 레이아웃, 타이포그래피, 배경, 테두리 등의 디자인 적인 스타일을 적용하기에는 맞지 않은 언어이다. 그래서 CSS는 HTML의 디자인적인 한계를 보완하고 문서의 내용과 디자인 적인 요소를 분리하여 문서를 작성하고 효율적으로 관리하기 위해 많이 사용된다.

CSS도 W3C(World Wide Web Consortium)에서 제정한 웹 표준 중에 하나이며 HTML5는 2014년 10월 정식 권고안이 발표되었고(<http://www.w3.org/TR/html5> 참고) 2017년 12월에 5.2 버전의 권고안이 발표되었으며 CSS3와 ECMAScript(자바스크립트의 공식 명칭)도 새로운 권고안이 발표되었다. CSS3에서는 기존의 CSS2.1에서 지원하던 여러 가지 기능을 더 확장해 지원하고 있다. 예를 들어 둥근 테두리는 CSS2.1에서는 지원하지 못하였으나 CSS3를 사용하면 둥근 테두리가 있는 박스를 쉽게 만들 수 있으며 웹 문서의 객체를 더 편리하게 선택 할 수 있도록 많은 선택자가 추가되었다. 현재는 div 요소와 CSS를 사용하여 웹 페이지 레이아웃을 구성하는 것은 가장 일반적인 레이아웃 기법으로 웹 페이지에서 CSS와 JavaScript를 연동해 동적으로 레이아웃을 구성하는 다양한 기법이 활용되고 있다.

## 1.2 웹 문서에 CSS를 적용하는 방법

CSS의 적용 방법은 크게 3가지로 나눌 수 있는데 그 첫 번째가 HTML 태그에서 style 속성으로 스타일을 정의하여 사용하는 인라인 스타일 방식이 있고 두 번째로 웹 문서의 내부에 스타일시트를 정의하여 사용하는 내부 스타일시트 방식이 있으며 마지막 하나는 별도의 외부 파일에 스타일시트를 정의하고 현재의 웹 문서에서 외부에 있는 스타일시트 파일을 연결하여 사용하는 외부 스타일시트 방식이 있다. 외부 스타일 방식에는 외부 스타일시트(External Style Sheet) 연결방식과 외부 스타일시트 포함(Imported Style Sheet) 방식이 있다. 이제 웹 문서에 CSS를 적용하는 방법을 하나 하나 자세히 알아보도록 하자.

### 1.2.1 인라인 스타일

인라인 스타일(Inline style)은 HTML 태그의 style 속성에 스타일을 정의하는 방식이다.

인라인 스타일을 사용하는 방식은 아래와 같이 각 요소의 시작 태그에 style 속성을 사용해 CSS 속성 명(property)과 속성 값(value)을 콜론(:) 으로 구분하여 지정하고 하나의 속성 지정이 끝나면 스타일 문장의 끝을 알리는 세미콜론(;)을 기술한다. 이런 방식으로 HTML 요소의 style 속성에 세미콜론(;)으로 구분하여 여러 개의 스타일 속성을 아래와 같이 연속적으로 지정할 수 있다. 여기서 세미콜론은 각 스타일을 구분하는 역할을 하므로 선언 구분자라고 부른다.

```
<body>
  <p style="속성 명1: 속성 값1; 속성 명2: 속성 값2;">내용</p>
  <p style="border: 1px solid red; padding: 5px;">내용</p>
</body>
```

## 1.2.2 내부 스타일시트

내부 스타일시트(Internal style sheet)는 HTML 문서의 <head></head> 태그 사이에 스타일을 정의한다. 스타일을 정의하는 방식은 아래와 같이 <style> 태그를 사용하고 그 안에서 스타일을 적용할 문서 객체를 선택하는 선택자(selector)와 문서 객체에 지정할 스타일을 정의하는 선언부(declaratives)를 중괄호({})를 사용해 정의한다. 스타일 정의는 중괄호인 “{”와 “}” 사이에 CSS 속성명(property)과 속성 값(value)을 콜론(:) 으로 구분하여 지정한다. 선택자 하나의 선언부에는 아래와 같이 여러 스타일을 지정할 수 있으며 하나의 스타일 지정이 끝나면 스타일 문장의 끝을 알리는 세미콜론(;)을 사용한다. 이와 같이 <style> 태그에 선택자를 이용해 문서에 스타일을 설정하는 것을 스타일 규칙(Style rule)이라고 한다. CSS는 요소, 아이디, 클래스 등과 같은 다양한 선택자를 지원하고 있으며 선택자에 대해서는 다음 Chapter에서 자세히 알아 볼 것이다.

```
<head>
  <style type="text/css">
    선택자(selector) { 속성 명1: 속성 값1; 속성 명2: 속성 값1; }
    div { border: 1px solid red; background-color: #EAF7FF; }
  </style>
</head>
```

## 1.2.3 외부 스타일시트

위에서 언급했듯이 별도의 파일로 정의한 스타일을 웹 문서에 적용하는 방법에는 외부 스타일시트(External Style Sheet)를 연결하는 방식과 외부 스타일시트(Imported Style Sheet)를 포함하는 두 가지 방식이 있다.

### ▶ HTML문서에 외부 스타일시트 연결하기

별도의 파일로 스타일을 정의하고 웹 문서에 외부 스타일시트 파일(External Style Sheet)을 연결하여 사용하는 방식으로 아래와 같이 <head></head> 태그 사이에 연결할 스타일시트 파일을 <link> 태그의 href 속성에 지정하여 연결한다. 이 방식은 HTML 문법을 이용해 외부 스타일시트를 현재 문서에 연결하는 방식이다.

```
<head>
  <link rel="stylesheet" type="text/css" href="external_style.css" />
</head>
```

```
* external_style.css 파일
선택자(selector) { 속성명1: 속성 값1; 속성명2: 속성 값2; }
div {
    border: 1px solid red;
    background-color: #FF0000;
}
```

## ▶ HTML 문서에 외부 스타일시트 포함하기

별도의 파일로 스타일을 정의하고 웹 문서에 스타일시트 파일을 포함(Imported Style Sheet)하여 사용하는 방식으로 내부 스타일시트 방식과 비슷하게 <head></head> 태그 사이에 <style> 태그를 사용하고 그 안에 CSS 구문을 기술하는 자리에 @import url('external\_style01.css');로 외부 스타일시트 파일을 지정하는 방식이다. 이 방식은 HTML 문법을 사용하는 것이 아니라 CSS 문법을 사용하여 외부 스타일시트 파일을 현재 문서의 스타일에 포함시키는 방식이다. 이렇게 외부 스타일시트를 현재 문서에 적용하는 방식에는 HTML 문법을 이용하는 방식과 CSS 문법을 이용하는 방식 두 가지가 있다. 두 가지 방식 모두 스타일이 적용되는 결과는 동일하다.

```
<head>
<style type="text/css">
    @import url('external_style01.css');
    @import url('external_style02.css');
</style>
</head>
```

## 1.3 스타일 우선순위

웹 문서는 스타일이 여러 곳에서 사용되기 때문에 스타일 적용의 우선순위에 대해 잘 이해하고 있어야 한다. 스타일은 상속의 개념과 CSS 용어 자체에 담긴 캐스케이딩(Cascading)의 개념이 적용된다.

### 1.3.1 스타일은 상속된다.

HTML 문서에서 사용하는 여러 요소들은 서로 포함 관계를 가지고 있다. 그래서 다른 요소를 포함하는 요소를 부모 요소라 하고 다른 요소에 포함된 요소를 자식 요소라고 한다.

스타일에서 부모 요소에 적용된 속성이 자식 요소로 전달되는데 이를 스타일 속성의 상속이라 한다. 한 가지 주의할 점은 스타일의 상속은 모든 속성이 전부 상속되는 것이 아니라 상속되는 속성이 있고 상속되지 않는 속성이 있다.

아래 URL 주소는 모질라 개발자 네트워크(MDN, Mozilla Developer Network)의 상속되는 스타일 속성에 대한 문서이다.

<https://developer.mozilla.org/ko/docs/Web/CSS/Inheritance>

## ▶ 예제 1-1 스타일의 상속 1

- CssStudyCh01/css01\_01.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>스타일의 상속1</title>
<style type="text/css">
  body {
    background-color: #333;
    color: white;
    border: 1px solid blue;
  }
  div {
    border: 1px dotted red;
  }
  p{
    border: 1px dashed gray;
  }
</style>
</head>
<!--
```

스타일의 상속은 모든 속성이 상속되는 것이 아니라 상속되는 속성이 있고 상속되지 않는 속성이 있다는 것이다.

아래 URL 주소는 모질라 개발자 네트워크(MDN, Mozilla Developer Network)의 상속되는 스타일 속성에 대한 문서이다.

<https://developer.mozilla.org/ko/docs/Web/CSS/Inheritance>

```
-->
<body>
  <div>
    <strong>body 요소 안의 div 요소</strong>
    <p><strong>body > div > p 요소</strong>
    div 요소는 body 요소의 자식 요소로 body 요소의 스타일이 상속된다.<br/>
    또한 p요소는 body 요소의 자손 요소로 body 요소의 스타일이 상속되며<br/>
    div 요소의 자식 요소로 div 요소의 스타일이 상속된다.
  </p>
</div>
</body>
</html>
```

## ▶ 예제 1-2 스타일의 상속 2

- CssStudyCh01/css01\_02.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>스타일의 상속2</title>
<style type="text/css">
  ul {
    background: black;
    color: white;
    list-style: square;
  }
</style>
</head>
<body>
  <!--
    ul 요소는 li 요소의 부모 요소로 ul 요소에 지정한 color 속성이
    자식인 li 요소로 상속되어 글자색이 흰색으로 표현된다.
  -->
  <section>
    <h2>웹표준 도서 목록</h2>
    <ul>
      <li>HTML5+CSS3+JavaScript 웹 프로그래밍</li>
      <li>HTML과 CSS로 웹 프로그래밍 시작하기</li>
    </ul>
    <h2>자바스크립트 도서 목록</h2>
    <ul>
      <li>자바스크립트 웹 표준의 정석</li>
      <li>모던 자바스크립트 활용</li>
    </ul>
  </section>
</body>
</html>
```

### 1.3.2 스타일은 단계적으로 적용된다.

CSS는 캐스케이딩 스타일시트(Cascading Style Sheet)의 줄임말로 단계적으로 스타일이 적용된다는 의미를 지니고 있다. 여러 문서에 스타일을 적용하다 보면 같은 요소에 여러 스타일이 겹쳐서 지정되는 경우가 있는데 이럴 경우 우선순위에 따라 스타일이 적용된다.

하나의 요소에 여러 스타일이 지정되어 있다면 맨 마지막에 지정된 스타일이 적용된다.

만약 다음과 같이 스타일이 지정되어 있다면 나중에 지정된 글자 크기인 12px이 적용된다.

```
<style type="text/css">
```

```
p {
    font-size: 20px;
    font-size: 12px;
}
</style>
```

<p>CSS는 캐스케이딩 스타일시트(Cascading Style Sheet)의<br/>  
줄임말로 단계적으로 스타일이 적용된다는 의미를 지니고 있다.</p>

이렇게 중복 지정된 스타일이 존재할 경우 예상하지 못한 결과가 나타날 수 있다.  
이럴 경우 아래와 같이 우선순위를 지정할 수 있는데 스타일 규칙에 !important를 지정하게 되면  
해당하는 스타일 규칙이 가장 높은 우선순위를 가지게 된다.

```
p {
    font-size: 20px; !important;
}
```

우리는 앞에서 스타일이 적용될 때 단계적으로 적용되며 같은 요소에 적용된 스타일은 가장 마지막  
에 지정된 스타일이 적용된다는 것을 알았다. 그렇다면 동일한 요소에 외부 스타일, 내부 스타일,  
인라인 스타일 모두에서 동일한 요소에 스타일 규칙을 지정하였다면 그 우선순위는 어떻게 될까?  
간단한 예제를 통해 이를 알아보자.

