

# 01. HTML 기본기 다지기

01-1

HTML과의 첫 만남

01-2

웹 브라우저와 웹 편집기

01-3

HTML 문서 기본 구조

01-4

웹 문서 만들고 업로드하기



```
index.html
1<!DOCTYPE html>
2<html lang="ko-KR">
3  <head>
4    <meta charset="UTF-8">
5    <title>My Times with Coffee</title>
6  </head>
7  <body>
8    <div id="header">
9      <div>My Times with Coffee</div>
10     <div>로그인</div>
11     <div>회원가입</div>
12     <div>회원정보수정</div>
13   </div>
14   <div id="nav">
15     <a href="#">Home
16     <a href="#">About
17     <a href="#">Contact
18   </div>
19   <div id="content">
20     <h1>My Times with Coffee</h1>
21     <p>My Times with Coffee</p>
22     <img alt="A cup of coffee with steam rising from it." data-bbox="350px auto 450px 250px"/>
23   </div>
24 </body>
25 </html>
```

```
index.css
1.choice {
2  float: right;
3  margin-right: 5px;
4}
5.choice .text {
6  float: left;
7}
8.footer {
9  clear: both;
10}
```

## HTML이란 무엇일까?

- 컴퓨터에서 사용하는 모든 파일에는 각각 고유의 형식이 있다.
- 웹에서는 웹에 맞는 형식인 \*.html(또는 \*.htm)로 문서를 저장해야 한다.
- 텍스트뿐만 아니라 이미지, 링크 등 여러 요소들을 다루고 표시할 수 있어야 한다.
- 웹에서 자유롭게 오갈 수 있는 웹 문서를 만드는 언어가 **HTML**

웹 문서를 보는 프로그램



웹 편집기

```

<!doctype html>
<html lang="ko">
  <head>
    <meta charset="utf-8">
    <title>기본 HTML 문서</title>
  </head>
  <body>
    <h1>시간이란..</h1>
    
    <p>내일 죽을 것처럼 오늘을 살고 영원히 살 것처럼 내일을 꿈꾸어라.</p>
  </body>
</html>

```

언어 팔초 부분이 HTML 태그(tag)

웹 문서를 작성하는 프로그램

## 웹 표준이란 무엇일까?

- 웹 사이트를 만들 때 지켜야 하는 약속들을 정리한 것
- 웹 표준을 지켜 사이트를 제작하면 장소나 브라우저와 상관없이 쉽게 웹 사이트를 볼 수 있다.
- 웹 표준으로 문서 하나를 만들면 어떤 기기에서나 볼 수 있기 때문에 웹 개발자와 디자이너의 시간 절약
- HTML5로 문서를 만드는 것 = 웹 표준을 지킨 문서를 만드는 것

## 왜 HTML5와 CSS3를 공부해야 할까?

### 최신 웹 표준에 맞는 웹 사이트 제작

HTML5과 CSS3를 사용하면 사용자가 접속한 기기에 따라 사이트 레이아웃을 다양하게 바꿀 수 있다

### 앱 화면을 디자인하기 위한 기초

HTML5의 API를 사용해 '웹앱'을 만들 수도 있고, 앱 화면은 HTML5과 CSS3를 사용해 디자인.

### 인터랙티브한 사이트 제작

플러그인을 제작할 줄 몰라도 HTML5과 함께 CSS3를 이용하면 인터랙티브한 사이트 제작 가능

### 사이트와 블로그 수정이 쉽다

HTML5를 공부하면 사이트나 블로그 소스를 이해할 수 있고 사이트를 원하는 모습으로 바꿀 수 있다.



## HTML5 ? HTML ?

대부분의 웹 브라우저에서 HTML5를 지원하게 되면서 현재 HTML5의 공식 명칭은 'HTML'이다.

## HTML5.1 vs HTML5.2

HTML5 표준안부터는 웹 브라우저 업체들이 함께 참여하고 있기 때문에 표준안이 업그레이드될 때마다 웹 브라우저 업체에서 빨리 수용하고 지원하며 살아있는 표준안이 되고 있다.

2016년 11월 현재,

- HTML 5.1이 최신 표준안(<http://www.w3c.org/TR/html>)
- HTML 코드의 표준화는 계속 진행되어 HTML 5.2 개발 진행 중.(<https://w3c.github.io/html/>)



# 웹 브라우저와 웹 편집기

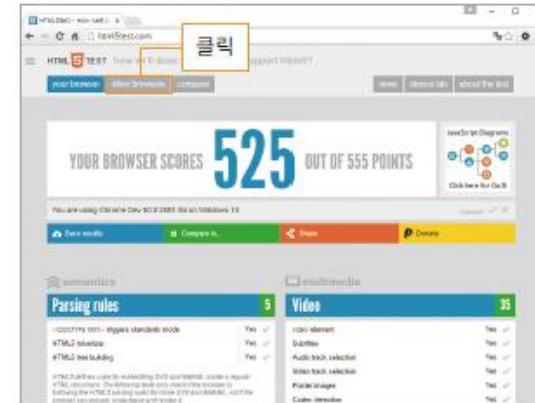
브라우저 이름	제조업체	특징
크롬	구글	빠른 업데이트를 통해 HTML5 표준에 가장 빨리 대응하고 있어 최근 가장 많이 사용되는 브라우저입니다. 
파이어폭스	모질라	일반 사용자들보다 개발자들이 자주 사용하는 웹 브라우저로 개발 도구와 부가 기능들이 뛰어납니다. 
인터넷 익스플로러	마이크로소프트	HTML5에 대한 지원은 취약하지만 아직까지 인터넷 익스플로러(IE)를 기준으로 하는 사이트가 많기 때문에 국내에서 많이 사용되고 있습니다. 윈도우 7 서비스팩1부터 지원하기 시작한 IE11을 마지막으로 더 이상 개발되지 않고 있습니다. ▶ 원도우 10에서는 '보조 프로그램' 안에 'IE11'가 포함되어 있습니다. 
엣지	마이크로소프트	마이크로소프트에서 인터넷 익스플로러 대신 HTML5를 지원하기 위해 새로 개발해 윈도우 10에서 기본으로 사용하는 웹 브라우저입니다. 

▶ 이외에도 애플의 '사파리'나 오페라 재단의 '오페라' 브라우저도 있습니다.

## [실습] 크롬 설치하고 기본 웹 브라우저로 정하기

사용 중인 웹 브라우저가 HTML5를 얼마나 지원하는지 확인하려면

<http://html5test.com>



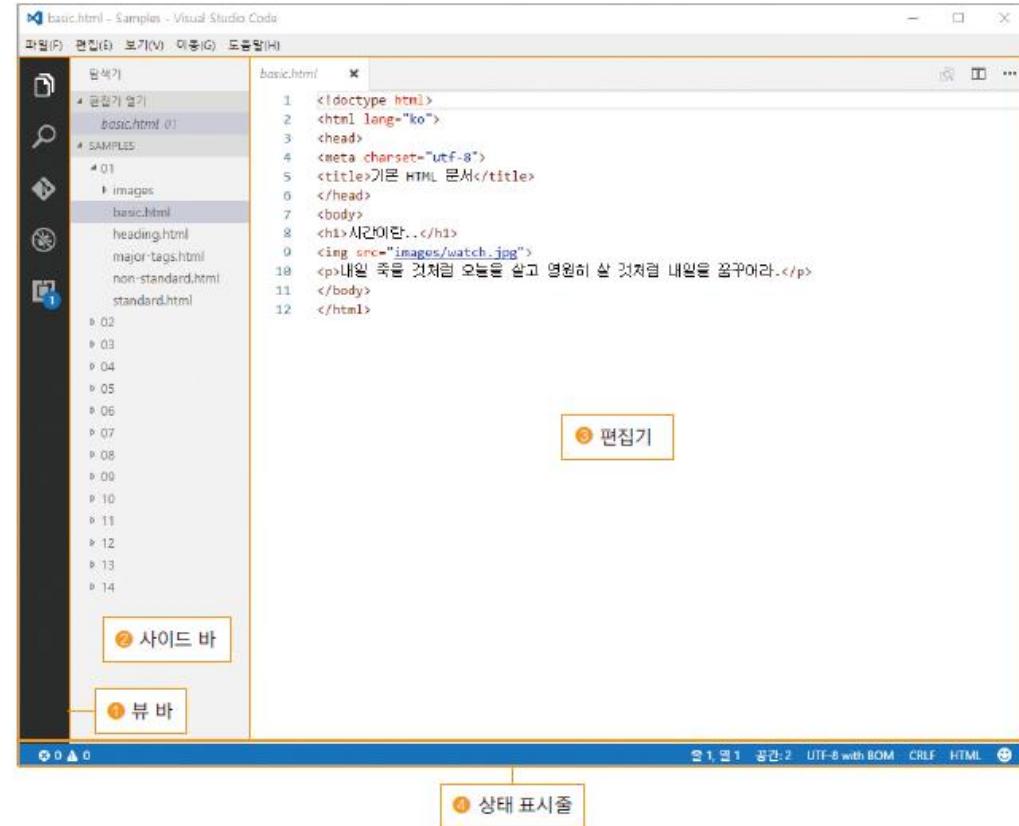
크롬 최신 버전의 점수가 가장 높습니다.

