

ch02.CSS

디지털마케팅 프로그래밍

by Wooyoung Choi, KMU Digital Marketing MBA, 2025

b,i vs strong,em

- b,i: 스타일의 용도(CSS font-style로 지정하는 것이 나음)
- strong,em: 의미 강조(웹접근성, SEO)

iframe

- 문서안에 프레임을 만들고 다른 컨텐츠나 외부 문서를 표현할 때 사용
- title 속성으로 iframe 컨텐츠의 내용을 알려줘야함(웹접근성)

```
<p>  
<iframe src="https://www.google.com/" id="google-home" name="google-home" frameborder="1"  
width="320" height="240" scrolling="auto" title="구글 웹사이트">
```

프레임이 지원되지 않는 환경입니다.

```
<a href="https://www.google.com/">https://www.google.com/</a> 페이지를 방문하여 주세요.</iframe>  
</p>
```

CSS

CSS(Cascading Style Sheets)

- HTML 또는 XML로 작성된 문서의 표현을 기술하기 위한 스타일시트언어
- W3C의 규격화에 따라 모든 웹 브라우저에서 표준화한 오픈 웹의 주요언어
- 구조와 표현을 분리해 마크업 코드가 복잡해지는 문제를 해결 가능
- CSS와 HTML의 분리로 HTML이 간결해지고 더욱 구조화 됨(SEO에 유리)

CSS Selector

- CSS 속성값을 적용하기 위해 어떤 HTML 요소를 선택해야 하는지 브라우저에 알려주는 요소 및 기타용어의 패턴
- 선택자의 대상: 선택자에 의해 선택되는 요소

```
selector { property: value; }
```

- selector: 선택자
- {}: 선언부(Declaration block)
- property: 속성
- value: 속성값
- ;: delimiter(구분자)

CSS 적용하는 법

1. External Stylesheet

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

2. Internal Stylesheet

```
<style></style>
```

3. In-line Stylesheet

```
<span style="">
```

id, class

- id: 문서 전체에서 특정한 요소에 유일(unique)한 이름을 부여하는 식별자
 - 목적: CSS, JavaScript와 연결할 때 요소를 식별하기 위함
 - 문서 전체에서 유일
- class: CSS와 JavaScript가 특정한 요소를 선택하여 접근하기 위한 속성
 - 목적: CSS, JavaScript에 미리 정의된 일을 중복 적용, 수행하게 하기 위함
 - 중복 사용 가능

```
<div id="main-wrapper"></div>
```

```
<p class="heading-text main-title "></p>
```

Selector

- 전체선택자(Universal Selector)
 - *: 모든 요소를 선택
- 요소선택자(Type Selector)
 - C: "C" 요소를 선택(HTML 요소 선택)
- 클래스선택자(Class Selector)
 - C.main-title: class가 main-title인 C요소를 선택
- 아이디선택자(ID Selector)
 - C#wrapper: id가 wrapper인 C태그 선택
- 속성선택자(Attribute Selector)
 - C[attr]: attr 속성을 가진 C요소 선택
 - C[attr="val"]: attr 속성의 값이 val인 C요소 선택
 - C[attr~="val"]: attr 속성의 여러 값 중 val을 가지는 C요소 선택

Selector

- 가상클래스선택자(Pseudo-classes Selector)
 - C:hover: 마우스 포인터가 올라간 C 요소를 선택
- 가상요소선택자(Pseudo-element Selector)
 - C::after: C콘텐츠 뒤로 생성된 가상요소를 선택

Selector

- 선택자 결합(Selector Combinators)
 - 하위선택자(Descendant Combinator)
 - #wrapper div: 선행선택자의 하위요소 중 후행선택자에 해당하는 요소 선택
 - 자식선택자(Child Combinator)
 - #wrapper div: 후행선택자의 부모요소가 선행선택자를 가지는 요소만 선택
- 형제선택자(Sibling Combinator)
 - h1 + p: 선행 선택자 뒤의 후행 선택자를 선택
 - .subtitle ~ p: 선행선택자의 뒤에 인접하여 등장하는 모든 후행 선택자 선택
- 선택자리스트(Selector List)
 - span, div: 선택자를 나열