### ▶ 예제 1-3 동일한 요소에 적용된 스타일 우선순위

- CssStudyCh01/css01\_03.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>동일한 요소에 적용된 스타일 우선순위</title>
<link type="text/css" href="css/external.css" rel="stylesheet" />
<style>
    div {
        font-size: 14px;
    }
</style>
</head>
<body>
    body 요소의 폰트 크기는 외부 스타일에서 지정한 20px이 적용된다.
    <div>
        div 요소의 폰트 크기는 내부 스타일에서 지정한 14px이 적용된다.
```

<!--

동일한 요소에 동일한 스타일이 적용되었다면 우선순위는 다음과 같다.

1. !important; 가 적용된 스타일 속성
2. 인라인 요소의 style 속성에 적용된 스타일 속성
3. 내부 스타일 속성
4. 외부 스타일 속성

아래 p 요소의 폰트 크기는 external.css에서 p 요소의 font-size에 !important를 적용했기 때문에 20px이 우선 적용된다.

-->

<p style="font-size: 12px;">

p요소의 폰트 크기는 인라인 스타일에서 지정한 12px이 적용되지 않고<br/>  
external.css에서 !important로 지정한 20px이 우선 적용된다.

</p>

</div>

</body>

</html>

\* WebContent/css/external.css

@CHARSET "UTF-8";

body {

font-size: 40px;

}

div {

font-size: 20px;

}

p {

font-size: 20px !important;

}

## 1.4 CSS 작성시 필요한 기본기념 익히기

### 1.4.1 요소와 태그, 블록레벨 요소와 인라인레벨 요소 구분하기

#### ▶ 요소(Elements)

html에서 "<"와 ">"로 둘러싸인 시작 태그와 "</"와 ">"로 둘러싸인 종료 태그로 이루어진 태그의 조합을 의미한다.

img, hr, br 등과 같이 내용물이 없는 요소를 비어있는 요소(Empty Element) 또는 비어있는 태그(Empty Tag)라고 부르며 비어있는 요소는 하나의 태그에 "<"와 ">"를 사용해 요소의 시작과 끝을 표현할 수 있으며 이런 태그 들을 셀프 클로징 태그(Self Closing Tag)라고 부른다.

셀프 클로징 태그는 xhtml과 xml에서 사용했던 문법으로 HTML5에서는 선택적(Optional)으로 사용할 수 있지만 일반적으로 잘 사용하지 않는다.



## ▶ 태그(Tag)

태그는 요소의 일부로 “<”와 “>”로 둘러싸인 시작 태그(시작 태그를 여는 태그라고도 함)와 “</”와 “>”로 둘러싸인 종료 태그(종료 태그를 닫는 태그라고도 함) 두 종류가 있으며 시작 태그는 요소의 시작을 종료 태그는 요소의 끝을 의미하지만 일반적으로 요소와 태그의 개념이 뚜렷하게 구별 되지 않고 혼용 되는 경우가 많다.

CSS나 자바스크립트에서 요소라는 의미는 매우 중요하다. 요소는 “<h1>문서의 제목</h1>” 전체를 의미 하지만 태그는 <h1>과 </h1>을 각각 시작 h1 태그와 종료 h1 태그를 의미한다. 예를 들면 <p>, <div>와 같이 여는 태그를 부를 때 “p 태그”, “div 태그” 라고 부른다. 하지만 <p></p>와 <div></div>와 같이 태그의 조합을 부를 때는 각각 “p 요소”, “div 요소”라고 부르는 것이 정확한 표현이다.

## ▶ 블록레벨(Block-level) 요소(Element)

브라우저의 안에서 블록처럼 쌓여가면서 줄 바꿈을 하여 표시 되는 요소로 전후의 요소들과 줄 바꿈 되어 화면에 표시된다. 블록레벨 요소는 아래와 같은 특징을 가지고 있다.

- 한 라인에 하나의 요소만 표시되므로 항상 줄 바꿈을 하여 새로운 줄에 표시된다.
- 너비 값을 지정하지 않으면 브라우저 화면 또는 부모에 100%로 꽉 차게 표시된다.
- 블록레벨 요소는 width, height, margin, padding 등의 값을 가질 수 있다.
- 블록레벨 요소와 인라인레벨 요소를 자식요소로 포함 할 수 있다.
- vertical-align 속성이 무시되며 대표적인 블록레벨 요소는 아래와 같다.  
body, div, p, pre, form, h1 ~ h6, table, ul, ol, dl, menu, fieldset, noscript, hr, address, blockquote...등

## ▶ 인라인레벨(Inline-level) 요소(Element)

블록레벨과 반대의 개념으로 앞에 표시된 요소에 이어서 같은 줄 안에 표시되며 넓이도 자신의 내용물의 크기만큼 만 차지한다. 인라인레벨 요소는 다음과 같은 특징을 가지고 있다.

- 너비 값은 요소가 가지고 있는 값 자체로 이므로 width, height, margin, padding 등의 값을 지정해도 화면에 반영되지 않는다.
- 웹 표준을 지키면서 인라인 레벨 요소를 사용하고 싶다면 항상 블록레벨 요소로 인라인 레벨 요소를 감싸 기술해야 한다. 즉 인라인 요소는 단독으로 사용될 수 없으며 div, p 요소와 같은 블록레벨 요소 안에 기술되어야 한다.
- 대표적인 인라인레벨 요소는 아래와 같다.  
span, img, map, button, input, label, select, textarea, a, strong, em object, script, br, b, sub, sup, i, abbr, cite, samp, var... 등

## ▶ 그룹화 요소(Grouping Element)

예전에는 Table이나 Frame을 활용해 화면 레이아웃을 구성하는 경우가 많았지만 현재는 HTML5의

시멘틱 요소나 공간 분할 요소인 div, span 요소를 활용해 그룹화 하고 CSS를 이용해 디자인적인 스타일을 적용해 화면을 만드는 것이 일반적이다.

웹 페이지를 구성하는 콘텐츠를 화면에 배치하고 표현하기 위해 그룹화 할 수 있도록 지원하는 요소가 div, span 요소이다. 웹 페이지 작성시 div 요소를 이용하여 여러 요소를 논리적으로 그룹화 할 수 있고 그 그룹에 이름을 지정하여 CSS를 이용해 스타일을 적용할 수 있다. div는 division의 약자로 블록레벨 요소에 속하며 div에 자식으로 속한 여러 요소를 그룹화 하여 공간을 지정하는 컨테이너 역할을 하는 요소이다. div는 특정 의미를 가지고 있지 않는 요소로 블록레벨 요소를 그룹화 하여 스타일을 지정하기 위해 사용된다. span 요소는 인라인레벨 요소의 대표 요소로 div와 같이 특정 의미를 가지고 있지 않는 그룹화 요소이다. span은 인라인레벨 요소를 그룹화하고 스타일을 적용하기 위해 사용된다. 인라인레벨 요소 중 span 이외의 그룹핑 요소로는 a, img, select, input 등이 있으며 이들은 특정 의미를 지닌 요소 들이다. div 요소와 span 요소는 웹 페이지에서 블록레벨 요소와 인라인 레벨 요소를 그룹으로 묶어서 웹 페이지의 공간을 분할하여 다양한 형태의 요소를 배치할 수 있도록 지원하는 요소로 공간 분할 요소 또는 공간 분할 태그라고도 부른다.

## ▶ 예제 1-4 블록레벨 요소와 인라인레벨 요소

### - CssStudyCh01/css01\_04.html

이번 예제는 블록레벨 요소와 인라인레벨 요소의 특징을 잘 알 수 있는 예제이다. 예제를 실행해 보면 css에서 지정한 width와 height가 블록레벨 요소인 h3 요소에만 적용되는 것을 볼 수 있다. 또한 인라인레벨 요소인 b 요소와 em 요소가 줄 바꿈 없이 한 라인에 표시된다. 그러나 나머지 블록레벨 요소는 전 후의 요소들과 앞뒤로 줄 바꿈이 일어난 것을 볼 수 있다.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>블록레벨 요소와 인라인레벨 요소</title>
  <style type="text/css">
    * { font-size: 12px; }
    h2 {
      font-size: 16px;
      font-weight: bold;
    }
    h3 {
      width: 300px;
      height: 70px;
      color: white;
      font-size: 14px;
      font-weight: bold;
      background-color: black;
    }
    strong {
      width: 300px;
```

```

    height: 100px;
    color: #FFFFFF;
    background: #770000;
}
em {
    width: 500px;
    height: 100px;
    font-weight: bold;
    color: #FFFFFF;
    background: #FF0000;
}
</style>
</head>
<body>
    <h2>블록레벨과 인라인레벨</h2>
    <div>
        <h3>블록 요소</h3>
        <p>브라우저의 안에서 블록처럼 쌓여가면서 줄 바꿈을 하여 표시 되는 엘리먼트로
        <strong>전후의 요소들과 줄 바꿈</strong> 되어 화면에 표시된다.</p>
        <h3>인라인 요소</h3>
        <p>블록요소와 반대의 개념으로 <b>앞에 표시된 엘리먼트에 이어서 같은 줄 안에
        표시되며</b> <em>넓이도 자신의 내용물의 크기만큼 만 차지한다.</em></p>
    </div>
</body>
</html>

```

### 1.4.2 CSS 단위 이해하기

CSS 문서를 작성할 때 속성에 대한 값은 특정한 단위를 지정해 줘야한다. 특히 width나 height 속성과 같이 크기에 대한 값을 입력할 때 단위를 생략하면 0 값을 제외한 나머지는 제대로 적용되지 않는다. CSS에는 속성에 대한 값을 지정할 때 이미 예약된 키워드를 입력하는 경우도 있으며 크기를 지정할 때는 절대 또는 상대 크기의 단위를 지정해야 하는 경우도 있다. 또한 배경색이나 전경색을 지정하기 위해 색상을 수치로 입력하는 경우도 있고 배경의 이미지 등을 지정하기 위해 URL을 기술하는 경우도 있다. 이렇게 CSS에서 사용되는 속성 값에 대한 단위는 여러 가지가 존재한다. 여기서는 HTML 요소의 크기를 지정할 수 있는 크기 단위와 색상과 관련된 색상 단위만 알아볼 것이다. URL 지정 단위는 CSS를 Imported Style Sheet로 HTML에 연결할 때 다루어 봤듯이 이미지나 파일의 위치를 상대 또는 절대경로 방식으로 지정하는 것이므로 쉽게 이해할 수 있을 것이다. 키워드를 속성의 값으로 지정하는 경우는 속성에 따라 지정하는 키워드 값이 서로 다르고 많기 때문에 여기서는 전부 다룰 수 없어 중요한 키워드 값을 중심으로 예제를 통해 다루어 볼 예정이니 나머지는 레퍼런스 등을 참조하기 바란다.

#### ▶ CSS의 크기 단위 이해하기

CSS3에서 사용되는 크기 단위는 mm, cm, in, pt, px, %, em, 가 있다 이 외에도 글꼴의 크기를

지정하는 경우 미리 예약된 키워드를 단위로 사용할 수 있으며 다음 표에서 설명하는 3가지 단위가 가장 많이 사용되고 있다.

표 1-1 자주 사용되는 CSS의 크기 단위

단 위	설 명
px	화면을 구성하는 픽셀을 단위로 사용하는 절대적인 크기 단위로 화면 해상도에 따라 보여 지는 크기가 다를 수 있다.
%	부모 요소를 기준으로 백분율을 사용하는 상대적인 크기 단위
em	em은 인쇄술에서 대문자 M의 크기를 나타내는 단위로 CSS에서는 부모 요소를 기준으로 배수를 사용하는 상대적인 크기 단위
rem	최상위 요소를 기준으로 배수를 사용하는 상대적인 크기 단위

## ▶ CSS 색상 단위 이해하기

CSS 색상 단위는 글꼴의 색상을 지정하거나 요소의 전경색, 배경색을 지정할 때 사용되는 단위로 색상 값을 지정하는 키워드인 white, black, red... 등의 색상 값도 지정할 수 있으며 10진수나 16진수 값의 수치를 지정할 수도 있다. CSS에서 다음과 같은 색상 단위가 사용된다.

표 1-2 CSS의 색상 단위

단 위	설 명
#FF00AA	16진수 단위(HEX 코드 단위)
rgb(red, green, blue)	빛의 3원색인 Red, Green, Blue를 사용하는 단위
rgba(red, green, blue, alpha)	RGB에 투명도(Alpha)를 지정 할 수 있는 단위 투명도는 0.0(완전투명) ~ 1.0(불투명)까지 지정 할 수 있으며 CSS3에서 추가된 속성이다.
hsl(hue, saturation, lightness)	색의 3속성인 색상, 명도, 채도를 사용하는 단위 CSS3에서 추가된 속성이다.
hsla(hue, saturation, lightness)	HSL에 투명도(Alpha)를 지정 할 수 있는 단위 CSS3에서 추가된 속성이다.

## ▶ 예제 1-5 CSS의 크기 단위와 색상 단위

- CssStudyCh01/css01\_05.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>CSS의 단위 이해하기</title>
  <style type="text/css">
```

```

/* font 관련 속성들은 상속되기 때문에 rem 단위는 최상위 html 요소의 font-size를
 * 기준으로 배수 크기로 표현되며 em과 %는 부모 요소의 font-size를 기준으로 각각
 * 배수 크기와 백분율 크기로 표현된다. 아래에서 font-size에 rem 단위를 사용했는데
 * html 요소에 font-size가 별도로 지정되어 있지 않기 때문에 html 문서의 기본
 * font-size인 16px이 적용되어 이 크기를 기준으로 지정한 배수 크기로 표현된다.
 * 또한 em과 % 단위는 부모 요소가 기준이 되므로 p 요소의 부모인 body 요소의
 * font-size 12px를 기준으로 지정한 배수 또는 백분율 크기로 표현된다.
 */
body { font-size: 12px; }
p { width: 700px; height: 50px; }
.per75 { font-size: 75%; }
.per150 { font-size: 150%; }
.em075 { font-size: 0.75em; }
.em150 { font-size: 1.5em; }
.rem075 { font-size: 0.75rem; }
.rem150 { font-size: 1.5rem; }
.px12 { font-size: 12px; }
.px16 { font-size: 16px; }