# 웹 브라우저와 웹 편집기

편집기 종류	특징	편집기 종류	특징
텍스트 편집기	<p>적절한 프로그램은 아니지만 HTML 태그에 익숙하다면 윈도우의 메모장이나 매킨토시의 텍스트 편집기를 이용해서도 웹 문서를 작성할 수 있습니다. (물론 한글이나 워드에서도 웹 문서를 작성할 수 있죠) 사용자가 태그를 일일이 직접 입력해야 하기 때문에 HTML 태그와 속성을 알고 있어야 하고 입력할 때 오류가 발생하기 쉽습니다.</p>		<p>웹을 개발할 때 대부분 단순히 HTML만 사용하는 것이 아니라 자바스크립트나 ASP.NET, PHP, node.js 등 여러 언어들을 함께 사용합니다. 웹 IDE는 웹 편집기의 편리함과 함께 사용자에게 필요한 개발 환경을 한꺼번에 구축하고 사용할 수 있게 해줍니다. 웹 IDE 프로그램에는 비주얼 스튜디오 2015(Visual Studio 2015)(무료, 유료), 웹스톰(Webstorm)(무료), 서브라임 텍스트(Sublime Text)(무료, 유료) 등이 있습니다.</p>
웹 편집기	<p>사용하기 권하는 편집기로 웹을 위한 전용 편집기입니다. 태그와 속성, 일반 텍스트 등을 서로 다른 색으로 표시해주므로 소스를 읽기 쉽고 편집기에 따라 태그나 속성의 일부만 입력하더라도 전체를 완성해주기 때문에 오타로 인한 문제를 줄일 수 있습니다. 윈도우용 편집기로는 노트패드++(무료)나 에디트 플러스(유료) 등이 있고 맥용 편집기로는 텍스트메이트(Textmate)(무료)나 코다(Coda)(유료) 등이 있으며 드림위버(Dreamweaver)(유료)나 비주얼 스튜디오 코드(Visual Studio Code)(무료) 같은 편집기는 윈도우와 맥에서 모두 사용 가능합니다.</p> <p>▶ 이 책에서는 비주얼 스튜디오 코드를 이용해 소스 코드를 작성할 것입니다.</p>		 <p>비주얼 스튜디오 2015      웹스톰      서브라임 텍스트</p>

## 비주얼 스튜디오 코드

- 대부분의 주요 플랫폼에서 모두 사용할 수 있습니다.
- 태그와 CSS 속성을 친절히 안내합니다
- 태그와 CSS 속성을 간편하게 입력할 수 있습니다
- 항목을 한꺼번에 변경할 수 있습니다
- 확장이 쉽습니다



[실습] 비주얼 스튜디오 코드 설치하고 살펴보기

## 태그, 이건 꼭 알아두세요!

- 태그는 소문자로 쓴다  
예) ``
- 여는 태그와 닫는 태그를 정확히 입력한다 (닫는 태그가 없는 태그도 있다)
- 적당하게 들여쓴다
- 태그는 속성과 함께 사용된다  
예) ``
- 인코딩 방식은 utf-8로 한다  
예) `<meta charset="utf-8">`

[실습] 비주얼 스튜디오 코드에서 간단한 HTML 문서 만들기

## HTML 문서 기본 구조 살펴보기

```
<!doctype html> -①
<html>
  <head>
    <meta charset="utf-8">
    <title>내가 처음 만드는 HTML 문서</title>
  </head>
  <body>
    <h1>시간이란..</h1>
    <p>내일 죽을 것처럼 오늘을 살고
      영원히 살 것처럼 내일을 꿈꾸어라.</p>
    
  </body>
</html>
```

❶ <!doctype html> - 현재 문서가 HTML5 언어로 작성된 웹 문서라는 뜻입니다.

❷ <html> ~ </html> - 웹 문서의 시작과 끝을 나타내는 태그입니다. 웹 브라우저가 <html> 태그를 만나면 </html>까지의 소스를 읽어 HTML 문법에 맞추어 브라우저에 표시합니다.

❸ <head> ~ </head> - 웹 브라우저가 웹 문서를 해석하기 위해 필요한 정보들을 입력하는 부분입니다. 여기

에 있는 정보는 실제 문서 내용이 아니기 때문에 문서 제목만 브라우저 창에 표시되고 나머지는 웹 브라우저 화면에 표시되지 않습니다. 스타일 및 스크립트 등이 포함되기도 합니다.

❹ <body> ~ </body> - 실제로 웹 브라우저 화면에 나타날 내용입니다. 앞으로 우리가 공부하게 될 HTML 태그들은 대부분 <body> 태그 안에 들어갑니다.

## HTML 문서와 DOCTYPE

<!doctype> - 웹 브라우저에게 '이제부터 처리할 문서는 HTML 문서이고 어떤 유형을 사용 했으니 그 버전에 맞는 방법으로 해석하라.' 라고 알려주는 것