Font, Text style

font-family

- 폰트 지정 속성
- - generic-family: serif, sans-serif, cursive, fantasy, monospace

```
h1, h2, h3, h4, h5, h6 {  
  font-family: sans-serif;  
}
```

web font

- <https://fonts.google.com/>
- <https://hangeul.naver.com/font/nanum>
- 웹 폰트 적용시 리소스 크기(사이트 퍼포먼스), 저작권 확인 후 사용

```
<link href="https://hangeul.pstatic.net/hangeul_static/css/nanum-square.css" rel="stylesheet">
```

```
h1, h2, h3, h4, h5, h6 {  
    font-family: 'NanumSquare', sans-serif;  
}
```

font-weight

- 폰트 굵기 설정
- 100~900(100 단위) 혹은 thin, normal, bold 등으로 표현

```
p {font-weight: 400;}
```

```
h1 {font-weight: bold;}
```

font-size

- 폰트 크기 설정
- 상대값 혹은 절대값으로 지정

구분	xx-small	x-small	small	medium	large	x-large	xx-large
비율	3/5	3/4	8/9	1	6/5	3/2	2/1
heading	h6		h5	h4	h3	h2	h1

```
body {font-size: 18px;}
```

```
div {font-size: 2em;}
```

```
div p {font-size: small;}
```

line-break

- 줄바꿈 규칙 지정
- strict, auto(default), loose

```
div {line-break: loose;}
```

word-break

- 줄바꿈시 단어 규칙 지정
- normal(default), break-all, keep-all

```
div p {word-break: break-all;}
```

text-align

- 단락 내 텍스트의 가로방향 정렬 방법 지정
- left, right, center, justify, [string] - CSS2.1
- start, end, match-parent, start end

```
div p {text-align: center;}
```

letter-spacing

- 글자간격

```
div p {letter-spacing: -0.2em;}
```

word-spacing

- 단어간격

제목텍스트: 글자간격은 줄이고 단어간격은 늘여야 가독성이 높아짐

```
div p {letter-spacing: 0.2em;}
```

line-height

- 줄간격

본문텍스트: 단어간격은 늘이고 문장간격은 늘여야 가독성이 높아짐

```
div p {line-height:1.8em;}
```

CSS Box

CSS Box

- CSS는 요소 표시를 위해 사각형의 박스를 생성함
- 요소 박스 크기는 안쪽여백(padding), 테두리(border), 바깥여백(margin)으로 결정
- block: 부모요소 너비 전체, 줄바꿈 있음
- inline: 컨텐츠쪽으로 너비 결정, 자동 줄바꿈
- inline-block: width 속성으로 요소 너비 직접 지정

```
div.hidden {display: none;}
```

```
span {display: block;}
```

```
div {display: inline;}
```

```
div {display: inline-block;}
```

padding

- 내부여백 설정(content와 border 사이)
- 값은 1~4개까지 지정(top 부터 시계방향, 지정안된 값은 반대편 값 상속)

border

- 테두리 지정 속성
- 너비(border-width), 스타일(border-style), 색깔(border-color) 설정 가능

```
div {border: 1px solid #00ffff;}
```

margin

- 외부여백 설정(border와 다른 컨텐츠 사이)
- 값은 1~4개까지 지정(top 부터 시계방향, 지정안된 값은 반대편 값 상속)

padding, margin

```
div {padding: 10px;}
```

```
div {padding: 10px 20px;}
```

```
div {padding: 10px 20px 30px;}
```

```
div {padding: 10px 20px 30px 40px;}
```

```
div {margin: 10px;}
```

```
div {margin: 10px 20px;}
```

```
div {margin: 10px 20px 30px;}
```

```
div {margin: 10px 20px 30px 40px;}
```

width, height

- 요소의 너비, 높이 지정

```
div {width: 50%;}
```

```
div {height: 300px;}
```

CSS flex

- 유연한 요소 배치를 위한 flexible box layout
- display: flex : block 수준의 유연한 박스
- display: inline-flex : inline 수준의 유연한 박스

flex-direction

- 요소박스의 배치 방향 지정
- row, column, row-reverse, column-reverse

```
div {  
    display: flex;  
  
    flex-direction: row; /* flex-flow: row wrap; */  
  
    flex-wrap: wrap; /* default: nowrap */  
}
```

flex-direction

flex-direction: [row, row-reverse, column, column-reverse]

flex container에서 아이템의 나열 방향 결정

flex-wrap

flex-wrap: [nowrap, wrap, wrap-reverse]

flex container에서 아이템을 일렬로 배치할지, 줄바꿈하여 표현할지 결정

flex-flow

flex-flow: [direction] [wrapping behavior]

shorthand-property, flex-direction+flex-wrap

flex practice

```
<style>
  .flex-container {
    display: flex;
    width: 295px;
    flex-wrap: wrap-reverse;
  }
  .flex-box {width: 100px; height: 100px; text-align: center;}
  .bg-red {background-color: red;}
  .bg-blue {background-color: blue;}
  .bg-gray {background-color: gray;}
</style>
<section class="flex-container">
  <article class="flex-box bg-red">1</article>
  <article class="flex-box bg-blue">2</article>
  <article class="flex-box bg-gray">3</article>
</section>
```