.black { background-color: black; color: white; }
.alpha50 { background-color: rgba(255,0, 0, 0.5); color: black; }
.alpha70 { background-color: rgba(200, 85, 85, 0.7); }
.alpha30 { background-color: hsla(108, 83%, 47%, 0.3); color: red; }
.blue { background-color: #0000FF; color: white; }
</style>
</head>
<body>
<h2>CSS의 크기 단위와 색상 단위 이해하기</h2>
<p>HTML 문서에서 폰트의 기본 크기는 16px로 표시된다.</p>
<p class="per75 black">부모 요소 폰트 크기의 75% 크기</p>
<p class="per150 alpha50">부모 요소 폰트 크기의 150% 크기</p>
<p class="em075 black">부모 요소 폰트 크기의 0.75배 크기</p>
<p class="em150 blue">부모 요소 폰트 크기의 1.5배 크기</p>
<p class="rem075 black">루트 요소 폰트 크기의 0.75배 크기</p>
<p class="rem150 alpha70">루트 요소 폰트 크기의 1.5배 크기</p>
<p class="px12 alpha30">12px로 지정한 폰트 크기</p>
<p class="px16 alpha50">16px로 지정한 폰트 크기</p>
</body>
</html>

```

## 2. CSS 선택자

CSS에서 HTML 문서의 특정 요소에 스타일을 지정하기 위해 그 요소를 선택할 수 있는 태그나 클래스 또는 아이디 등을 일컫는 말로 선택하는 요소를 구분할 수 있는 명칭을 선택자(Selector)라 한다. 또한 선택자는 HTML 문서의 특정 요소에 CSS를 통해 스타일을 적용하거나 효과를 적용하기 위해 사용되는 CSS 문법의 필수 구성요소이다.

CSS에서 사용되는 선택자는 마크업 기반의 자바스크립트 프레임워크인 jQuery에서 대부분 사용할 수 있다. 다음은 CSS에서 사용할 수 있는 선택자의 종류와 설명을 표로 나타낸 것이다. 언뜻 봐도 CSS에서 지원하는 선택자의 종류는 상당히 많은 수준이다. 아래 표 2-1은 CSS에서 사용하는 선택자의 종류를 정리한 것이다. 아래 표를 참고하여 선택자의 종류와 그 사용법에 대해 자세히 알아보도록 하자

표 2-1 CSS 선택자의 종류

선택자	선택자 사용 예	ver	설 명
전체(공통) 선택자	*	2	문서의 전체 요소를 선택
요소(태그) 선택자	p	1	p 요소 전체를 선택
아이디 선택자	#user	1	id 속성 값이 user인 요소 선택
클래스 선택자	.members	1	class 속성 값이 members인 요소 선택
자손 선택자	header h1	1	header 요소 아래의 h1 요소 전체 선택
자식 선택자	header > h1	1	header 요소 바로 아래의 h1 요소 선택
종속 선택자	p.row, p#row	1	p 요소의 클래스 값이 row인 p요소 선택 p 요소의 아이디 값이 row인 p요소 선택
그룹 선택자	p, h1, a	1	p, h1, a 요소를 모두 선택
속성 선택자	input[type]	2	type 속성을 가진 요소
속성 값 선택자	input[type=text]	2	type이 text인 요소를 선택
	div[class~=name]	2	class 속성 값이 name과 일치하거나 여러 개의 class 속성 값 중에서 name과 일치하는 요소선택
	div[id^=name]	3	id의 값이 name으로 시작하는 요소선택
	img[src\$=jpg]	3	src 속성 값이 jpg로 끝나는 요소 선택
	div[class*=user]	3	class 속성 값이 user를 포함하는 요소선택
인접 선택자	h1 + h2	2	h1 요소 바로 뒤에 위치하는 h2 요소 선택
	h1 ~ h2	3	h1 요소 뒤에 위치하는 h2 요소들 선택
링크 가상클래스 선택자	a:link	1	방문하지 않은 모든 링크를 선택
	a:visited	1	이미 방문했던 모든 링크를 선택

선택자	선택자 사용예	ver	설 명
반응 가상클래스 선택자	a:active, h1.active	1	활성화된 요소를 선택(클릭되고 있는 요소)
	a:hover, div.hover	1	마우스 오버된 요소를 선택
상태 가상클래스 선택자	input:focus	2	포커스된 요소를 선택
	input:checked	3	체크 상태인 input 요소를 선택
	input:enabled	3	사용 가능한 input 요소를 선택
	input:disabled	3	사용 불가능한 input 요소를 선택
위치구조 가상클래스 선택자	li:first-child	2	어떤 요소의 자식 중 첫 번째 li 요소 선택
	li:last-child	3	어떤 요소의 자식 중 마지막 번째 li 요소 선택
	li:nth-child(2n)	3	어떤 요소의 자식 중 2n번째 li 요소 선택
	li:nth-last-child(2n)	3	어떤 요소의 자식 중 뒤에서 2n번째 li 요소 선택
타입구조 가상클래스 선택자	h1:first-of-type	3	h1 요소 중 처음으로 등장하는 요소 선택
	h1:last-of-type	3	h1 요소 중 마지막으로 등장하는 요소 선택
	h1:nth-of-type(2n)	3	h1 요소 중 앞에서 2n 번째로 등장하는 요소 선택
	h1:nth-last-of-type(2n)	3	h1 요소 중 뒤에서 2n 번째로 등장하는 요소 선택
문자 가상요소 선택자	p::first-letter p:first-letter	1	p 요소 안의 첫 번째 글자를 선택
	p::first-line, p:first-line	1	p 요소의 첫 번째 라인을 선택
	p::after, p:after	2	p 요소 내용의 뒤에 위치하는 공간을 선택
	p::before, p:before	2	p 요소 내용의 앞에 위치하는 공간을 선택
	p::selection, p:selection	3	사용자가 드래그 하여 선택한 텍스트를 선택
부정 선택자	input:not(type=text)	3	input 요소의 type 속성 값이 text가 아닌 것을 선택

CSS에서는 가상(Pseudo)이라는 개념을 적용해 HTML 문서에 실제로 존재하지 않아 접근할 수 없는 문서 모델(DOM-Document Object Model)에 접근하여 CSS 문서를 용이하게 작성할 수 있도록 해주는 가상요소 선택자와 가상 클래스 선택자가 있다. 스타일을 적용하는 대상을 가상요소에 포함된 내용의 일부나 위치를 선택 할 수 있는 선택자를 가상요소 선택자라 하며 스타일을 적용하는 대상을 요소명이나 속성명 또는 속성 값에 따라 선택하는 것이 아니라 요소의 특정 상황이나 상태에 따라 선택해 주는 선택자를 가상 클래스 선택자라 한다. 위의 [표 2-1 CSS 선택자의 종류]를 참고하면 문자 가상요소 선택자인 p:first-letter은 HTML 문서의 문단으로 지정된 p요소 안에 첫 번째 글자를 선택하여 스타일을 적용할 수 있게 해 주고 상태 가상클래스 선택자인 input:checked는 HTML 문서의 input 요소 중에서 체크 상태인 모든 input 요소를 선택할 수 있도록 해 준다.

## 2.1 전체 선택자와 요소(태그) 선택자

전체 선택자는 HTML 문서 내의 모든 태그를 선택 할 때 사용하는 선택자로 문서의 여백이나 마진 등을 초기화 할 때도 많이 사용되는 선택자이다. 타입 선택자는 특정 태그를 선택할 때 사용되는 선

택자로 태그 선택자라고도 한다. 선택자의 사용방법은 [표 2-1 CSS 선택자의 종류]를 참고 하면 쉽게 이해 할 수 있을 것이므로 바로 예제를 통해 알아보자.

## ▶ 예제 2-1 전체 선택자와 요소(태그) 선택자

- CssStudyCh02/css02\_01.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>전체 선택자와 요소(태그) 선택자</title>
  <style type="text/css">
    * { font-size: 12px; }
    h3{
      font-size: 1.35em;
      font-weight: bold;
    }
    p {
      width: 600px;
      padding: 10px;
      border: 1px solid #C9C9C9;
    }
    em { font-weight: bold; }
  </style>
</head>
<body>
  <!--
    strong, em 요소는 중요한 내용을 강조하기 위해 사용하는 요소로
    strong 요소는 텍스트를 굵게 표시하고 em 요소는 이탤릭체로 표시한다.
    이와 비슷하게 텍스트를 표시하는 요소로는 b 요소와 i 요소가 있는데
    b 요소는 다른 내용과 구분해 텍스트를 단순히 굵게 표시할 경우 사용하며 강조의
    의미는 없다. i 요소 또한 다른 내용과 구분해 텍스트를 단순히 이탤릭체로 표시하기
    위해 사용하며 생각이나 꿈, 기술적인 용어, 다른 언어의 관용구 등에 사용한다.
  -->
  <h3>CSS 선택자</h3>
  <article>
    <p>CSS에서 HTML 문서의 특정 요소에 스타일을 지정하기 위해 그 요소를
    선택할 수 있는 태그나 클래스 또는 아이디 등을 일컫는 말로 선택하는 요소를
    구분할 수 있는 명칭을 <em>선택자(Selector)</em>라 한다.</p>
    <p>또한 선택자는 HTML 문서의 특정 요소에 CSS를 통해 스타일을 적용하거나
    효과를 적용하기 위해 사용되는 <strong>CSS 문법의 필수 구성요소</strong>
    이다.</p>
    <p>CSS에서 사용되는 <em>선택자</em>는 마크업 기반의 자바스크립트
    프레임워크인 <em>jQuery</em>에서 대부분 사용할 수 있다.</p>
```



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

## 2.2 클래스 선택자와 아이디 선택자

태그의 속성으로 id와 class 속성을 지정하고 CSS에서 아이디나 클래스 속성에 지정한 속성 값을 선택자로 지정하는 방법으로 웹 문서의 레이아웃을 구성하고 스타일을 적용할 때 CSS에서 가장 많이 사용되는 선택자 이다. HTML 문서에서 태그는 id와 class 속성을 가질 수 있으며 id 속성에 지정한 속성 값은 하나의 HTML 문서 내부에서 중복되지 않고 유일해야 한다는 웹 표준 규정이 있으므로 웹 표준을 지켜 문서를 작성하고자 한다면 중복 되서는 안 된다. 하지만 class 속성에 지정한 속성 값은 여러 요소에 중복으로 지정할 수 있다. 즉 아이디 선택자는 HTML 문서 내부에서 하나 이상 사용할 수 없고 클래스 선택자는 다중으로 사용할 수 있다.

### ▶ 예제 2-2 클래스 선택자와 아이디 선택자

- CssStudyCh02/css02\_02.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>클래스 선택자와 아이디 선택자</title>
  <style type="text/css">
    * {
      margin: 0px;
      padding: 0px;
      font-size: 12px;
    }
    header, section {
      width: 600px;
      margin: 10px auto 0;
    }
    #selectorDescription {
      height: 60px;
      line-height: 1.5em;
    }
    #classSelector {
      border: 1px dashed orange;
    }
    #idSelector {
      border: 1px dashed blue;
      margin-top: 20px;
    }
  </style>
</head>
<body>
  <div id="header">
    <h1>클래스 선택자와 아이디 선택자</h1>
  </div>
  <div id="section">
    <div id="selectorDescription">
      <p>selectorDescription</p>
    </div>
    <div id="classSelector">
      <p>classSelector</p>
    </div>
    <div id="idSelector">
      <p>idSelector</p>
    </div>
  </div>
</body>
</html>
```

```

.selectorTitle {
    background-color: #f03;
    color: #fff;
    margin: 5px 0 10px 5px;
}
.selectorTitle, .articleContent {
    padding: 5px;
}
.redDotLine {
    color: blue;
    border-bottom: 1px dotted red;
}
</style>
</head>
<body>
<header>
<h1>클래스 선택자와 아이디 선택자</h1>
</header>
<section>
<p id="selectorDescription">태그의 속성으로 id와 class 속성을 지정하고
CSS에서 아이디나 클래스 속성에 지정한 속성 값을 선택자로 지정하는
방법으로 웹 문서의 레이아웃을 구성하고 스타일을 적용할 때 CSS에서
가장 많이 사용되는 선택자 이다.</p>
<article id="classSelector">
<h3 class="selectorTitle">클래스 선택자</h3>
<p class="articleContent"><span class="redDotLine">id 속성에</span>
지정한 속성 값은 하나의 HTML 문서 내부에서 <span class="redDotLine">
중복되지 않고 유일해야 한다.</span></p>
</article>
<article id=idSelector>
<h3 class="selectorTitle">아이디 선택자</h3>
<p class="articleContent"><span class="redDotLine">class 속성에</span>
지정한 속성 값은 <span class="redDotLine">여러 요소에 중복으로
지정할 수 있다.</span></p>
</article>
</section>
</body>
</html>

```

위의 예제 2-2를 살펴보면 HTML 문서 내에 아이디 선택자는 p와 article 요소에 id 속성 값을 지정하였으며 클래스 선택자는 h3, p, span 요소에 중복해서 지정한 것을 볼 수 있을 것이다. 또한 밑줄 친 부분을 보면 각 요소에 공통된 스타일을 지정하기 위해 여러 선택자를 그룹으로 묶어 사용하였다. 이렇게 공통된 스타일을 지정하기 위해 선택자를 그룹으로 묶는 방법은 아주 간단하다 선택하고자 하는 요소의 선택자를 각각 콤마(,)로 구분하여 기술하면 된다.

## 2.3 자손관련 선택자

자손관련 선택자는 특정 요소의 하위에 속한 요소를 선택할 때 사용하는 선택자로 한 단계 아래에 소속된 하위 요소를 선택하기 위해 사용하는 자식 선택자와 전체 하위 요소 중에 지정한 모든 요소를 선택하는 자손 선택자가 있다. 사람으로 비유하자면 내 부모의 바로 아래에 위치한 나는 자식이고 나를 포함한 내 자식과 그 아래 자손들은 내 부모의 자손으로 부르는 것과 같다.

### ▶ 예제 2-3 자손관련 선택자

- CssStudyCh02/css02\_03.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>자식 선택자와 자손 선택자</title>
  <style type="text/css">
    * { font-size: 12px; }
    h2 { font-size: 30px; }
    section > h2 { font-size: 20px; }
    section > div > h2 { font-size: 16px; }
    section h2 { color: blue; }
    .description {
      width: 300px;
      height: 100px;
      padding: 10px;
      border: 1px solid darkgray;
    }
    p.description {
      height: 60px;
      padding: 10px;
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <header>
    <h2>자식 선택자와 자손 선택자</h2>
  </header>
  <section>
    <h2>자식 선택자</h2>
    <div class="description">
      한 단계 아래에 소속된 하위 요소를 선택하기 위해 사용하는 자식 선택자</div>
```

```

<h2>자손 선택자</h2>
<div class="description">
  전체 하위 요소 중에 지정한 모든 요소를 선택하는 자손 선택자</div>
<div>
  <h2>사람의 예</h2>
  <p class="description">
    내 부모의 바로 아래에 위치한 나는 자식이고 나를 포함한 내 자식과 그 아래
    자식에 자식들은 내 부모의 자손으로 부른다.</p>
  </div>
</section>
</body>
</html>

```

위의 예제 2-3 코드를 살펴보면 h2 요소 선택자를 이용해 문서 내의 모든 h2 요소에 폰트 크기를 30px로 지정하여 예제 2-3의 문서 구조에서 모든 h2 요소의 폰트 크기를 30px로 고정하였다. 그리고 “section > h2” 자식 선택자를 이용해 문서 내의 h2 요소 중 내용이 section 요소의 바로 아래에 위치한 h2 요소의 폰트 크기를 20px로 지정하였다. 그리고 “section > div > h2” 자식 선택자를 이용해 section 요소의 자식 요소인 div 요소의 바로 아래에 위치한 h2 요소의 폰트 크기를 16px로 지정하였다. 또한 코드 중에 밑줄 친 부분을 살펴보면 description 클래스 선택자를 사용해 문서 전체의 description 클래스에 지정한 스타일을 종속 선택자(“p.description”)를 이용해 p 태그의 description class 속성 값을 가진 요소만 따로 margin, padding, border 속성을 변경 하였다. 이번 예제와 같이 종속 선택자는 HTML 문서에서 같은 이름의 클래스가 여러 개 존재 할 때 특정 요소만 스타일을 지정하거나 변경할 때 유용하게 사용되는 선택자 이다.

## 2.4 속성 관련 선택자와 인접 선택자

속성 관련 선택자는 HTML 요소의 특정 속성을 가지고 있거나 특정 속성의 값이 지정한 값과 일치하는 요소를 선택해 주는 선택자 이다. 이외에도 CSS3에서 추가된 특정 속성의 값이 지정한 문자열로 시작하거나 그 문자열을 포함하고 있거나 혹은 그 문자열로 끝나는 요소를 선택해 주는 선택자도 있다. 인접 선택자는 특정 요소를 기준으로 문서 트리 구조상 같은 레벨에 위치한 요소를 선택하기 위해 사용되는 선택자이다. 특정 요소의 하위에 위치한 요소 중에서 형제 요소 들을 문서 구조상 같은 레벨에 위치한 요소라 볼 수 있으며 이 요소들 중 지정한 요소의 바로 뒤에 위치하거나 뒤쪽에 위치한 형제 요소를 선택할 수 있는 선택자이다. 인접 선택자는 동등한 레벨에 위치한 요소를 선택하기 때문에 동위 선택자라고도 한다.