<!DOCTYPE html> 또는 <!doctype html>

## <html> 태그 – 웹 문서 시작을 알리는 태그

- 실제 문서 정보와 내용이 시작되고 끝나는 것을 표시하는 태그
- lang이라는 속성을 사용해 문서에서 사용할 언어 지정 가능
- 문서 정보를 지정하는 <head> 부분과  
실제 화면에 보이는 문서 내용을 입력하는 <body> 부분

```
<!doctype html>
<html lang="ko">
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

## 브라우저에게 정보를 주는 `<head>` 태그

- 웹 브라우저 화면에는 보이지 않지만, 웹 브라우저가 알아두어야 할 정보들 입력
- 문서에서 사용할 외부 파일들 링크

`<title>` 태그 : 브라우저의 제목 표시줄에 표시되는 내용.      `<title> 문서 제목 </title>`

`<meta>` 태그 : 문자 인코딩 방법 및 문서의 키워드와 요약 정보를 지정      `<meta charset="utf-8">`

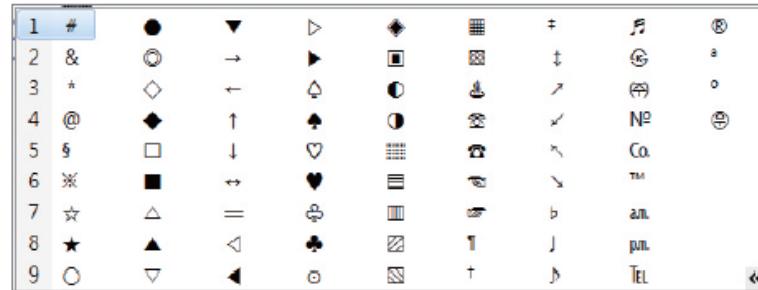
## 문서의 몸통 `<body>` 태그

- 실제 브라우저에 표시될 내용 입력
- 이 책에서 설명하는 대부분의 태그가 `<body>` 태그와 `</body>` 태그 사이에서 사용하는 태그들

## 특수 문자 입력하기

키보드에서 한글 자음을  
누른 후 <한자>키를 누르면 특수문자가 표시된다.

한글 각 자음마다 서로 다른 특수 문자가 표시된다



'ㅁ'과 '한자'를 눌렀을 때 사용할 수 있는 특수 문자

## 특수 기호 입력하기

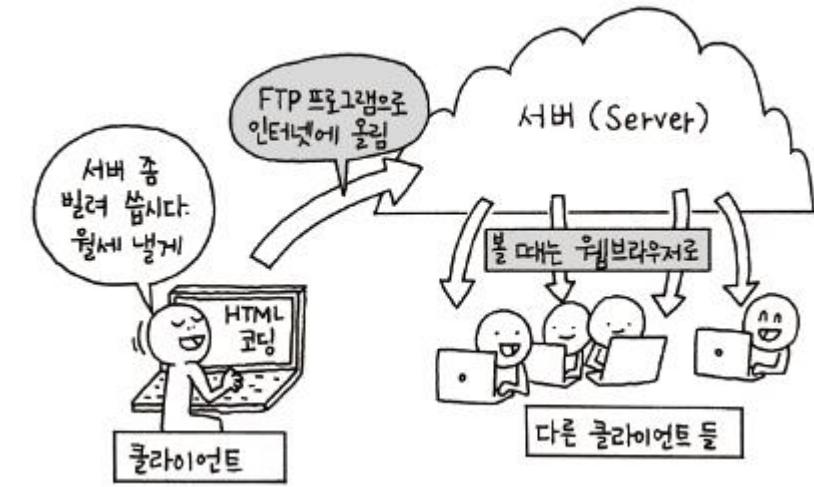
HTML 문서의 특성상 여러 개의 공백을 나타내거나  
따옴표, '<' 같은 꺾쇠 괄호를 화면에 표시할 때에도  
특수 기호로 입력한다.

화면에 표시되는 모습	특수 기호
&	&amp;
(공백 한 칸)	&ampnbsp
<	&lt;
>	&gt;
"	&quot;
	&#124;
(	&#40;
)	&#41;
,	&#44;
-	&#45;
'	&acute;

# 웹 문서 만들고 업로드하기

## 호스팅 서버 준비하기

- HTML로 만든 웹사이트를 다른 사람들이 볼 수 있도록 하려면  
→ 웹 문서와 사용한 파일들을 서버 컴퓨터로 업로드해야 한다.
- 웹 서버를 직접 구입하고 관리하기 어렵기 때문에  
매달 혹은 몇 년마다 일정 금액을 내고 사용하는  
호스팅 서비스를 많이 이용한다.



## FTP 프로그램 설치하기

FTP 프로그램 : 사용자 컴퓨터에서 작성한 웹 문서와 각종 파일을 서버로 업로드(upload)하거나  
서버에서 다운로드(download)할 수 있도록 해주는 프로그램

[실습] 닷홈에 무료 호스팅 서비스 신청하기

[실습] FileZilla 설치하고 서버에 접속하기

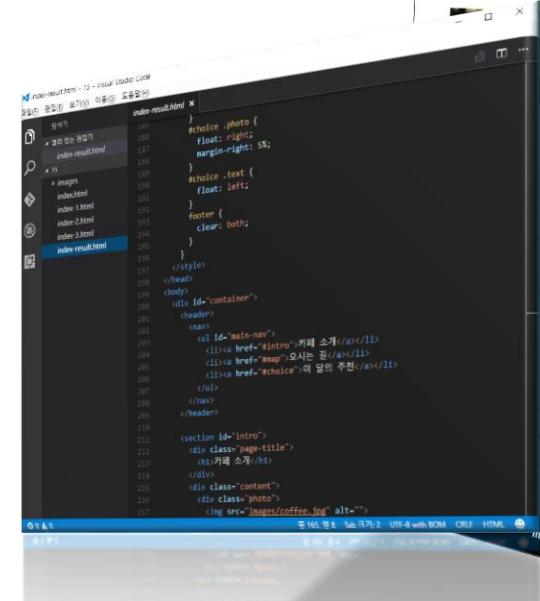
# 02. 텍스트 관련 태그들

02-1 텍스트를 묶어주는 태그

02-2 텍스트를 한 줄로 표시하는 태그

02-3 목록을 만드는 태그

02-4 표를 만드는 태그



The screenshot shows a browser displaying a coffee shop's website. The page features a large banner with a photo of a coffee shop interior and Korean text: "매점 소개", "경영 시간: 오전 9시 ~ 밤 10시", "호주: 매주 수요일 (수요일의 경우 점장 경우 수요일 영업 나갈 날 예상)", and "서귀포시 연락연 사계리 0000-0000". Below the banner, there's a header with a navigation menu and a section titled "인테리어" containing a heading "카페 소개" and three links: "오시는 길", "이 달의 주제", and "チョイス". The code editor in the foreground shows the corresponding HTML and CSS code for this section.

```
<div id="index-intro">
    <div class="page-title">
        <h1>카페 소개</h1>
    </div>
    <div class="content">
        <div class="photo">
            
        </div>
        <div>
            <h2>매점 소개</h2>
            <p>경영 시간: 오전 9시 ~ 밤 10시</p>
            <p>호주: 매주 수요일 (수요일의 경우 점장 경우 수요일 영업 나갈 날 예상)</p>
            <p>서귀포시 연락연 사계리 0000-0000</p>
        </div>
    </div>
</div>
```

# 텍스트를 묶어주는 태그

## <hn> 태그 – 제목 표시

- 기본형 : <hn> 제목 </hn>
- 각 웹 콘텐츠 영역에서 제목을 표시할 때 사용하는 태그
- h1 > h2 > h3 > h4 > h5 > h6

## <p> 태그 – 텍스트 단락

- 기본형 : <p> 텍스트 </p>
- 입력한 내용 앞뒤로 빈 줄이 생기면서 텍스트 단락이 만들어짐

## <br> 태그 – 줄 바꾸기

- 기본형 : <br>
- 줄을 바꿀 위치에 <br> 태그를 사용. 닫는 태그가 없음

## <blockquote> 태그 – 인용문 넣기

- 기본형 : <blockquote> 인용 내용 </blockquote>
- 다른 텍스트보다 안으로 들어 써짐.

## <hr> 태그 – 분위기 전환

- 기본형 : <hr>
- 주제가 바뀔 때 분위기 전환. 수평 줄 생김

## <pre> 태그 – 입력한 그대로 표시

- 기본형 : <pre> 텍스트 </pre>
- 소스에 표시한 공백이 그대로 표시됨.
- 프로그램 소스를 표시할 때 유용함.

# 텍스트를 묶어주는 태그