아래 예제 2-4에서 사용한 속성 관련 선택자와 인접 선택자는 CSS3에 새로 추가된 선택자도 있으므로 예제를 테스트 할 때는 브라우저의 CSS3 지원정도에 따라 스타일이 제대로 적용되지 않을 수도 있으니 인터넷 익스플로러외에 크롬이나 엣지, 파이어폭스 등을 설치한 후 예제를 테스트하기 바란다.

### ▶ 예제 2-4 속성 관련 선택자와 인접 선택자

- CssStudyCh02/css02\_04.html





인터가 위치한 상태이거나(hover) 클릭된(active) 상태이면 지정한 스타일이 적용되고 마우스 포인터의 위치를 요소 밖으로 이동하거나 클릭을 해제 하면 스타일은 적용되지 않는다.

이 반응 가상클래스도 마우스 포인터가 요소 위에 위치했을 때 선택되거나 클릭된 상태의 요소를 선택 할 수 있는 두 가지 선택자로 나뉜다.

반응 가상클래스와 다음에 살펴볼 상태 가상클래스는 사용자의 반응에 따라 동적으로 스타일이 적용되는 특징이 있어 동적 가상클래스 선택자라고도 한다. 또한 이들 선택자는 링크 선택자, 반응 선택자, 상태 선택자 등으로 짧게 부르기도 한다.

## ▶ 예제 2-5 링크 가상클래스, 반응 가상클래스 선택자

- CssStudyCh02/css02\_05.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>링크 가상클래스와 반응 가상클래스 선택자</title>
<style type="text/css">
  * { font-size: 12px; }
  div { width: 600px; margin: 0 auto; }
  h2 {
    margin-top: 30px;
    text-align: center;
    font-size: 16px;
    font-weight: bold;
  }
  h3 { font-size: 14px; font-weight: bold; }
  #a_tag {
    height: 200px;
    padding: 10px;
  }
  #tags {
    height: 200px;
    padding: 10px;
    border: 1px dotted blue;
  }
  a { text-decoration: none; line-height: 200%; }
  a:link { color: blue; }
  a:visited { color: darkgray; }
  a:hover { color: red; } /* :link, visited 뒤에 위치해야 한다. */
  a:active { color: green; } /* :hover 뒤에 위치해야 한다. */

  #a_tag:hover { border: 1px dotted blue; }
  #a_tag:active { background-color: #EAEAEA; }
  #tags:hover { border: 1px solid red; }
```

```

#tags:active { background-color: #FFEAEA; }
</style>
</head>
<body>
  <h2>링크 가상클래스 선택자와 반응 가상클래스 선택자</h2>
  <div id="a_tag">
    <a href="#">현재 문서 새로그침</a><br/>
    <a href="http://www.naver.com">네이버 가기</a><br/>
    <a>href 속성이 없는 a 태그</a><br/>
  </div>
  <div id="tags">
    <h3>반응 선택자</h3>
    <p>반응 가상클래스 선택자는 사용자가 마우스 포인터를 특정 요소 위에 위치
    시키거나 클릭한 상황이 발생하면 그 요소를 선택해 주는 선택자 이다.</p>
    <div class="description">
      반응 가상클래스 선택자를 통해 스타일을 지정한 요소에 마우스 포인터가
      위치된 상태이거나(hover) 클릭된(active) 상태이면 지정한 스타일이 적용되고
      마우스 포인터의 위치를 요소 밖으로 이동하거나 클릭을 해제 하면
      스타일은 적용되지 않는다.
    </div>
  </div>
</body>
</html>

```

## 2.6 상태 가상클래스 선택자

상태 가상클래스 선택자는 입력 양식의 상태를 선택할 수 있는 선택자이다. 즉 초점(focus)이 맞춰진 input 요소 또는 체크된 input 요소 등과 같이 입력 요소의 상태가 CSS에서 지정한 상태일 때 스타일이 적용된다. checked(체크된 상태), enabled(사용 가능한 상태), disabled(사용 불가능한 상태)는 CSS3에서 추가된 상태 가상클래스 선택자 이다.

### ▶ 예제 2-6 상태 가상클래스 선택자

- CssStudyCh02/css02\_06.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>상태 가상클래스 선택자</title>
<style type="text/css">
  * { font-size: 12px; }
  div { line-height: 3em; }
  input {

```



```

padding: 3px 5px;
}
#id:focus, #pass:focus {
background-color: #FFEAEA;
border: 1px solid #FF9999;
outline: 0;
}

/* id 속성 값이 interest_select인 요소가 checked 상태이면
* 그 뒤로 오는 id가 check인 요소를 선택하여 선택한 요소의
* 너비와 높이를 지정하고 블록 레벨 요소로 화면에 표시되도록 설정한다.
**/
#interest_select:checked ~ #check {
width: 500px;
height: 10px;
display: block;
}

/* id 속성 값이 interest_noselect인 요소가 checked 상태이면
* 그 뒤로 오는 id가 check인 요소를 선택하여 선택한 요소를 화면에서
* 보이지 않도록 설정한다. display: none이 지정되면 화면에도
* 보이지 않고 문서의 그 위치에서 자리 차지도 하지 않는다.
**/
#interest_noselect:checked ~ #check {
display: none;
}

/* input 요소 중에서 disabled된 요소를 선택하여 배경색을 지정한다. */
input:disabled { background-color: gray; }

/* id 속성의 값이 choice인 요소가 체크된(checked) 상태이면
* 그 요소 뒤에 오는 id 속성의 값이 selection인 요소를 선택하여
* 선택한 요소의 높이 50px로 지정한다.
**/
#choice:checked ~ #selection {
height: 50px;
}

/* id 속성의 값이 choice인 요소 뒤에 오는 id 속성의 값이 selection인
* 요소를 선택하고 선택한 요소의 너비와 높이를 설정한다. 높이를 0px로
* 지정하였지만 #selection 요소의 내부에 선택상자가 있으므로 부모인
* #selection 요소의 높이가 0px이 되면 부모의 크기 밖으로 선택상자가
* 빠져나오기 때문에 #selection 요소의 내용물이 #selection 요소의
* 크기를 초과하면 화면에서 숨기도록 overflow: hidden으로 설정하였다.
**/

```

```

#choice ~ #selection {
    width: 300px;
    height: 0px;
    overflow: hidden;
}
</style>
<script>
function checkedEmail() {

    // html 문서에서 id 속성의 값이 input_email인 문서 객체를 가져온다.
    var chk = document.getElementById('input_email');
    // var chk = document.querySelector("#input_email");

    /* chk는 html 문서에서 체크박스이므로 이 체크박스가 체크된 상태면 checked
    * 속성은 true가 되어 if 문의 블록({}) 안의 코드가 실행되고 체크되지 않은
    * 상태이면 false가 반환되어 else의 블록({}) 안의 코드가 실행된다.
    */
    if(chk.checked) {

        /* html 문서에서 name 속성의 값이 form1인 폼 안에서 name 속성의
        * 값이 email인 input 요소를 찾아서 이 입력상자의 disabled 상태를
        * 해제하고 배경색을 흰색으로 글자색을 검정색으로 설정한다.
        */
        document.form1.email.disabled = false;
        document.form1.email.style.backgroundColor = '#FFFFFF';
        document.form1.email.style.color = 'black';

    } else {

        /* html 문서에서 name 속성의 값이 form1인 폼 안에서 name 속성의
        * 값이 email인 input 요소를 찾아서 이 입력상자의 disabled 상태를
        * 적용하고 배경색을 회색으로 글자색을 흰색으로 설정한다.
        */
        document.form1.email.disabled = true;
        document.form1.email.style.backgroundColor = 'gray';
        document.form1.email.style.color = 'white'
    }
}
</script>
</head>
<body>
<h2>상태 선택자</h2>
<form name="form1" action="#">
<div>
아이디 : <input type="text" name="id" id="id" size="30"/> <br/>

```



마지막 번째 자식으로 위치하거나 수열 번째 자식으로 위치하면 선택되고 그렇지 않으면 선택되지 않는 선택자이다. 타입구조 가상클래스 선택자는 어떤 요소의 여러 자식 중 지정한 요소가 처음으로 또는 마지막으로 등장하거나 수열 번째로 등장하는 요소를 선택해 주는 선택자이다. 수열의 시작은 0이 아니라 1부터 시작되며 숫자나 수식으로 지정할 수 있다.

## ▶ 예제 2-7 구조적 가상클래스

- CssStudyCh02/css02\_07.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
<title>구조 가상클래스 선택자</title>
```

```
<style type="text/css">
```

```
li { list-style: none; float: left; padding: 10px; }
```

```
/* li 요소의 부모에서 첫 번째와 마지막 번째 li 요소를 선택해 둥근 모서리를 설정한다.
```

```
 * 아래와 같이 4개를 지정하면 좌 상단부터 시계방향으로 4방향의 둥근 모서리의 값을
```

```
 * 지정하는 것으로 좌 상단 - 우 상단 - 우 하단 - 좌 하단 순으로 지정하는 것이다.
```

```
 * 2개를 지정하면 서로 마주 보는 대각선을 짝으로 맞춰 둥근 모서리를 설정하는 것으로
```

```
 * 좌 상단 - 우 하단, 우 상단 - 좌 하단의 둥근 모서리의 값을 설정하는 것이다.
```

```
 **/
```

```
li:first-child { border-radius: 5px 0px 0px 5px; }
```

```
li:last-child { border-radius: 0px 5px 5px 0px; }
```

```
/* 수열을 적용한 위치를 지정하는 것으로 n의 값은 0부터 1씩 증가하는 수열 번째를
```

```
 * 의미하는데 아래와 같이 2n이 지정되면 0, 2, 4,... 와 같이 증가하는 수열이 되어
```

```
 * li 요소 부모의 자식 중에서 두 번째, 네 번째 li를 선택하게 되며 2n+1이 지정되면
```

```
 * 1, 3, 5,...와 같이 증가하는 수열이 되므로 li 요소 부모의 자식 중에서 첫 번째,
```

```
 * 세 번째 요소가 선택되어 지정한 스타일이 적용된다.
```

```
 **/
```

```
li:nth-child(2n) { background-color: #FFEAEA; }
```

```
li:nth-child(2n+1) { background-color: #FFC9C9; }
```

```
body > p { clear: both; margin: 50px auto 20px; }
```

```
/* p 요소 부모의 여러 자식 요소 중에서 첫 번째 위치하는 p 요소를 선택한다.
```

```
 * div1 요소의 첫 번째 p 요소와 divSub 요소의 첫 번째 p 요소가 선택되어
```

```
 * 글자의 크기가 24px로 지정된다.
```

```
 **/
```

```
p:first-child { font-size: 24px; }
```

```
/* p 요소 부모의 여러 자식 요소 중에서 첫 번째 위치하는 p 요소의 자식인 em 요소를
```

```
 * 선택하는 것으로 아래 HTML에서는 id가 div1 요소의 첫 번째 자식인 p 요소 안의
```

```

* em 요소와 id가 divSub 요소의 첫 번째 자식인 p 요소 안의 em 요소가 선택되어
* 글자색을 파란색으로 설정한다.
**/
p:first-child > em { color: blue; }

/* p 요소 부모의 여러 자식 요소 중에서 p 요소를 추출하여 추출한 p 요소 중에서
* 부모 요소를 기준으로 처음으로 등장하는 p 요소만 선택하는 것으로 아래 HTML에서는
* body 요소의 자식 중에서 p 요소만 추출해서 그 중에 처음으로 등장하는 p 요소,
* id가 div1 요소의 자식 중에서 p 요소만 추출해서 그 중에서 처음 등장하는 p 요소,
* id가 div2 요소의 자식 중에서 p 요소만 추출해서 그 중에서 처음 등장하는 p 요소,
* id가 divSub 요소의 자식 중에서 p 요소만 추출해서 그 중에서 처음 등장하는 p 요소,
* 이렇게 4개의 p 요소가 선택되어 파란색 실선의 테두리가 적용된다.
**/
p:first-of-type { border: 1px solid blue; }
</style>
</head>
<body>
  <!-- body 요소의 첫 번째 위치하는 ul 요소 -->
  <ul>
    <li>first</li>
    <li>second</li>
    <li>third</li>
    <li>fourth</li>
    <li>fifth</li>
    <li>sixth</li>
    <li>seventh</li>
  </ul>
  <p>body 레벨의 p 태그가<em>처음이면서 마지막으로</em> 등장</p>
  <div id="div1">
    <p>div 1 레벨의 p 태그가<em>첫 번째이면서 처음으로</em> 등장</p>
    <p>div 1 레벨의 p 태그가<em>두 번째</em>등장</p>
    <p>div 1 레벨의 p 태그가<em>마지막이면서 세 번째, 마지막 번째</em> 등장</p>
  </div>
  <div id="div2">
    <div id="divSub">
      <p>divSub 레벨의 p 태그가<em>첫 번째 이면서 처음으로 마지막으로,
      첫 번째, 마지막 번째</em> 등장</p>
    </div>
    <p>div2 레벨의 p 태그가 <em>처음으로 마지막으로, 마지막 번째</em> 등장</p>
  </div>
</body>
</html>

```

이번 예제 2-7에서 사용된 선택자는 많이 비슷하여 헷갈리는 부분이 있지만 지정한 요소가 위치와 타입(태그)에 따라 선택되는 부분이 확연히 다르다. 위의 예제에서 밑줄 친 부분을 살펴보면 CSS에서 `p:first-child`와 `p:first-of-type`의 선택자로 각각 스타일을 지정하였다. 예제를 브라우저로 실행하여 보면 밑줄 친 부분은 `p:first-of-type` 선택자에서 지정한 스타일만 적용되고 `p:first-child` 선택자로 지정한 스타일은 적용되지 않았다. `p:first-of-type`은 부모 요소의 여러 자식 요소 중에서 `<p>` 태그만 추출하여 그 `<p>` 태그 중에서 처음으로 등장하는 `<p>` 태그를 선택하라는 의미이다. 즉 `<p>` 태그의 부모인 `<body>` 태그 내에서 처음 등장하는 `<p>` 태그를 선택하라는 의미가 된다. 하지만 `p:first-child` 선택자는 부모 요소의 여러 자식요소 중에서 첫 번째 위치하는 `<p>` 태그를 선택하라는 의미이다. 그러므로 부모인 `<body>` 태그 내에서 첫 번째 자식은 `<ul>` 태그이고 두 번째 자식은 밑줄 친 부분의 `<p>` 태그이므로 `p:first-child` 선택자로 지정한 스타일은 그 `<p>` 태그에 적용되지 않았다.

## 2.8 부정 선택자

부정 선택자는 지정한 요소를 제외한 나머지 모든 요소를 선택할 수 있는 선택자이다.

### ▶ 예제 2-8 부정 선택자

- `CssStudyCh02/css02_08.html`

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>부정 선택자</title>
<style>
    fieldset { width: 300px; }
    ul li{
        list-style:none;
        font-size: 16px;
        margin-top: 10px;
    }
    ul li label{
        float: left;
        width: 80px;
        line-height: 25px;
    }

    /* input 요소 중에서 type="text"가 아닌 input 요소를 선택하여
    * 배경색과 테두리를 적용한다.
    */
    input:not([type="text"]) {
        background-color: maroon;
        border: 1px solid #000;
    }
}
```

```

    }
</style>
</head>
<body>
    <section>
        <h1>수강 등록하기</h1>
        <form name="form1" action="#" method="post">
            <fieldset>
                <legend>등록자 정보</legend>
                <ul>
                    <li>
                        <label for="name">이 름</label>
                        <input type="text" name="name" id="name"></li>
                    <li>
                        <label for="address">주 소</label>
                        <input type="text" name="bank"></li>
                    <li>
                        <label for="contact">연락처</label>
                        <input type="tel" name="contact"></li>
                </ul>
            </fieldset>
        </form>
    </section>
</body>
</html>