```

<body>
  <h1>제주 이색 여행지</h1>
  <h2>야외 텐트를 닮은 건축물 "테쉬폰"</h2>
  <p>아일랜드 출신 임피제 신부가 1954년 제주에 오면서 목장 숙소
로 짓기 시작한 후 사료공장, 성당으로 활용됐습니다.</p>
  <p>제주에서 점차 다른 지방으로 보급됐지만 현재 제주에만 건축물
이 남아있는데, <br> 국내 근현대 건축사의 한 페이지를 보여주는 가치
를 지닌다고 전문가들은 평가합니다.</p>
  <hr>
  <blockquote>
    성이시돌목장은 제주특별자치도 제주시 한림읍 금악리에 있는 목
장이다. 특히 이시돌목장은 제주 지역 최초의 전기업목장(全企業牧場)으
로 1961년 11월 말 제주시 한림읍 금악리에 세워 양돈 사업을 실시하
였으며 면양을 사육하였던 것으로 알려져 있다. 이시돌목장의 특색있는
건축양식으로 테쉬폰도 유명하다. (출처:향토문화전자대전)
  </blockquote>
</body>

```

예제 02/text-1.html

## 제주 이색 여행지

### 야외 텐트를 닮은 건축물 "테쉬폰"

아일랜드 출신 임피제 신부가 1954년 제주에 오면서 목장 숙소로 짓기 시작한 후 사료공장, 성당으로 활용됐습니다.

제주에서 점차 다른 지방으로 보급됐지만 현재 제주에만 건축물이 남아있는데, 국내 근현대 건축사의 한 페이지를 보여주는 가치를 지닌다고 전문가들은 평가합니다.

성이시돌목장은 제주특별자치도 제주시 한림읍 금악리에 있는 목장이다. 특히 이시돌목장은 제주 지역 최초의 전기업목장(全企業牧場)으로 1961년 11월 말 제주시 한림읍 금악리에 세워 양돈 사업을 실시하였으며 면양을 사육하였던 것으로 알려져 있다. 이시돌목장의 특색있는 건축양식으로 테쉬폰도 유명하다. (출처:향토문화전자대전)

# 텍스트를 한 줄로 표시하는 태그

## <strong>, <b> – 굵게 표시

- <strong> - 중요한 내용이라서 강조해야 할 때
- <b> - 단순히 굵게 표시할 때

<h2>제주 이색 여행지 - 이중섭 거리</h2>  
 <p><strong>주말</strong>마다 <b>'서귀포문화예술디자인시장'</b>이 열립니다.</p>  
 <p><em>'아트마켓'</em>이라고도 부르는데, <i>문화예술체험</i>이나 <i>공연관람</i>을 할 수도 있고 <br> 작가들이 직접 만든 창작예술품 등을 판매하기도 합니다.</p>

## <em>, <i> – 이탤릭체로 표시하기

- <em> - 흐름상 특정 부분을 강조하고 싶을 때
- <i> - 단순히 이탤릭체로 표시할 때

### 제주 이색 여행지 - 이중섭 거리

주말마다 '서귀포문화예술디자인시장'이 열립니다.

'아트마켓'이라고도 부르는데, 문화예술체험이나 공연관람을 할 수도 있고 작가들이 직접 만든 창작예술품 등을 판매하기도 합니다.

예제 02/text-3.html

## <q> 태그 – 인용 내용 표시

- 줄바꿈 없이 다른 내용과 한 줄에 인용 내용 표시
- 인용 내용 앞뒤에 따옴표(" ") 추가됨

### <h1>웹 접근성</h1>

<p>웹의 창시자인 팀 버너스 리 (Tim Berners-Lee)의 <a href="http://www.w3.org/standards/webdesign/accessibility">웹의 힘은 보편성에 있다. 장애에 구애없이 모든 사람이 접근할 수 있는 것이 필수적인 요소이다.</a>라는 말로 웹 접근성을 설명한다. </p>

### 웹 접근성

웹의 창시자인 팀 버너스 리 (Tim Berners-Lee)의 “웹의 힘은 보편성에 있다. 장애에 구애없이 모든 사람이 접근할 수 있는 것이 필수적인 요소이다.”라는 말로 웹 접근성을 설명한다.

# 텍스트를 한 줄로 표시하는 태그

## <mark> - 형광펜 효과

<mark> 태그로 묶은 부분의 배경색이 노랑으로 표시됨.

## <span>, <div> - 영역 묶기

- <span> - 줄 안에서 (인라인) 묶기
- <div> - 줄 바꿔 (블록) 단락으로 묶기

## <ruby> - 동아시아 글자에 주석 표시

<ruby> 태그 안에 <rt> 태그를 사용해 주석 표시

**<h2>**야외 텐트를 닮은 건축물 **<mark>"테쉬폰"</mark></h2>**  
**<p>**아일랜드 출신 임피제 신부가 1954년 제주에 오면서 목장 숙소로 짓기 시작한 후 사료공장, 성당으로 활용됐습니다. 제주에서 점차 다른 지방으로 보급됐지만 현재 제주에만 건축물이 남아있으며, **<span style="color:blue;">**국내 근현대 건축사의 한 페이지를 보여주는 가치를 지닌다</span>고 전문가들은 평가합니다.</p>

### 야외 텐트를 닮은 건축물 "테쉬폰"

아일랜드 출신 임피제 신부가 1954년 제주에 오면서 목장 숙소로 짓기 시작한 후 사료공장, 성당으로 활용됐습니다. 제주에서 점차 다른 지방으로 보급됐지만 현재 제주에만 건축물이 남아있으며, 국내 근현대 건축사의 한 페이지를 보여주는 가치를 지닌다고 전문가들은 평가합니다.

예제 02/text-5.html

**<p>**루비(Ruby)는 1995년, 일본의 프로그래머인 마츠모토 유키히로(<ruby>松本行弘<rt>まつもとゆきひろ</rt></ruby>)가 만든 프로그래밍 언어입니다.</p>

루비(Ruby)는 1995년, 일본의 프로그래머인 마츠모토 유키히로(<ruby>まつもとゆきひろ</rt></ruby>)가 만든 프로그래밍 언어입니다.

예제 02/text-6.html

# 목록을 만드는 태그

## <ul>, <li> - 순서 없는 목록

- 각 항목 앞에 불릿이 붙여짐
- CSS의 list-style-type 속성으로 불릿 수정

## <ol>, <li> - 순서 목록

- 각 항목 앞에 숫자가 붙여짐
- <ol> 태그의 속성
  - type 속성 : 불릿 앞의 숫자 조정
  - start 속성 : 중간 번호부터 수정
  - reserved 속성 : 역순으로 표시

```

<ul>
  <li>1일차
    <ol type="a">
      <li>해녀박물관</li>
      <li>낚시체험</li>
    </ol>
  </li>
  <li>2일차
    <ol type="a" start ="3">
      <li>용눈이오름</li>
      <li>만장굴</li>
      <li>카약체험</li>
    </ol> start>
  </li>
</ul>

```

예제 02/ol.html

### 1박 2일 가족 여행 코스



★ 여러 항목이 나열될 때 </li> 태그를 생략해도 다음에 오는 <li> 태그를 만나면 자동으로 그 전에 </li> 태그가 있는 것처럼 인식함

# 목록을 만드는 태그

## <dl>, <dt>, <dd> - 설명 목록

- '제목'과 그에 대한 '설명'으로 이루어진 목록
- <dl>과 <dt>, <dd> 태그 사용
- 하나의 <dt>에 여러 개의 <dd> 값을 가질 수 있다

예제:02/dl.html

&lt;dl&gt;