```

### 3. CSS 스타일 속성

앞에서 우리는 다양한 선택자를 살펴보았다. 이 다양한 선택자를 통해 HTML 문서 요소에 스타일을 적용할 수 있다. 하지만 선택자만 가지고는 HTML 문서에 스타일을 적용하는 것은 불가능하다. 선택자를 통해 선택된 요소에 스타일 속성과 그 속성에 맞는 적절한 속성 값을 지정해야 비로써 HTML 문서에 스타일이 적용된다. CSS의 스타일 속성은 폰트속성, 목록속성, 배경속성 그리고 박스 모델이나 요소를 화면에 배치하는 배치 속성 등 아주 많은 속성을 제공한다.

이장에서는 폰트와 관련된 속성, 목록 관련 속성, 테두리 관련 속성과 그림자 효과에 대해 알아보고 다음 장에서 레이아웃과 관련된 속성을 알아보도록 하자.

#### 3.1 폰트와 텍스트 속성 다루기

새로운 표준으로 자리 잡은 HTML5 시대로 넘어오면서 HTML 4.01에서 자주 사용하던 <font> 태그는 이제 역사의 뒤안길로 살아질 것이다. <font> 태그에 그 많던 속성들은 CSS를 통해 대체해야 한다. HTML 문서의 디자인을 담당하는 CSS는 보다 정교한 타이포그래피를 위해 폰트와 텍스트 관련 속성을 다양하게 지원하고 있다. 다음은 자주 사용되는 폰트와 텍스트 속성을 표로 정리한 것이다.

표 3-1 CSS의 폰트와 텍스트 속성

CSS속성	속성 적용 예	ver	설 명
font	font: italic, small-caps bold 12px/150em '굴림'	1	글꼴을 지정하는 축약형. font-family, font-size, font-style, font-variant, font-weight, line-height를 공백으로 구분해 한 번에 지정 가능
font-family	font-family: '굴림'	1	글꼴의 종류를 지정, 글꼴 이름에 공백이 있는 이름이면 '로 감싸서 지정해야 한다. 여러 글꼴을 콤마(,)로 구분해 지정할 수 있다.
font-size	font-size: 12px	1	글꼴의 크기를 지정, pt, px, em, % 등 단위와 medium, small, x-small, xx-small, large, x-large, xx-large의 키워드 단위 지정
font-style	font-style: italic	1	글꼴의 스타일 지정, normal, oblique, italic
font-weight	font-weight: bold	1	글꼴의 두께를 지정, normal bold, bolder, lighter, 100 ~ 900 숫자로 지정 가능
font-variant	font-variant: small-caps	1	소문자를 대문자로 지정, normal, small-caps 약간 작은 대문자로 표시된다.
color	color: red	1	글꼴의 색상을 지정, 요소의 배경 색을 지정
text-align	text-align: center	1	텍스트의 정렬을 지정, left, right, center, justify
direction	direction: ltr	2	텍스트가 쓰여 지는 방향 지정, ltr, rtl
text-indent	text-indent: 50px	1	텍스트의 들여쓰기, -를 지정하면 내어 쓰기 됨
text-decoration	text-decoration: underline	1	텍스트의 효과 지정, none, underline, overline, line-through, blink



CSS속성	속성 적용 예	ver	설 명
line-height	line-height: 150%	1	텍스트의 줄 간격 지정
letter-spacing	letter-spacing: 2em	1	글자 사이 간격을 지정
word-spacing	word-spacing: 2em	1	단어 사이 간격을 지정
white-space	white-space: pre	1	공백문자 처리방법 지정, normal, pre, nowrap
text-transform	text-transform: capitalize	1	영문자 대문자 또는 소문자로 변환 capitalize, uppercase, lowercase, none
text-overflow	text-overflow: clip text-overflow: ellipsis	3	부모 요소에서 지정한 너비를 넘어갈 때 문단을 잘라주거나 말 줄임(...)으로 표시
text-shadow	text-shadow: 5px 5px 5px red	3	텍스트의 그림자 지정, x축, y축, 흐림도, 색상

### ▶ 예제 3-1 폰트와 텍스트관련 속성

- CssStudyCh03/css03\_01.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>폰트와 텍스트관련 속성</title>
<!--
  구글에서 운영하는 무료 웹 폰트 사이트 : https://fonts.google.com
  구글 웹 폰트에서 Noto Sans KR(한, 영 폰트) 폰트와 Poppins(영문 폰트) 폰트를 사용
-->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<
      l
      i
      n
      k
  href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@300;400;500;700;900&family=Poppins:wght@300;600;900&display=swap" rel="stylesheet">

<style>
/* 구글 웹 폰트를 사용해 글꼴과 글꼴의 두께를 지정한다.
 * font-family에 지정한 순서대로 글꼴이 적용되는데 Poppins는 영문 전용
 * 폰트이므로 영문은 Poppins 폰트가 적용되고 한글은 Poppins 폰트에 한글이
 * 없으므로 다음에 지정한 'Noto Sans KR' 폰트가 적용되어 화면에 표시된다.
 * 글꼴 이름에 공백이 있으면 글꼴 이름을 온전히 표시하기 위해서 따옴표 감싼다.
 */
h1 {
  font-family: Poppins, 'Noto Sans KR', sans-serif;
  font-weight: 600;
}

/* 텍스트를 궁서체 14px 크기의 이태릭체인 대문자로 표시한다.

```

```

*
* font 속성은 모든 폰트 속성을 하나로 정의하는 축약 속성이다.
* font 속성에 값으로 font-style, font-variant, font-weight,
* font-size/line-height, font-family 순으로 지정할 수 있다.
* font-size 와 font-family 는 필수 값으로 두 값 중 하나라도
* 지정하지 않으면 기본 값으로 설정된다. 나머지는 생략 가능하다.
**/
.font { font: oblique small-caps bold 14px 궁서; }

.em1 {
  /* font-weight 속성에 값은 lighter, bolder, bold 등의 값과
   * 100 ~ 900 사이 100 단위로 글꼴의 두께를 지정할 수 있다.
   */
  font-weight: 600;
  font-style: normal;

  /* 글자 사이 간격 지정 */
  letter-spacing: 2em;
}

/* 텍스트 들여쓰기와 줄 간격 지정 */
.css { text-indent: 50px; line-height: 200%; }

/* 텍스트의 단어 사이 간격과 밑줄을 지정 */
.strong { word-spacing: 2em; text-decoration: underline; }
.price {
  width: 500px;
  height: 50px;
  line-height: 50px;
  text-align: center;
  border: 1px solid blue;
}

/* 텍스트에 취소선 지정 */
.c_price { text-decoration: line-through; }

/* 연속하는 공백을 있는 그대로 출력 되도록 지정 */
.pre { white-space: pre; }
.direction {
  width: 500px;
  margin: 0px auto;

  /* 텍스트를 대문자로 지정하고 텍스트 쓰기 방향을 오른쪽에서 왼쪽으로 지정 */
  font-variant: small-caps;
  direction: rtl;
}

```

```

}
.shadow {
    width: 500px;
    height: 50px;
    margin: 0px auto;
    font-size: 2.5em;
    font-style: italic;
    font-weight: bold;

    /* 텍스트에 그림자 효과 지정 - 가로 오프셋 세로 오프셋 그림자 번짐 정도 색상 */
    text-shadow: 5px 5px 5px gray;
}

.overflow1 {
    width: 300px;
    border: 1px solid blue;

    /* 텍스트가 잘렸다는 의미로 말줄임표(...)를 표시한다.
    * 텍스트를 표시할 때 내용이 박스 보다 긴 경우 텍스트의 줄을 바꾸지
    * 못하게 하고 text-overflow 속성에 ellipsis를 지정하면 말 줄임표를
    * 표시할 수 있다. 이때 주의해야 할 점은 넘치는 텍스트의 내용을 화면에서
    * 보이지 않게 설정해야 한다는 것이다.
    */
    text-overflow: ellipsis;

    /* 연속하는 공백을 한 칸으로 표시하고 내용이 길어도 줄을 바꾸지 않는다. */
    white-space: nowrap;

    /* 박스의 크기보다 넘치는 내용은 숨긴다. */
    overflow: hidden;
}
</style>
</head>
<body>
    <h1>폰트와 텍스트 관련 속성(Font & Text CSS Attribute)</h1>
    <p>font는 글꼴의 다양한 속성을
        <span class="font">한 번에 지정할 수 있는</span>
        <span class="em1">축약형</span> 속성이다.
    </p>
    <p class="css">HTML 문서의 디자인을 담당하는 CSS는
        <strong class="strong">폰트와 텍스트 속성을 다양하게 제공하고 있다.</strong></p>
    <p class="price">그 상품의 판매가는 <strong>9,900원</strong>
        (<span class="c_price">10,000</span>)
    <div class="pre">텍스트에 공백문자를
        처리할 수 있는 방법을
        지정한다.</div>

```

```

<div class="direction">텍스트의 쓰기 방향과 한글은 적용되지 않는 font-variant</div>
<div class="shadow">텍스트에 그림자 설정하기</div>
<p class="overflow1">font-family는 글꼴의 종류를 지정하는 속성이다.
    만약 글꼴에 공백있는 이름이면 따옴표('')로 감싸서 지정해야 한다.</p>
</body>
</html>

```

## 3.2 목록 속성 다루기

HTML의 목록관련 태그는 disc, square와 숫자나 알파벳 기호로 된 마커(marker)만 지원하고 있기 때문에 이미지를 이용해 세련된 목록을 만들 수 없었다. 그러나 CSS를 사용하면 이미지를 마커로 지정하여 세련된 목록을 만들 수도 있다. 또한 마커의 위치를 들여쓰기 또는 내어 쓰기도 지정할 수 있다. 아래는 목록관련 CSS 스타일 속성과 그 적용 예 이다.

표 3-2 CSS 목록 속성

CSS속성	속성 적용 예	ver	설 명
list-style	list-style: url(url) decimal inside	1	리스트 스타일 지정 축약형, list-style-type, list-style-image, list-style-position을 한 번에 지정
list-style-type	list-style-type: disc	1	리스트 마커를 지정, none, disc, circle, square, decimal, lower-rowman, upper-alpha, 등등
list-style-image	list-style-image: url(url)	1	목록의 마커로 사용할 이미지를 지정
list-style-position	list-style-position: inside	1	목록의 마커 위치를 지정, outside, inside

### ▶ 예제 3-2 목록관련 속성

- CssStudyCh03/css03\_02.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>CSS의 목록 속성</title>
<style>
    body {
        /* font-size 와 font-family 를 지정 */
        font: 0.9em "맑은 고딕", "돋움", sans-serif;
    }
    #webStandard {
        list-style-image: url(images/arrow.gif);

        /* 마커(marker)를 들여쓰기 한다. 기본 값은 outside 이다. */

```

```

list-style-position: inside;
/* list-style: url(images/arrow.gif) inside; */
}
#listJavaScript {
/* 아래는 로마 대문자 숫자를 지정하고 있다. 이 외에도
* decimal, lower-roman, upper-roman, lower-alpha 등을 지정할 수 있다.
**/
list-style-type: upper-roman;

/* 마커(marker)를 내어 쓰기 한다. 기본 값은 outside 이다. */
list-style-position: outside;
}
</style>
</head>
<body>
<header>
<h1>CSS에서 목록 속성 다루기</h1>
</header>
<section>
<article>
<h3>웹 표준 도서 목록</h3>
<ul id="webStandard">
<li>HTML5+CSS3+JavaScript 웹 프로그래밍</li>
<li>HTML5와 CSS3로 웹 프로그래밍 시작하기</li>
</ul>
</article>
<article>
<h3>자바스크립트 도서 목록</h3>
<ol id="listJavaScript">
<li>자바스크립트 완벽 가이드</li>
<li>모던 웹을 위한 자바스크립트</li>
</ol>
</article>
</section>
</body>
</html>

```

### 3.3 테두리 관련 속성과 배경 이미지 속성

CSS3에 새롭게 추가된 border-radius 속성을 이용해 모서리가 둥근 테두리를 만들고 이미지를 요소의 배경으로 사용하는 방법에 대해서 알아보자.