```

<dt>올레 1코스</dt>
<dd>코스 : 시흥 초등학교 옆 - 광치기 해변</dd>
<dd>거리 : 14.6km(4~5시간)</dd>
<dd>난이도 : 중</dd>
<dt>올레 2코스</dt>
<dd>코스 : 광치기 해변 - 온평 포구</dd>
<dd>거리 : 14.5km(4~5시간)</dd>
<dd>난이도 : 중</dd>
</dl>

```

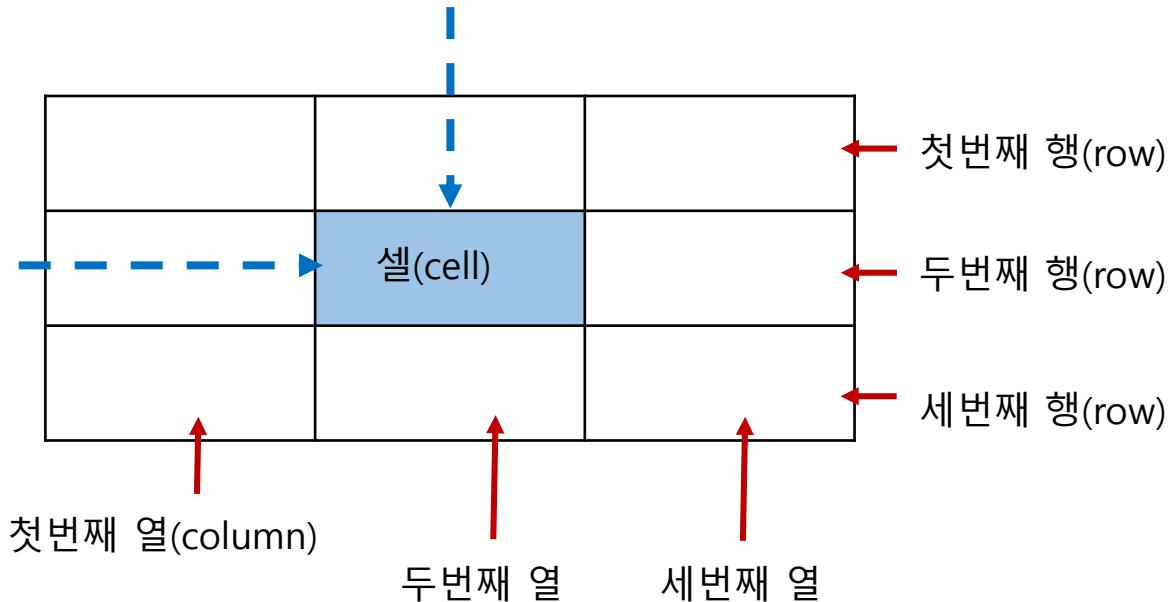
### 제주 올레

올레 1코스
코스 : 시흥 초등학교 옆 - 광치기 해변
거리 : 14.6km(4~5시간)
난이도 : 중
올레 2코스
코스 : 광치기 해변 - 온평 포구
거리 : 14.5km(4~5시간)
난이도 : 중

# 표를 만드는 태그

## 표(table)

자료를 보기 좋게 정리한 것



## 표를 만드는 태그

<table> ~ </table> : 표 전체

<tr> ~ </tr> : 행

<td> ~ </td> : 셀, <th> ~ </th> 제목 셀

### 예) 2\*3 (2행 3열) 표

```

<table>
  <tr>
    <th>1행 1열(제목)</th>
    <td>1행 2열</td>
    <td>1행 3열</td>
  </tr>
  <tr>
    <th>2행 1열(제목)</th>
    <td>2행 2열</td>
    <td>2행 3열</td>
  </tr>
</table>

```

1행 1열(제목)	1행 2열	1행 3열
2행 1열(제목)	2행 2열	2행 3열

# 표를 만드는 태그

**colspan, rowspan 속성 – 행 또는 열 합치기**


colspan="3"


rowspan="2"

```
<table border="1">
```

```
  <colgroup>
```

```
    <col style="width:100px;">
    <col style="width:300px;">
    <col style="width:100px;">
    <col style="width:300px;">
```

```
  </colgroup>
```

```
  <tr>
```

```
    <th>이름</th>
```

```
    <td></td>
```

```
    <th>연락처</th>
```

```
    <td></td>
```

```
  </tr>
```

```
  <tr>
```

```
    <th>주소</th>
```

```
    <td colspan="3"></td>
```

```
  </tr>
```

```
  <tr>
```

```
    <th>자기소개</th>
```

```
    <td colspan="3"></td>
```

```
  </tr>
```

```
</table>
```

예제 02/table-2.html

이름		연락처	
주소			
자기 소개			

# 표를 만드는 태그

## <caption> – 표 제목

제목이 위쪽 중앙에 표시됨

```
<table border="1">
  <caption>
    <strong>Modern Browser</strong>
    <p>국내에서 자주 사용하는 모던 브라우저</p>
  </caption>
  <tr> ..... </tr>
  <tr> ..... </tr>
  <tr> ..... </tr>
  <tr> ..... </tr>
  <tr> ..... </tr>
</table>
```

브라우저	제조업체	다운로드
크롬(Chrome)	Google	<a href="https://www.google.com/chrome/">https://www.google.com/chrome/</a>
파이어폭스(Firefox)	Mozilla	<a href="https://www.mozilla.org/ko/firefox/">https://www.mozilla.org/ko/firefox/</a>
엣지(Edge)	Microsoft	<a href="https://www.microsoft.com/ko-kr/windows/microsoft-edge">https://www.microsoft.com/ko-kr/windows/microsoft-edge</a>

## <figure>, <figcaption> – 표 제목

<figcaption> 위치에 따라 표의 위나 아래에 제목 표시

```
<figure>
  <figcaption>
    <p>국내에서 자주 사용하는 <b>모던 브라우저</b></p>
  </figcaption>
  <table border="1">
    <tr> ..... </tr>
    <tr> ..... </tr>
    <tr> ..... </tr>
    <tr> ..... </tr>
    <tr> ..... </tr>
  </table>
</figure>
```

국내에서 자주 사용하는 모던 브라우저

브라우저	제조업체	다운로드
크롬(Chrome)	Google	<a href="https://www.google.com/chrome/">https://www.google.com/chrome/</a>
파이어폭스(Firefox)	Mozilla	<a href="https://www.mozilla.org/ko/firefox/">https://www.mozilla.org/ko/firefox/</a>
엣지(Edge)	Microsoft	<a href="https://www.microsoft.com/ko-kr/windows/microsoft-edge">https://www.microsoft.com/ko-kr/windows/microsoft-edge</a>

# 표를 만드는 태그

## <thead>, <tbody>, <tfoot> – 표 구조 정의하기

- 표의 구조를 제목 부분과 실제 본문 그리고 요약 부분이 있는 부분으로 나눈다.
- <thead>, <tbody>, <tfoot> 태그 사용
- 시각 장애인도 화면 판독기를 통해 표의 구조를 쉽게 이해 할 수 있다.
- 표의 본문이 길 경우 자바스크립트를 이용하면 제목과 바 닥 부분을 고정하고 본문만 스크롤되도록 할 수 있다.

요안도라 객실

방 이름	대상	크기	가격
유채방	여성 도미토리	4인실	1인 20,000원
동백방	동성 도미토리		
천혜향방	가족 1팀		
바깥채 전체를 렌트합니다			

기본형

```
<thead>
<tr> ... </tr>
</thead>
<tbody>
<tr> ... </tr>
</tbody>
<tfoot>
<tr> ... </tr>
</tfoot>
```

또는

```
<thead>
<tr> ... </tr>
</thead>
<tfoot>
<tr> ... </tr>
</tfoot>
<tbody>
<tr> ... </tr>
</tbody>
```

# 표를 만드는 태그

## <col>, <colgroup> – 열끼리 묶어 스타일 지정

- <col> 태그 - 한 열에 있는 모든 셀에 같은 스타일을 적용하고자 할 때 사용. 닫는 태그 없음.
- <col> 태그에 span 속성을 사용해 여러 열을 묶을 수 있음.
- <colgroup> 태그로 여러 열을 묶을 수도 있는데, <colgroup> 태그 안에 묶는 열의 개수만큼 <col> 태그를 사용. 닫는 태그 있음.
- <col> 태그와 <colgroup> 태그는 <caption> 태그 뒤에, <tr>, <td> 태그 전에 사용해야 함.
- <colgroup> 태그 안에 있는 <col> 태그를 비롯해 단독으로 사용한 <col> 태그의 개수와 표의 열의 개수가 같아야 함.

```

<table border="1">
  <caption>colgroup 연습</caption>
  <colgroup>
    <col>
    <col span="2" style="background-color:blue;">
    <col style="background-color:yellow">
  </colgroup>
  <tr>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
  </tr>
  <tr>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
  </tr>
</table>
  
```

colgroup 연습

[실습] 온라인 프로필 – 표 추가하기

# 03. 이미지와 하이퍼링크

03-1 이미지



03-2 링크 만들기

A screenshot of a browser displaying a website. The page includes a header with a logo and navigation links, a main content area with a large image, and a footer with social media icons and text. The browser's developer tools are open, showing the HTML and CSS code for the page.

```
<ul id="main-nav">
    <li><a href="#intro">끼제 소개</a></li>
    <li><a href="#map">오시는 길</a></li>
    <li><a href="#choice">이 달의 추천</a></li>
</ul>
```

```
body {
    margin: 0;
    padding: 0;
    font-family: Arial, sans-serif;
}

#header {
    background-color: #f1f1f1;
    padding: 10px;
}

#header .logo {
    width: 100px;
    height: 50px;
    background: url("logo.png");
    display: block;
    margin-bottom: 10px;
}

#header .nav {
    text-align: right;
}
```

```
#choice {
    text-align: center;
    margin-top: 20px;
}

#choice h2 {
    font-size: 1.5em;
    margin-bottom: 10px;
}

#choice p {
    font-size: 0.9em;
    color: #777;
    margin-bottom: 20px;
}

#choice .photo {
    float: right;
    margin-right: 50px;
}

#choice .text {
    float: left;
    margin-right: 20px;
}

#choice .text p {
    font-size: 0.8em;
    color: #777;
}

#choice .text .link {
    color: #0070C0;
    text-decoration: none;
}

#choice .text .link:hover {
    color: #0056B3;
}
```

```
.footer {
    clear: both;
    text-align: center;
    margin-top: 20px;
}

.footer p {
    font-size: 0.8em;
    color: #777;
    margin-bottom: 10px;
}
```

03-3 SVG 이미지

## 웹 문서와 이미지

웹 페이지에서 사용할 수 있는 이미지 파일은 파일 크기가 크지 않으면서도 화질은 좋게 유지해야 하기 때문에 몇 가지 파일 형식만 사용할 수 있다.

파일 형식	설명
GIF(Graphic Interchange Form)	표시할 수 있는 색상 수가 최대 256가지뿐이지만 다른 이미지 파일 형식에 비해 파일 크기가 작기 때문에 아이콘이나 블릿 등 작은 이미지에 주로 사용합니다. 투명한 배경이나 움직이는 이미지를 만들 수 있다는 장점이 있습니다.
JPG/JPEG(Joint Photographic Experts Group)	사진을 위해 개발된 형식으로 GIF에 비해 다양한 색상과 명암을 표현할 수 있습니다. 저장을 반복하다 보면 화질이 떨어질 수도 있습니다.
PNG(Portable Network Graphics)	투명 배경을 만들면서 다양한 색상도 표현할 수 있으며 네트워크용으로 개발되었기 때문에 최근 많이 사용합니다.

## <img> 태그

웹 문서에 이미지를 삽입할 때 사용

기본형 

사려니 숲길
예제 03/img-1.html



```
<h1>사려니 숲길</h1>

```

## <img> 태그의 속성

- src 속성 : 이미지 파일 경로
  - 웹 문서 파일의 위치를 기준으로 이미지 경로 지정
  - 웹 사이트에 있는 이미지도 주소를 알아내어 src 속성에 사용할 수 있다.

```

```

```

```

- alt 속성 : 이미지를 설명하는 대체 텍스트
  - 대체 텍스트를 화면 낭독기가 읽어 줌.
  - 이미지를 표시할 수 없는 상황일 때 대체 텍스트 표시

```

```



- width, height 속성 : 이미지 크기 조정하기
  - 이 속성을 사용하지 않으면 원래 이미지 크기대로 표시
  - 이 속성을 이용해 화면에 표시하는 이미지 크기 조정,  
but 이미지 파일의 용량은 그대로

사려니 숲길

예제 03/img-2.html

사려니 숲길

예제 03/img-2.html

<h1>사려니 숲길</h1>

 원래 크기 이미지

 크기 조절한 이미지

1. src 2. height

03-1

# 이미지

## <figure>, <figcaption> - 이미지에 설명글 붙이기

- <figure> 태그 : 설명글을 붙일 대상 지정

기본형 <figure> 요소 </figure>

- 웹 문서 안에서 한 단위가 되는 요소를 묶을 때 사용
- 설명 글을 표시할 때 <figure>로 먼저 묶어야 함

- <figcaption> 태그 : 이미지를 설명하는 대체 텍스트

기본형 <figcaption> 설명 글 </figcaption>

- 대체 텍스트를 화면 낭독기가 읽어 줌.
- 이미지를 표시할 수 없는 상황일 때 대체 텍스트 표시

### 제주의 오름

거인 철큰대칼王爷이 제주도와 육지 사이에 다리를 놓으려고 치마쪽에 풀을 막아 나를 때 지마 둠새로 한중씩 물어진 풀영이들이 오름이 되었다는 전설이 있다.



완만하고 부드러운 용눈이 오름

```
<figure>
  
  <figcaption>완만하고 부드러운 용눈이 오름</figcaption>
</figure>
```

[실습] 온라인 프로필 - 이미지 삽입하기

## 하이퍼링크

- 다른 문서, 혹은 다른 사이트로 바로 연결해 주는 기능
- 외부 사이트나 외부 페이지로도 연결.
- 메뉴 뿐만 아니라 원하는 곳에 링크를 만들 수 있다.

## <a> 태그, href 속성

```
기본형 <a href="링크할 주소" [속성="속성 값"]> 텍스트 </a>
          <a href="링크할 주소" [속성="속성 값"]> </a>
```

반드시 href 속성을 함께 사용해서 어떤 대상으로 연결하는지 알려주어야 한다

## <a> 태그에서 사용할 수 있는 속성

속성	설명
href	링크한 문서나 사이트의 주소를 입력합니다.
target	링크한 내용이 표시될 위치(현재 창 또는 새 창)를 지정합니다.
download	링크한 내용을 보여 주는 것이 아니라 다운로드합니다.
rel	현재 문서와 링크한 문서의 관계를 알려줍니다.
hreflang	링크한 문서의 언어를 지정합니다.
type	링크한 문서의 파일 유형을 알려줍니다.

## target 속성 – 새 탭에서 링크 열기

- 다른 사이트로 링크하거나 현재 페이지를 유지한 상태에서 링크 페이지를 표시할 때
- 현재 페이지는 그대로 유지하면서 새 창이나 새 탭에 표시

사용할 수 있는 값

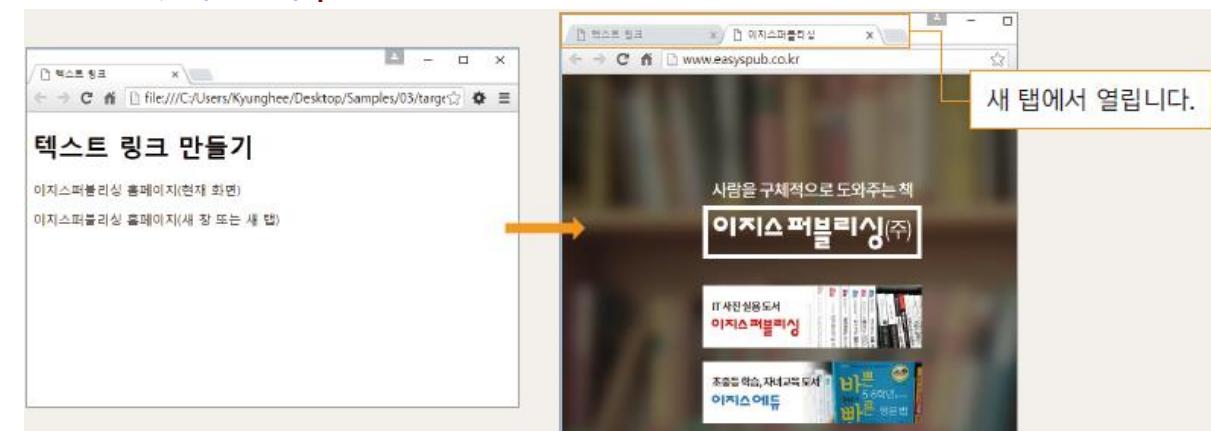
속성 값	설명
_blank	링크 내용이 새 창이나 새 탭에서 열립니다.
_self	target 속성의 기본값으로 링크가 있는 화면에서 열립니다.★
_parent	프레임을 사용했을 때 링크 내용을 부모 프레임에 표시합니다.
_top	프레임을 사용했을 때 프레임을 벗어나 링크 내용을 전체 화면에 표시합니다.

<h1>텍스트 링크 만들기</h1>

<p><a href="http://www.easyspub.com">이지스퍼블리싱 홈페이지(현재 화면)</a></p>

<p><a href="http://www.easyspub.com"

target="\_blank">이지스퍼블리싱 홈페이지(새 창 또는 새 탭)</a></p>



## 한 페이지 안에서 점프하기 - 앵커

- 앵커를 사용하려면 우선 이동하고 싶은 위치마다 id 속성을 이용해 앵커를 만듦.
- <a> 태그의 href 속성을 사용해 링크함. 단, 앵커 이름 앞에 #을 붙여 앵커 표시.

**기본형**