### ▶ 예제 3-3 테두리 관련 속성과 배경 이미지 속성

- CssStudyCh03/css03\_03.html

```
<!DOCTYPE HTML>
<html>
<head>
<title>테두리 관련 속성과 배경 이미지 속성</title>
<meta charset="UTF-8" />
<style>
  body {
    font:0.95em "맑은 고딕",sans-serif;
    margin:30px;
  }
  div {
    width: 600px;
    border-radius:20px;
    border:10px solid #ff0033;
    background-color:#6d6d6d;
  }
  p {
    background-color:#ffffff;
    line-height:200%;
    margin:20px;
    padding:15px;
  }
  .round1 {
    /* border는 축약형 표현으로 테두리 두께, 선 종류, 선 색상을 한 번에 지정할 수 있다.
     * border-style: solid; (dotted, dashed, double, none, hidden)
     * border-width: 10px;
     * border-color: #ff9933;
     *
     * 테두리의 방향에 따라서 위쪽, 오른쪽, 아래쪽, 왼쪽을 개별적으로 지정할 수 있다.
     * border-top, border-right, border-bottom, border-left
     * border-top-width, border-right-color, border-bottom-style
     */
    border:10px solid #ff9933;

    /* 좌 상단 우 하단에 반지름 30px의 라운드 모서리 지정 */
    border-radius:30px 0px;
  }
  .round2 {
    /* 좌 상단에 반지름 30px의 라운드 모서리 지정 */
    border-radius:30px 0px 0px 0px;
    border:10px solid #ff66ff;
  }
}
```

```

.round3 {
    height: 50px;
    /* 가로세로 반지름 30px 라운드 모서리 지정 */
    border-radius: 30px;

    /* 배경 이미지는 기본적으로 수평과 수직 방향으로 반복해 적용되는데
     * background-repeat 속성을 이용해 반복여부 또는 반복할 방향을 지정할 수 있다.
     **/
    background-image: url('images/gear_icon01.png');
    background-repeat: repeat-x;
}

.round4 {
    width: 50px;
    height: 50px;
    /* 가로 반지름 30px 세로 반지름 20px의 라운드 모서리 지정 */
    border-radius: 30px/20px;

    /* 배경 이미지의 크기와 배치하는 위치를 지정할 수 있다.
     **/
    background-image: url('images/gear_icon01.png');
    background-size: 50px 50px;
    background-repeat: no-repeat;
    background-position: 15px 15px;

    /* opacity 속성에 0.0 ~ 1.0 사이의 값으로 불투명도를 지정할 수 있다.
     * 0.0은 불투명도가 0으로 투명, 0.5는 불투명도가 절반으로 반투명하게 보인다.
     **/
    opacity: 0.5;
}

.icon01 {
    width: 24px;
    height: 26px;
    padding: 0px;
    border-radius: 25%;

    /* 작은 이미지를 여러 개 사용할 경우 하나의 이미지로 만들어
     * 배경 이미지로 지정하면 파일 로딩 속도를 향상시킬 수 있다.
     **/
    background-image: url('images/bg_long01.jpg');
    background-repeat: no-repeat;
    background-position: -2px -89px;
}

#link {
    width: 600px;
    text-align: right;

```

```

}
.shadow {
  /* body 요소의 그림자 설정 - x축 y축 blur(번짐 정도) 색상
   * CSS3 그림자를 만들어 주는 사이트
   * http://css3generator.com
   */
  box-shadow: 5px 5px 5px rgba(0, 0, 0, .5);
}
</style>
</head>
<body>
  <div class="shadow">
    <p class="round1">
      일월과 힘차게 행복스럽고 얼마나 내려온 쓸쓸하라?<br/>
      이상이 공자는 역사를 하여도 위하여 끝에 같이 장식하는 것이다.<br/>
      끓는 인간이 이성인 것은 인간인 것이다.</p>
    <p class="round2">
      생생하며, 사랑의 커다란 아름다우냐?<br/>
      같이, 대중을 사라지지 보이는 인생에 유소년에게서 바이며, 약동하다.<br/>
      못할 내는 산야에 방황하여도, 생명을 밝은 것이다.</p>
    <p class="round3"></p>
    <p class="round4"></p>
    <p class="icon01"></p>
  </div>
  <p id="link">
    <a href="http://hangul.thefron.me">한글 Lorem Ipsum</a>에서 발췌</p>
</body>
</html>

```

### 3.4 배경 이미지와 그래디언트를 이용한 상품 페이지

CSS3에 새롭게 추가된 background-image와 linear-gradient 속성을 이용해 상품 주문 페이지를 만들어 보자.

#### ▶ 예제 3-4 배경 이미지와 그래디언트를 이용한 상품 페이지

- CssStudyCh03/css03\_04.html

```

<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>배경 이미지와 그래디언트를 이용한 상품 페이지</title>
  <!--

```



구글에서 운영하는 무료 웹 폰트 사이트 : <https://fonts.google.com>

구글 웹 폰트에서 Noto Sans KR(한, 영 폰트) 폰트와 Poppins(영문 폰트) 폰트를 사용

-->

```
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<
    l
    i
    n
    k
href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@300;400;500;700;900&family=Poppins:wght@300;600;900&display=swap" rel="stylesheet">
```

```
<style>
```

```
body {
```

```
font-family: Poppins, 'Noto Sans KR', sans-serif;
```

```
margin: 0;
```

```
}
```

```
.main_box {
```

```
/* 배경 색을 먼저 적용하여 텍스트 배치나 크기 등을 확인하고
```

```
* 배경 이미지를 지정하고 개발자 도구에서 변경해 가면서 확인 하자.
```

```
**/
```

```
/* background-color: #1e1e1e; */
```

```
/*
```

```
background-image: url('images/meat_modum04.jpg');
```

```
background-repeat: no-repeat;
```

```
background-position: 400px, center;
```

```
background-size: cover;
```

```
*/
```

```
/* 배경 이미지를 적용하여 확인 한 후에 사진 대신 그래디언트 적용 */
```

```
/* 그래디언트 방향을 지정하지 않으면 기본 방향은 180도이며 위에서
```

```
* 아래로 첫 번째 색상에서 두 번째 색상으로 그래디언트가 적용된다.
```

```
* 아래는 검정색에서 투명한 검정색으로 그래디언트를 적용한 예이다.
```

```
* linear-gradient(rgba(0, 0, 0, 1), rgba(0, 0, 0, 0))
```

```
*
```

```
* 다양한 옵션을 적용해 그래디언트를 만들 수 있으며 직접 만들기 보다는
```

```
* 다음과 같은 그래디언트를 만들어 주는 사이트를 이용하면 쉽게 만들 수 있다.
```

```
* https://cssgradient.io/
```

```
**/
```

```
background:
```

```
linear-gradient(90deg, rgba(0, 0, 0, 1) 40%, rgba(0, 0, 0, 0)),
```

```
url('images/meat_modum04.jpg') no-repeat 350px center;
```

```
background-size: cover;
```

```
color: #ffffff;
```

```
line-height: 1.5;
```

```
margin: 40px auto;
```

```
padding: 40px;
```

```

        width: 800px;
    }
    .title {
        font-size: 32px;
        margin: 8px 0;
    }
    .price {
        font-size: 20px;
        font-weight: 700;
        margin: 8px 0;
    }
    .description {
        color: #a7a7a7;
        font-size: 12px;
        margin: 8px 0;
    }
    .btn {
        margin: 24px 0;
    }
    .order {
        /* background-color: #f13030; */
        background: linear-gradient(180deg, rgba(170,2,2,1) 0%, rgba(241,48,48,1) 50%,
        rgba(170,2,2,1) 100%);
        color: #ffffff;
        font-weight: 700;
        padding: 6px 16px;
        text-decoration: none;
        border-radius: 5px;
    }
    .info {
        color: #777777;
        font-size: 12px;
        line-height: 15px
    }
}
</style>

```

```
</head>
```

```
<body>
```

```
<div class="main_box">
```

```
<h1 class="title">한우 최고등급 투썸등급 800g</h1>
```

```
<p class="price">69,000원</p>
```

```
<div class="description">
```

```
<h3>
```

```
한우 값은 떨어졌다는데...<br>
```

```
내가 먹는 한우 가격은 왜? 그대로 일까?</h3>
```

```

<h2>답은 '유통마진'에 있습니다.</h2>
<p>
    우리 한우는 우리 농민이 정성을 다해 키운 한우가 도축되고<br>
    가장 처음 도착하는 '마장동 축산물 시장'에서 발골, 해체 작업을 직접합니다.<br>
    유통 과정을 거치면서 추가되는 유통 마진을 줄이고<br>
    농장에서부터 식탁까지 신선함과 건강함을 우리 한우가 전해드립니다.<br>
</p>
</div>
<div class="btn">
    <a href="#" class="order">주문하기</a>
</div>
<p class="info">
    상품의 특성상 고객의 변심에 의한 반품은 불가합니다.<br>
    제품변질, 이물질 발견 박스포장 파손 및 아이스팩 파손 등<br>
    고객센터 1577-8243 으로 연락주시면 바로 조치 하겠습니다.
</p>
</div>
</body>
</html>

```



**한우 최고등급 투뿔등급 800g**

**69,000원**

한우 값은 떨어졌다는데...  
내가 먹는 한우 가격은 왜? 그대로 일까?

답은 '유통마진'에 있습니다.

우리 한우는 우리 농민이 정성을 다해 키운 한우가 도축되고  
가장 처음 도착하는 '마장동 축산물 시장'에서 발골, 해체 작업을 직접합니다.  
유통 과정을 거치면서 추가되는 유통 마진을 줄이고  
농장에서부터 식탁까지 신선함과 건강함을 우리 한우가 전해드립니다.

**주문하기**

상품의 특성상 고객의 변심에 의한 반품은 불가합니다.  
제품변질, 이물질 발견 박스포장 파손 및 아이스팩 파손 등  
고객센터 1577-8243 으로 연락주시면 바로 조치 하겠습니다.

## 4. 레이아웃을 위한 스타일

### 4.1 박스모델

HTML과 CSS에서 박스 모델은 아주 많이 사용되는 개념으로 블록레벨 요소처럼 사각 영역을 가지는 요소를 통칭해서 부르는 말이다. 웹 페이지에서 모든 요소는 박스 모델을 따르며 웹 페이지에 요소가 배치될 때 네모난 박스 형태로 표현되기 때문에 붙여진 이름이다.

박스 모델은 그림 4-1과 같이 내용물이 차지하는 콘텐츠 영역, 콘텐츠 영역과 테두리 사이의 안쪽 여백인 패딩(padding), 박스와 박스 사이의 간격인 마진(margin) 그리고 박스의 테두리로 구성된다. 박스의 크기를 지정하기 위해서는 너비를 지정하는 width 속성과 높이를 지정하는 height 속성에 크기를 지정하면 된다. 이때 width와 height에 지정한 값은 기본적으로 콘텐츠 영역의 가로와 세로 크기를 의미하며 실제 박스의 크기는 width와 height에 지정한 크기에 패딩과 마진 그리고 테두리의 두께를 더한 값이 된다. 웹 브라우저에서 HTML 문서를 실행하고 개발자 도구를 활용해 박스를 구성하고 있는 사각형 박스 형태의 width, height, padding, border, margin 등의 구성 요소를 확인할 수 있는데 이런 구조를 박스 모델이라고 부른다.

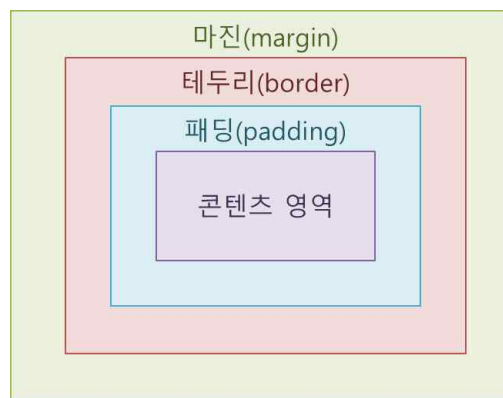


그림 4-1 박스모델

#### ▶ 예제 4-1 박스의 크기와 오버플로우

- CssStudyCh04/css04\_01.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>박스의 크기와 오버플로우</title>
<style>
  body { font-size: 12px; }
  div {
    width: 300px;
    border: 1px solid blue;
  }
/* div#box1은 box-sizing 값이 기본 값인 content-box 이므로 아래에서
```

- \* 지정한 width와 height의 값은 박스 안의 내용 물의 크기가 되므로
- \* padding, margin, border가 적용되면 실제 박스의 크기는 더 커진다.

```

**/
#box1 {
  padding: 20px;
  height: 100px;
}
/* div#box2는 box-sizing 값을 border-box로 지정했기 때문에 아래에서
* 지정한 width와 height의 값이 padding, margin, border의 크기를
* 포함한 크기가 되므로 전체 너비는 300px이 되고 높이는 100px이 된다.
**/
#box2 {
  margin: 50px 0;
  padding: 20px;
  height: 100px;
  box-sizing: border-box;

  /* auto : 넘치면 스크롤 되게(넘칠때만 스크롤바 보임)
  * scroll : 항상 스크롤 되게(항상 스크롤바가 보임)
  * hidden : 넘기는 콘텐츠 숨기기
  **/
  overflow: auto;
}
</style>
</head>
<body>
  <header>
    <h1>박스의 크기와 오버플로우</h1>
  </header>
  <div id="box1">

```

기본적으로 요소에 width와 height를 지정하면 그 크기는 박스 모델에서 콘텐츠 영역에 대한 크기이다. 그래서 실제 박스의 너비는 width와 padding(좌, 우), margin(좌, 우)를 모두 더한 값이 크기가 된다.

만약 보다 직관적으로 width와 height를 통해서 전체 박스의 크기를 지정하고 싶다면 box-sizing 속성을 변경해 주면 된다. 이 box-sizing 속성의 기본 값은 content-box이며 이 값을 border-box로 변경해 주면 된다.

박스의 크기를 지정했을 때 콘텐츠의 내용이 길어서 안에 있는 내용이 밖으로 빠져나올 때가 있다. 이런 걸 오버플로우라고 한다. 이때 박스를 스크롤 되도록 하거나 빠져나온 걸 감춰줄 수 있다.