```
<태그 id="앵커 이름"> 텍스트 또는 이미지 </태그>
<a href="#앵커 이름"> 텍스트 또는 이미지 </a>
```

### [실습] 온라인 프로필 – 링크 추가하기

### [실습] 인터넷에 내 온라인 프로필 올리기

**앵커 만들기**

웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한 번에 이동하는 링크를 만들 수 있는데, 이 기능을 앵커(anchor)라고 합니다.

- 메뉴1
- 메뉴2
- 메뉴3

**내용1**

웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한 번에 이동하는 링크를 만들 수 있는데, 이 기능을 앵커(anchor)라고 합니다.

웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한 번에 이동하는 링크를 만들 수 있는데, 이 기능을 앵커(anchor)라고 합니다.

웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한 번에 이동하는 링크를 만들 수 있는데, 이 기능을 앵커(anchor)라고 합니다.

target을 지정하지 않을 때

```
<ul id="menu"> <
    <li><a href="#content1">메뉴1</a></li>
    <li><a href="#content2">메뉴2</a></li>
    <li><a href="#content3">메뉴3</a></li>
</ul>
<h2 id="content1">내용1</h2> <-
<p>웹 문서가 너무 길 경우 ..... 앵커(anchor)라고 합니다. </p>
<p><a href="#menu">[메뉴로]</a></p>
<h2 id="content2">내용2</h2> <-
<p>웹 문서가 너무 길 경우 ..... 앵커(anchor)라고 합니다. </p>
<p><a href="#menu">[메뉴로]</a></p>
<h2 id="content3">내용3</h2> <-
<p>웹 문서가 너무 길 경우 ..... 앵커(anchor)라고 합니다. </p>
<p><a href="#menu">[메뉴로]</a></p>
```

**이 기능을 '앵커(anchor)'라고 합니다.**

웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한 번에 이동하는 링크를 만들 수 있는데, 이 기능을 앵커(anchor)라고 합니다.

**[메뉴로]**

**내용3**

웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한 번에 이동하는 링크를 만들 수 있는데, 이 기능을 앵커(anchor)라고 합니다.

웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한 번에 이동하는 링크를 만들 수 있는데, 이 기능을 앵커(anchor)라고 합니다.

웹 문서가 너무 길 경우 필요한 곳마다 문서 안에 이름을 붙여놓고 그 위치로 한 번에 이동하는 링크를 만들 수 있는데, 이 기능을 앵커(anchor)라고 합니다.

**[메뉴로]**

target을 지정하지 않을 때

예제 03/anchor.html



## <area> 태그, usemap 속성 - 이미지맵

이미지맵 : 한 이미지 상에 여러 다른 링크를 만드는 것

```
기본형 <map name="맵이름">
    <area>
    <area>
    ...
</map>


```

### <area> 태그의 속성들

속성	설명		
alt	대체 텍스트를 지정합니다.		
coords	링크로 사용할 영역을 시작 좌표와 끝 좌표를 이용해 지정합니다.		
download	링크를 클릭했을 때 링크 문서를 다운로드합니다.		
href	링크 문서(사이트) 경로를 지정합니다.		
media	링크 문서(사이트)를 어떤 미디어에 최적화시킬지 지정합니다.		
rel	현재 문서와 링크 문서 사이의 관계를 지정합니다.	속성 값	alternate, bookmark, help, license, next, nofollow, noreferer, prefetch, prev, search, tag
shape	링크로 사용할 영역의 형태를 지정합니다.	속성 값	default, rect, circle, poly
target	링크를 표시할 대상을 지정합니다.	속성 값	_blank, _parent, _self, _top, 프레임 이름
type	링크 문서의 미디어 유형을 지정합니다.		

```


<map name="favorites">
    <area shape="rect" coords="10,10,160,200"
href="http://cafe.naver.com/doithml5" target="_blank"
alt="do it html5 네이버 카페로 가기">
    <area shape="rect" coords="220,10,380,200"
href="http://www.facebook.com/do.it.htm15"
target="_blank" alt="do it html5 페이스북 페이지로 가기">
</map>
```



## SVG 파일 형식이란

- Scalable Vector Graphics(스케일 러블 벡터 그래픽)의 약자
- 이미지를 확대하거나 축소하더라도 테두리가 원래의 깨끗한 상태로 유지되는 벡터 이미지  
(참고: jpg나 png 등의 이미지는 확대/축소하면 테두리가 유통불통해지는 비트맵 이미지)
- 로고나 아이 콘에서 많이 사용됨.
- 데이터 시각화에서 차트나 다이어그램, 지도 등을 구현할 때도 많이 사용됨.  
(예, [d3.js](#) / [Raphael.js](#) 등의 시각화 라이브러리)

- <img> 태그를 이용해 파일 형태로 삽입할 수도 있고,
- 태그를 이용해 직접 만들 수도 있음.
- 일러스트레이터나 포토샵에서 제작, 편집할 수 있음.

<img> 태그로 svg 파일 삽입

SVG 이미지 삽입하기

```
<h1>SVG 이미지 삽입하기</h1>

```



편집기에서 svg 파일 내용 확인



```
automobile.svg - Visual Studio Code
파일(F) 편집(E) 보기(V) 이용(G) 도움말(H)
automobile.svg ✘
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE svg PUBLIC '-//W3C//DTD SVG 1.1//EN' 'http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd'>
3 <svg style="shape-rendering:geometricPrecision;text-rendering:geometricPrecision;image-rendering:optimizeQuality" viewBox="0 0 20000 10000">
4   <defs>
5     <style type="text/css">
6       <![CDATA[
7         .str0 {stroke:#373435;stroke-width:20}
8         .fill {fill:#4B4B4D}
9         .fil2 {fill:#373435}
10        .fil4 {fill:#E0D237}
11        .file {fill:#201E1E}
12        .fil9 {fill:url(#id0)}
13        .fil6 {fill:url(#id1)}
```

# 04. 폼 관련 태그들

폼 만들기

<input> 태그

<input> 태그의 다양한 속성

여러 데이터 나열해서 보여주기

기타 다양한 폼 요소들



A screenshot of a Visual Studio Code editor window. The left side shows a file tree with files like 'index.html', 'index.css', and 'index.js'. The main area shows the following code:

```
index.html
<!DOCTYPE html>
<html>
  <head>
    <title>My Times with Coffee</title>
  </head>
  <body>
    <div id="container">
      <header>
        <nav>
          <ul id="main-nav">
            <li><a href="#intro">끼리 소개</a></li>
            <li><a href="#map">오시는 길</a></li>
            <li><a href="#choice">이 달의 추천</a></li>
          </ul>
        </nav>
      </header>
      <section id="intro">
        <div class="page-title">
          <h1>끼리 소개</h1>
        </div>
        <div class="content">
          