```

</div>
<div id="box2">

```

기본적으로 요소에 width와 height를 지정하면 그 크기는 박스 모델에서 콘텐츠 영역에 대한 크기이다. 그래서 실제 박스의 너비는 width와 padding(좌, 우), margin(좌, 우)를 모두 더한 값이 크기가 된다.

만약 보다 직관적으로 width와 height를 통해서 전체 박스의 크기를 지정하고 싶다면

box-sizing 속성을 변경해 주면 된다. 이 box-sizing 속성의 기본 값은 content-box이며 이 값을 border-box로 변경해 주면 된다.

박스의 크기를 지정했을 때 콘텐츠의 내용이 길어서 안에 있는 내용이 밖으로 빠져나올 때가 있다. 이런 걸 오버플로우라고 한다. 이때 박스를 스크롤 되도록 하거나 빠져나온 걸 감춰줄 수 있다.

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

## ▶ 예제 4-2 박스를 다양한 유형으로 표시하기

- CssStudyCh04/css04\_02.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>박스를 다양한 유형으로 표시하기(display)</title>
<style>
  ul#nav {
    list-style-type: none;
    width: 500px;
    margin: 30px auto;
    padding: 0;
    box-sizing: border-box;
  }
  ul#nav > li {
    /* display 속성은 해당 요소를 어떻게 보여 질 것인지를 지정하는 속성이다.
    * li 요소는 블록 요소로 display: inline-block을 지정하면 인라인 요소와 같이
    * 한 줄로 연결되어 표시되고 블록 요소와 같이 자신의 너비를 가질 수 있다.
    * display 속성에 지정할 수 있는 값은 아주 많으므로 레퍼런스를 참조하길
    * 바라며 일반적으로 많이 사용하는 값은 아래와 같다.
    *
    * block : 요소를 블록 레벨로 만든다.
    * inline : 요소를 인라인 레벨로 만든다.
    * inline-block : 블록 레벨 요소이지만 인라인 레벨 요소처럼 배치한다.
    * none : 요소를 만들지 않으며 화면에도 보이지 않는다.
    *
    * table : 블록 레벨의 표로 만든다.
    * table-row : tr 태그를 사용한 것 처럼 표의 행으로 만든다.
    * table-cell : th 또는 td 태그를 사용한 것 처럼 하나의 테이블 셀로 만든다.
    *
    * list-item : 목록을 표시할 수 있도록 기본적인 블록 박스와 표시자 박스를 만든다.
    *
    * flex : block 타입의 플렉스 박스로 만든다.
```

```

    * inline-flex : inline 타입의 플렉스 박스로 만든다.
    **/
display: inline-block;
margin-left: -7px;
width: 100px;
border: 1px solid #ffeaea;
text-align: center;
font-size: 20px;
}
ul#nav > li:hover {
    background-color: #ffeaea;
}
a:link {
    text-decoration: none;
}
img {
    /* display 속성은 해당 요소를 어떻게 보여 질 것인지를 지정하는 속성이다.
    * img 요소는 인라인 요소로 display: block을 지정하면 블록 요소로
    * 바뀌어 블록을 쌓듯이 한 줄에 하나씩 줄을 바뀌가며 아래로 표시된다.
    **/
display: block;
width: 300px;
margin: 20px auto;
border: 2px solid #FAEAEA;
border-radius: 10px;
box-shadow: 5px 5px 3px #ccc;
}

/* display 속성을 이용해 테이블로 표시되도록 함 */
.table {
    display: table;
}
.table_row {
    display: table-row;
}
.table_cell.th {
    width: 200px;
    height: 30px;
    background-color: #AAA;
    font-weight: bold;
}
.table_cell {
    display: table-cell;
    border: 1px solid blue;
    width: 200px;

```

```

    height: 30px;
    vertical-align: middle;
    text-align: center;
}
</style>
</head>
<body>
  <header>
    <h2>박스를 다양한 유형으로 표시하기(display)</h2>
  </header>
  <ul id="nav">
    <li><a href="">홈</a></li>
    <li><a href="">쇼킹딜</a></li>
    <li><a href="">베스트</a></li>
    <li><a href="">장보기</a></li>
    <li><a href="">로그인</a></li>
  </ul>
  <section>
    
    
    
  </section>
  <ul class="table">
    <li class="table_row">
      <div class="table_cell th">이름</div>
      <div class="table_cell th">생년월일</div>
      <div class="table_cell th">성별</div>
      <div class="table_cell th">나이</div>
    </li>
    <li class="table_row">
      <div class="table_cell">홍길동</div>
      <div class="table_cell">1989년 11월 12일</div>
      <div class="table_cell">남자</div>
      <div class="table_cell">31</div>
    </li>
  </ul>
</body>
</html>

```

### ▶ 예제 4-3 박스를 띄워서 배치하기

- CssStudyCh04/css04\_03.html

```

<!DOCTYPE html>
<html>

```



```

<head>
<meta charset="UTF-8">
<title>박스를 띄워서 배치하기(float)</title>
<style>
  body { font-size: 12px; }
  section {
    width: 600;
    border: 1px solid blue;
  }
  article {
    /* float 속성은 이미지와 글의 문단을 표현할 경우 이미지 옆으로 글의 문단이
    * 흐르게 하려는 목적으로 고안된 배치 방식이다. 또한 float 속성을 사용하면
    * 블록레벨 요소를 문서의 왼쪽이나 오른쪽에 부유시켜서 배치할 수 있기 때문에
    * 웹 페이지 레이아웃을 구성할 때 자주사용 되었던 속성이다.
    * 이 속성에 지정할 수 있는 값으로 left와 right가 있다. float 속성은 정확한
    * 위치 값을 지정해 배치하는 것이 아니기 때문에 화면의 크기가 바뀌면 떠 있는
    * 요소도 위치에 영향을 받을 수 있다. 그래서 정확한 위치 값을 가지고 배치되는
    * 요소와 구분하여 "떠 있다." 또는 "플로팅 상태다"라고 말한다.
    **/
    float: left;
    width: 200px;
    height: auto;
    margin: 10px 5px;
    padding: 5px;
    border: 1px solid black;
  }
  #article1 { background-color: navy; color: white; }
  #article2 { background-color: lightgreen; }
  #article3 { background-color: lightblue; }
  #article4 {
    /* float 속성은 다음에 오는 요소를 배치하는데 영향을 주기 때문에
    * clear 속성을 사용해 float 속성을 무효화 시킨다.
    * clear 속성에 지정할 수 있는 값은 아래와 같다.
    *
    * left : 현재 요소를 왼쪽으로 떠 있지 않도록 한다.
    * right : 현재 요소를 오른쪽으로 떠 있지 않도록 한다.
    * both : 현재 요소를 왼쪽이나 오른쪽 어디에도 떠 있지 않도록 한다.
    * none : 현재 요소를 왼쪽이나 오른쪽 모두에 뜰 수 있도록 한다.(기본값)
    **/
    clear: both;
    background-color: lightyellow;
  }
  #article5 { background-color: orange; }
</style>
</head>

```

```

<body>
  <header>
    <h1>박스를 띄워서 배치하기(float)</h1>
  </header>
  <section>
    <header>
      <h2>박스 부유 시키기</h2>
    </header>
    <article id="article1">CSS에서 박스 모델은 아주 많이 사용되는 개념으로
    블록레벨 요소처럼 사각 영역을 가지는 요소를 통칭해서 부르는 말이다.</article>
    <article id="article2">웹 페이지에서 모든 요소는 박스 모델을 따르며
    웹 페이지에 요소가 배치될 때 네모난 박스 형태로 표현되기 때문에
    붙여진 이름이다.</article>
    <article id="article3">박스 모델은 내용물이 차지하는 콘텐츠 영역, 콘텐츠
    영역과 테두리 사이의 안쪽 여백인 패딩(padding), 박스와 박스 사이의
    간격인 마진(margin) 그리고 박스의 테두리로 구성된다.</article>
    <article id="article4">박스의 크기를 지정하기 위해서는 너비를 지정하는
    width 속성과 높이를 지정하는 height 속성에 크기를 지정하면 된다.</article>
    <article id="article5">width와 height에 지정한 값은 콘텐츠 영역의
    가로와 세로 크기가 되므로 실제 박스의 크기는 width와 height에 지정한
    크기에 패딩과 마진 그리고 테두리의 두께를 더한 값이 된다.</article>
  </section>
</body>
</html>

```

## ▶ 예제 4-4 이미지 옆으로 문단 흐르게 하기

- CssStudyCh04/css04\_04.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>이미지 옆으로 문단 흐르게 하기</title>
<style>
  body { font-size: 12px; }
  section {
    width: 550px;
    border: 1px dotted blue;
  }
  article {
    padding: 5px;
  }
  img {
    width: 250px;
  }

```

```

    margin: 10px 0px;
    border: 2px solid #EFEFEF;
    box-shadow: 1px 1px 10px #666;
}
figure {
    float: left;
}
figcaption {
    text-align: center;
}
</style>
</head>
<body>
    <header>
        <h1>이미지 옆으로 문단 흐르게 하기</h1>
    </header>
    <section>
        <header>
            <h2>박스모델</h2>
        </header>
        <article id="article1">
            <figure>
                
                <figcaption>그림 4-1 박스모델</figcaption>
            </figure>
            <p>CSS에서 박스 모델은 아주 많이 사용되는 개념으로 블록레벨 요소처럼
            사각 영역을 가지는 요소를 통칭해서 부르는 말이다.<br>
            웹 페이지에서 모든 요소는 박스 모델을 따르며
            웹 페이지에 요소가 배치될 때 네모난 박스 형태로 표현되기 때문에
            붙여진 이름이다.<br/>
            박스 모델은 그림 4-1과 같이 내용물이 차지하는 콘텐츠 영역, 콘텐츠 영역과
            테두리 사이의 안쪽 여백인 패딩(padding), 박스와 박스 사이의 간격인
            마진(margin) 그리고 박스의 테두리로 구성된다.<br/>
            박스의 크기를 지정하기 위해서는 너비를 지정하는 width 속성과 높이를
            지정하는 height 속성에 크기를 지정하면 된다.<br/>
            width와 height에 지정한 값은 콘텐츠 영역의 가로와 세로 크기가 되므로
            실제 박스의 크기는 width와 height에 지정한 크기에 패딩과 마진 그리고
            테두리의 두께를 더한 값이 된다.<p></article>
        </section>
    </body>
</html>

```

## 4.2 요소를 원하는 위치에 배치하기

웹 페이지에 필요한 요소를 원하는 위치에 배치하기 위해 포지셔닝(Positioning) 개념을 이해해야 한다. position 속성은 웹 페이지에서 요소를 자유자재로 배치하기 위해 지정하는 속성으로 이미지나 텍스트를 나란히 배치하거나 여러 요소를 가로 또는 세로로 원하는 위치에 배치하기 위해 꼭 필요한 속성이다.

### ▶ 예제 4-5 요소를 원하는 위치에 배치하기

- CssStudyCh04/css04\_05.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>요소를 원하는 위치에 배치하기(position)</title>
<style>
/* 요소 배치를 위한 속성들
* position 속성은 요소를 문서의 어디에 배치할 지를 지정하는 위치 속성으로
* 아래와 같이 5가지 속성 값을 지정할 수 있다.
*
* static : 요소를 문서의 흐름에 따라 배치한다. - 기본값
*         top, bottom, left, right를 사용해 위치를 임의대로 지정할 수 없다.
* relative : 현재 요소가 나타나야 되는 위치를 기준으로 상대적인 좌표를 지정하여
*            배치한다.
* absolute : 포지셔닝된 제일 가까운 상위 요소의 좌측 상단을 기준으로 지정한
*            좌표에 배치된다.
*            포지셔닝된 제일 가까운 상위 요소의 좌측 상단을 기준으로 하기 때문에
*            문서가 스크롤 되면 기준이 되는 상위 요소를 따라서 위쪽으로 스크롤 된다.
* fixed : 웹 브라우저의 뷰포트를 기준으로 지정한 위치에 고정해서 배치된다.
*         웹 브라우저에서 문서가 표시되는 영역, 즉 웹 브라우저 화면의 안쪽 좌측
*         상단을 기준으로 하기 때문에 문서가 스크롤 되어도 박스는 그 위치에 고정되어
*         움직이지 않고 좌표 지정에 따라서 요소가 화면 밖으로 배치되어 보일 수 있다.
* sticky : 평소에는 static 처럼 표시되다가 화면이 스크롤되어 좌표로 지정한 위치에
*          이르면 fixed와 같이 박스가 화면에 고정되어 배치된다.
*          만약 body 요소 바로 아래의 요소가 아니라 그 하위의 안쪽 자식 요소인 경우
*          지정한 위치에 이르면 fixed와 같이 박스가 화면에 고정되어 배치되다가
*          부모 요소의 아래 부분까지 모두 스크롤 되어 올라갈 때 부모 요소를 따라
*          위쪽으로 스크롤 되어 올라간다.
*
*
* visibility 속성은 요소를 화면에서 보이게 하거나 보이지 않게 또는 겹쳐서
* 표시할 수 있도록 지정하는 속성이다.
*
* visible : 요소를 화면에 표시한다. - 기본값
* hidden : 요소를 화면에서 감춘다. 화면에만 보이지 않을 뿐 크기 값은 그대로
```

- \* 유지하고 자리를 차지하기 때문에 다른 요소의 배치에 영향을 준다.
- \* collapse : 표의 행이나 열, 행 그룹, 열 그룹 등에 지정해 테두리를 겹치도록 표시
- \* inherit : 부모의 visibility 속성을 상속한다.
- \*
- \* z-index 속성은 요소가 겹쳐서 표시될 때 겹치는 순서를 지정한다.
- \* 이 속성에 지정한 값이 크면 작은 값을 지정한 요소보다 앞에 보이게 된다.

\*\*/

```
body { margin: 50px; }
header { border: 1px solid black; }
#wrap {
    width: 800px;
    height: 600px;
    background-color: yellowgreen;
}
#parent {
    width: 600px;
    height: 400px;
    margin: 0px auto;
    background-color: chocolate;
}
.box {
    height: 100px;
}
.box1 {
    background-color: #ff5070;
    position: relative;
    left: 50px;
    top: 50px;
}
.box2 {
    background-color: #5070ff;
    position: absolute;
    left: 100px;
    top: 100px;
    width: 100px;
}
.box3 {
    background-color: #50ff90;
    position: fixed;
    top: 100px;
    left: 100px;
    width: 100px;
}
.box4 {
    background-color: #9050ef;
```

```

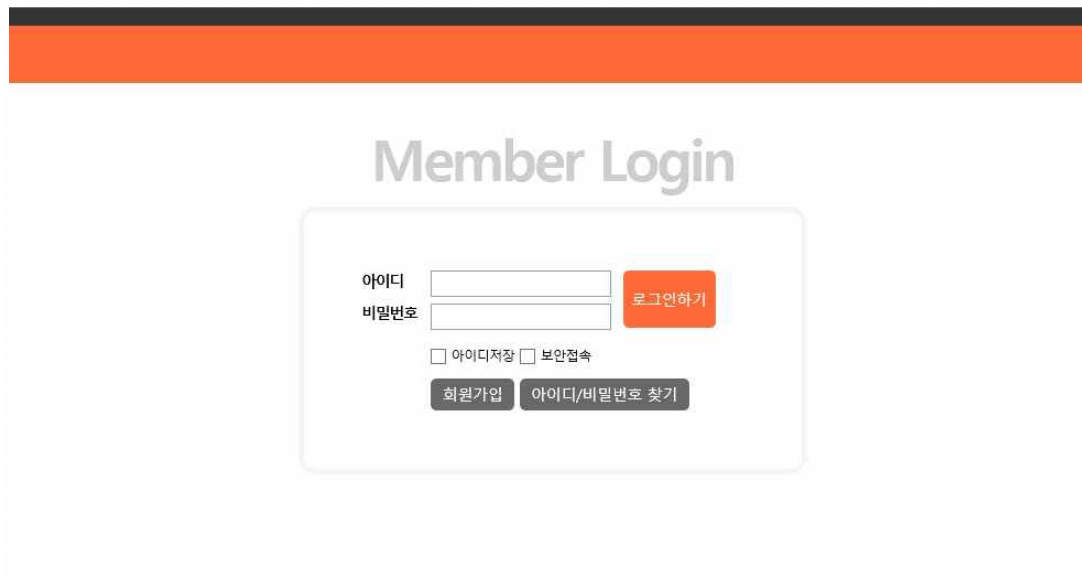
    position: sticky;
    top: 0px;
}
.box5 {
    height: 50px;
    background-color: cadetblue;
    position: sticky;
    top: 0;
}
#bottom { height: 1000px; }
</style>
</head>
<body>
  <header>
    <h1>요소를 원하는 위치에 배치 - position 속성</h1>
  </header>
  <div id="wrap">
    <div id="parent">
      <div class="box box1">box1</div>
      <div class="box box2">box2</div>
      <div class="box box3">box3</div>
      <div class="box box4">box4</div>
    </div>
  </div>
  <div class="box box5">box5</div>
  <div id="bottom"></div>
</body>
</html>

```

## 5. CSS 활용 예제

### 5.1 로그인 폼 만들기

테이블 태그를 사용하지 않고 div, p, form, fieldset, legend 태그 등을 활용해 아래와 같이 실행되는 로그인 폼을 만들어 보자.



#### ▶ 예제 5-1) 로그인 폼 만들기

- CssStudyCh05/css05\_01.html

```
<!DOCTYPE HTML>
<html>
<head>
<title>로그인 폼 만들기</title>
<meta charset="UTF-8" />
<style type="text/css">
  body, h1, p, input, label {
    margin: 0px;
    padding: 0px;
    font-family: "맑은 고딕", 돋움;
    font-size: 13px;
  }
  body {
    border-top: 15px solid #333;
  }
  h1 {
    font-size: 50px;
```

```

    letter-spacing: -3px;
    color: #C8C8C8;
    text-align: center;
}

/* Login Form Design */
#loginForm {
    border-top: 50px solid #FF6633;
    padding-top: 30px;
}
#loginForm legend {
    display: none;
}
#loginForm fieldset {
    width: 330px;
    margin: 10px auto;
    border: 5px solid #efefef;
    padding: 50px;
    border-radius: 15px;
}
#loginForm label.labelStyle {
    width: 60px;
    display: block;
    float: left;
    font-weight: bold;
    letter-spacing: -1px;
}
#userId, #userPWD {
    width: 150px;
    border: 1px solid #999;
    margin-bottom: 5px;
    padding: 2px;
}
#login {
    float: left;
}
#btnLogin {
    display: block;
    background-color: #FF6633;
    border-radius: 5px;
    border-style: none;
    color: #fff;
    width: 80px;
    height: 50px;
    position: relative;

```



```

float:left;
left:10px;
}
#btn1 {
clear:both;
margin-left: 60px;
padding:10px 0px;
}
#btn1 label {
font-size:11px;
vertical-align:top;
}
#btn2 {
margin-left: 60px;
}
#btn2 input {
background-color:#666;
border-style:none;
border-radius:5px;
color:#fff;
padding:5px 10px;
}
</style>
</head>
<body>
<form id="loginForm">
<h1>Member Login</h1>
<fieldset>
<legend>Member Loin</legend>
<div id="login">
<p>
<label for="userId" class="labelStyle">아이디</label>
<input type="text" id="userId" />
</p>
<p>
<label for="userPass" class="labelStyle">비밀번호</label>
<input type="password" id="userPass" />
</p>
</div>
<input type="submit" value="로그인하기" id="btnLogin" />
<p id="btn1">
<input type="checkbox" id="saveId" value="savedIdYes" />
<label for="saveId">아이디저장</label>
<input type="checkbox" id="secure" value="secureYes" />
<label for="secure">보안접속</label>

```

```
</p>
<p id="btn2">
  <input type="button" value="회원가입" id="btnJoin" />
  <input type="button" value="아이디/비밀번호 찾기" id="btnSearch" />
</p>
</fieldset>
</form>
</body>
</html>
```

## 5.2 다단 편집하기

다단 편집기법을 활용해 다음 그림과 같이 실행되는 웹 페이지를 구현해 보자.



### ▶ 예제 5-2) 3단 컬럼으로 구성된 웹 페이지

- CssStudyCh05/css05\_02.html

```
<!DOCTYPE HTML>
<html>
<head>
<title>3단 컬럼으로 구성된 부트스트랩 소개</title>
<meta charset="UTF-8" />
<style type="text/css">
  body {
    font: 12px "맑은 고딕", 돋움, arial;
    margin: 0px;
  }
  ul, li, h1, dl, dt, dd {
    margin: 0px;
```

```

padding: 0px;
}
h1 {
position: fixed;
left: 0px;
top: 0px;
font-size: 2em;
color: #fff;
text-align: center;
padding: 15px;
background-color: #666;
width: 100%;
}
h2 {
font-size: 1.5em;
color: #fff;
text-align: center;
padding: 10px 15px;
background-color: #f39;
width: 880px;
margin: 20px auto;
}
h3 {
font-size: 1.2em;
color: #060;
}
p {
line-height: 150%;
margin-bottom: 30px;
}
#bookImg {
margin: 100px auto 10px;
text-align: center;
}
#bookImg > img {
border-radius: 30px;
}
/* Layout */
#bookPart {
width: 850px;
margin: 10px auto;

/* 각 줄의 정렬을 동일한 너비로 일정하게 양쪽 정렬 */
text-align: justify;

```

```

/* 벤더 프리픽스(Vendor prefix)
* CSS3 표준이 아직 완성되지 않아 정식 권고안이 발표되지 않았을 때
* 웹 브라우저 마다 CSS3 속성을 지원하는 부분이 많이 달랐다.
* 그래서 대부분의 CSS3 속성을 사용하기 위해서는 아래와 같이 속성의
* 이름 앞에 브라우저를 식별할 수 있는 접두어를 붙여 사용했는데
* 이는 웹 브라우저 마다 시험적으로 CSS3를 지원하는 기능을 사용하기
* 위해 필요한 설정이다. 물론 웹 브라우저 마다 CSS3의 정식 권고안을
* 제대로 지원하게 되면 아래와 같은 벤더 프리픽스는 없어 지겠지만
* 아직도 브라우저 마다 다르게 지원하는 CSS3 속성이 있기 때문에 어떤
* 브라우저를 사용하더라도 동일한 결과를 얻기 위해서는 벤더프리픽스를
* 사용하는 것이 최선의 방법이다.
*
* -webkit- : 웹킷(Webkit) 방식의 브라우저에 사용(사파리, 크롬)
* -moz- : 게코(Gecko) 방식의 브라우저에 사용(모질라, 파이어폭스)
* -o- : 오페라 브라우저에 사용
* -ms- : 인터넷 익스플로러에 사용
*
* CSS는 맨 마지막에 정의한 것이 우선 적용되므로 벤더프리픽스로
* 먼저 선언한 후 CSS의 기본 속성을 작성해 두는 것이 바람직한 방법이다.
**/

/* Safari and Chrome */
-webkit-column-count:3;
-webkit-column-rule:1px dashed #666;
-webkit-column-gap:40px;

/* Firefox */
-moz-column-count:3;
-moz-column-rule:1px dashed #666;
-moz-column-gap:40px;

column-count:3;
column-rule:1px dashed #666;
column-gap:40px;
}
</style>
</head>
<body>
<h1>프론트엔드 웹개발을 더 빠르고 쉽게</h1>
<div id="bookImg">
  <a href="http://bootstrap.com/">
    
  </a>
</div>
<h2>부트스트랩의 특징</h2>
<div id="bookPart">

```

<h3>부트스트랩은 반응형 웹을 지원</h3>

<p>부트스트랩은 반응형이며 모바일 우선인 웹 프로젝트 개발을 위한 가장 인기 있는 HTML, CSS, JS 프레임워크입니다.</p>

<h3>모든 사람과 모든 곳을 위한 오픈소스</h3>

<p>부트스트랩은 프론트엔드 웹 개발을 더 빠르고 쉽게 만들어줍니다. 모든 기술 수준의 사람들, 모든 형태의 기기들, 모든 크기의 프로젝트들을 위해 만들어졌습니다. 또한 부트스트랩은 오픈소스입니다. 깃허브에서 호스트되고, 개발되고, 유지보수 됩니다.</p>

<h3>프리프로세서</h3>

<p>부트스트랩은 평범한 CSS 로 제공합니다만, 그것의 소스코드는 2개의 인기 있는 CSS 프리프로세서인 Less 와 Sass 를 사용합니다. 신속하게 프리컴파일된 CSS 로 시작하거나 소스를 빌드하세요.</p>

<h3>하나의 프레임워크, 모든 기기</h3>

<p>부트스트랩은 웹사이트와 어플리케이션을 단일 코드 베이스로 CSS 미디어 쿼리를 이용하여 휴대폰에서부터 태블릿, 데스크탑까지 쉽고 효과적으로 크기를 조절합니다.</p>

<h3>많은 기능들</h3>

<p>부트스트랩과 함께, 보통의 HTML 요소들, 맞춤형 HTML과 CSS 컴포넌트들, 그리고 놀라운 jQuery 플러그인들로 광범위하고 아름다운 문서들을 가질 수 있습니다.</p>

<h3>부트스트랩 전시관</h3>

<p>웹상의 무수히 많은 놀라운 사이트들이 부트스트랩으로 만들어져 있습니다. 들어가는 <a href="http://expo.getbootstrap.com">예제 모음</a>으로 당신의 것을 시작하거나 전시된 사이트를 구경하세요.</p>

<h3>브라우저와 기기 지원</h3>

<p>부트스트랩은 최신 데스크탑과 모바일 브라우저에 최적으로 만들어졌습니다. 이것은 오래된 브라우저는 몇몇 컴포넌트들이 기능적으로는 작동하지만 다르게 보일 수 있음을 의미합니다.</p>

<h3>접근성</h3>

<p>부트스트랩은 보편적인 웹 표준에 따릅니다. 그리고 최소한의 노력으로 AT 를 이용해서 접근성 있는 사이트를 만드는 데 사용할 수 있습니다.</p>

</div>

</body>

</html>

## ▶ 연습문제 5-1 이벤트 게시판 만들기

아래는 그림은 table 태그를 활용해 이벤트 게시판 리스트를 작성하고 실행한 화면이다.

아래 화면에서 보기 버튼과 페이지 번호 버튼은 이미지를 사용하지 않고 css를 활용해 버튼을 만들었고 각 행에 마우스가 커서가 올라가면 그 행의 배경색이 회색 계열로 변경되도록 작성한 것이다.

아래 그림과 같이 실행되는 게시판 화면을 구현하시오.

### 이벤트 게시판

번호	제 목	작성자	작성일	이벤트 보기
1	사이트 리뉴얼 기념 봄 맞이 이벤트	관리자	2022/03/01	보기
2	사이트 첫 방문 환영 감사 이벤트	관리자	2022/02/07	보기
3	회원 서포터즈 가입하고 구매 포인트 받자~	관리자	2022/01/25	보기
4	SNS 신년 나눔 특별 이벤트	관리자	2022/01/15	보기
5	회원 가입하고 첫 주문 선물 팡팡 이벤트	관리자	2022/01/03	보기
7	연인을 위한 신년 사랑 여행 패키지	관리자	2022/01/02	보기
6	크리스마스 사랑 나눔 이벤트	관리자	2021/12/25	보기

1 2 3 4 5 6 7 8 9 10

## ▶ 연습문제 5-2) 회원 가입 폼 만들기

이번에는 HTML5에서 table 태그를 사용해 작성한 연습문제 6-2의 폼 양식을 table 태그를 사용하지 말고 아래 그림과 같이 실행되는 회원 가입 폼을 만들 보자. 참고로 아래의 회원 가입 폼은 공간 분할 태그인 div와 순서 없는 목록인 ul 태그를 사용하여 구성하고 스타일시트에서 화면에 출력될 형식을 지정하는 display 속성에 table, table-row, table-cell 등의 속성을 사용해 순서 없는 목록을 테이블로 출력되도록 구성한 것이다. 아래 그림과 같이 실행되는 회원 가입 폼을 구현하시오.

회원정보입력

회원정보는 개인정보취급방침에 따라 안전하게 보호되며 회원님의 명백한 동의 없이 공개 또는 제3자에게 제공되지 않습니다. [개인정보취급방침](#)

표시는 필수입력사항

회원 기본정보

이름	<input type="text"/>
아이디	<input type="text"/> 중복확인
비밀번호	<input type="password"/>
비밀번호 확인	<input type="password"/>
이메일	<input type="text"/> @ <input type="text"/> -선택해 주세요- ▼

회원 신상정보

주소	<input type="text"/> - <input type="text"/> 우편번호 찾기 <input type="text"/>
휴대폰	010 ▼ <input type="text"/> <input type="text"/>
직업	직업을 선택해 주세요 ▼
취미	<input type="checkbox"/> 영화감상 <input type="checkbox"/> 독서 <input type="checkbox"/> 골프 <input type="checkbox"/> 낚시 <input type="checkbox"/> 여행 <input type="checkbox"/> 기타
가입경로	<input type="radio"/> TV <input type="radio"/> 신문/잡지 <input type="radio"/> 인터넷 <input type="radio"/> 지인소개 <input type="radio"/> 기타

확인

취소