        </div>
      </section>
    </div>
  </body>
</html>
```

The status bar at the bottom indicates the file is 'index.html' and shows tabs for 'HTML', 'CSS', and 'JavaScript'.

## 웹에서 만나는 폼

- 사용자가 웹 사이트로 정보를 보낼 수 있는 요소들은 모두 폼 (form)
- 폼과 관련된 대부분의 작업들은 정보를 저장하거나 검색하거나 수정하는 일들
- 이런 작업은 모두 데이터베이스를 기반으로 한다.
- 아이디를 입력하는 텍스트 필드나 버튼 같은 폼의 형태를 만드는 것은 HTML 태그
- 폼에 입력한 사용자 정보를 처리하는 것은 ASP나 PHP 같은 서버 프로그래밍 이용
- 여기에서는 서버 프로그래밍에 대해서는 다루지 않고 브라우저에 표시될 폼을 만드는 태그들에 대해 살펴본다.



## 폼의 동작 방식

LOGIN

로그인을 하시면 더욱 편리하게 이용하실 수 있습니다.

아이디를 입력하세요.

비밀번호를 입력하세요.

로그인

아이디 찾기

비밀번호 찾기

- 1) 사용자가 [아이디]와 [비밀번호]에 정보 입력 → [로그인] 클릭
- 2) 사용자가 입력한 아이디와 비밀번호가 웹 서버로 보내짐
- 3) 서버는 자신이 가지고 있는 사용자 데이터베이스를 뒤져서 사용자가 보내온 아이디와 비밀번호가 서로 일치하는 정보인지 확인하고 그 결과를 브라우저에 보낸다.
- 4) 회원이라면 로그인한 후의 결과 화면이 나타나고, 아니라면 로그인 실패 화면이 나타남.

## <form> 태그

폼을 만드는 기본 태그. <form>과 </form> 사이에 여러 폼 요소 삽입

기본형 <form [속성="속성 값"]> 여러 폼 요소 </form>

```
<form action="search.php" method="post">
  <input type="text" title="검색">
  <input type="submit" value="검색">
</form>
```

## <form> 태그에서 사용하는 속성들

속성	설명	
		사용자가 입력한 내용들을 서버 쪽 프로그램으로 어떻게 넘겨줄지 지정합니다.
method	속성 값	get - 주소 표시줄에 사용자가 입력한 내용이 그대로 드러납니다. 256byte~4,096byte까지의 데이터만 서버로 넘길 수 있습니다.
		post - 대부분 이 방식을 사용합니다. 사용자의 입력을 표준 입력(standard input)으로 넘겨주기 때문에 입력 내용의 길이에 제한을 받지 않고 사용자가 입력한 내용이 드러나지 않습니다.
name	폼의 이름을 지정합니다. 한 문서 안에 여러 개의 <form> 태그가 있을 경우, 폼들을 구분하기 위해 사용합니다.	
action	<form> 태그 안의 내용들을 처리해 줄 서버 상의 프로그램을 지정합니다.	
target	<action> 태그에서 지정한 스크립트 파일을 현재 창이 아닌 다른 위치에 열도록 지정합니다.	

 검색

1. 검색어를 입력하고 [검색] 버튼을 클릭
2. 입력한 내용이 웹 서버에 있는 search.php 파일로 전송
3. 서버에서 search.php 파일 실행 후
4. 그 결과가 다시 웹 브라우저로 전달되어 화면에 표시됨

## <label> 태그

- 폼 요소에 레이블(텍스트)을 붙이는 태그
- 라디오 버튼이나 체크 박스에서 텍스트 부분을 클릭해도 라디오 버튼과 체크 박스 버튼이 선택된다.

[방법1]

기본형 <label [속성="속성 값"]> 레이블 <input ...> </label>

<label>아이디(6자 이상)<input type="text" .....></label>

[방법2]

기본형 <label for="id이름">  
    <input id="id이름" [속성="속성 값"]>  
  </label>

<label for="user-id">아이디(6자 이상)</label>  
<input type="text" id="user-id">

# 폼 만들기

## <fieldset> 태그

폼 요소를 그룹으로 묶는 태그

```
기본형 <fieldset [속성="속성 값"]> ... </fieldset>
```

## <legend> 태그

그룹으로 묶는 구역에 제목을 붙이는 태그

개인 정보

이름

메일 주소

---

로그인 정보

아이디

비밀번호

```

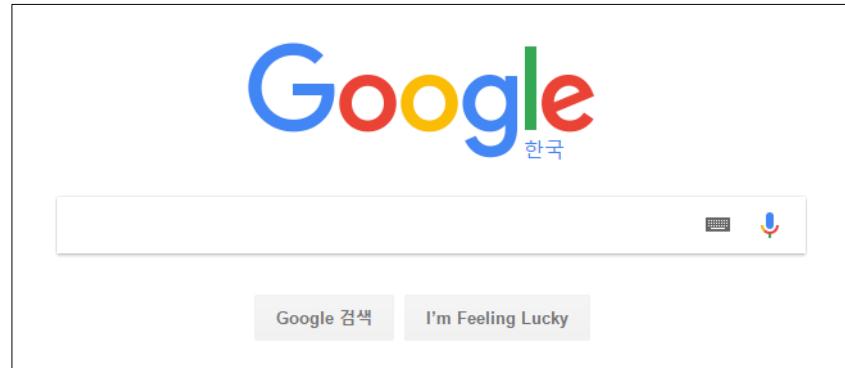
<form>
  <fieldset>
    <legend>개인 정보</legend>
    <ul>
      <li>
        <label for="name">이름</label>
        <input type="text" id="name">
      </li>
      <li>
        <label for="mail">메일 주소</label>
        <input type="text" id="mail">
      </li>
    </ul>
  </fieldset>
  <fieldset>
    <legend>로그인 정보</legend>
    <ul>
      <li>
        <label for="id">아이디</label>
        <input type="text" id="id">
      </li>
      <li>
        <label for="pwd">비밀번호</label>
        <input type="text" id="pwd">
      </li>
    </ul>
  </fieldset>
</form>

```

# <input> 태그

- 사용자가 입력하는 부분은 거의 <input> 태그를 이용해 처리
- 입력하는 내용의 종류는 <input> 태그의 type 속성을 통해 지정
- type 속성 값에 따라 함께 사용할 수 있는 속성들도 달라진다

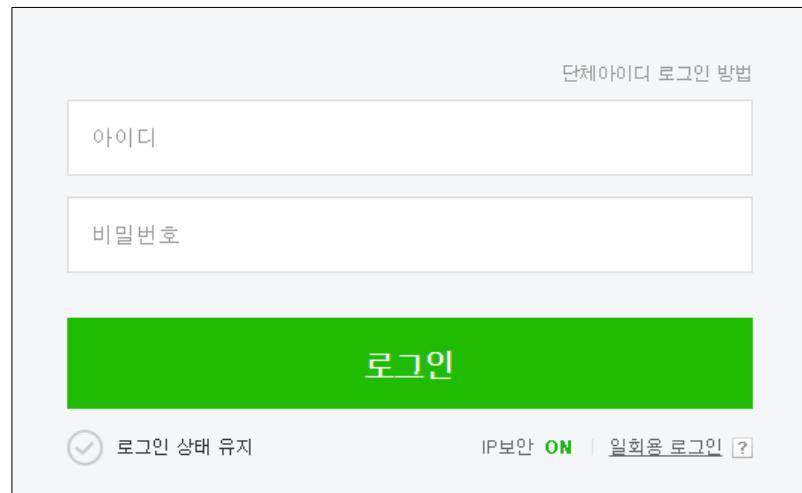
**기본형**    `<input type="유형" [속성="속성 값"]>`



## <input> 태그의 id 속성

- 여러 번 사용된 폼 요소를 구분하기 위해 사용
- <label> 태그를 이용해 캡션을 붙일 수 있다.
- CSS를 이용해 각 요소마다 다른 형태로 꾸밀 수 있다.

```
<input type="text" id="user-name" size="10">
<input type="text" id="addr" size="60">
```



# <input> 태그

유형	설명
hidden	사용자에게는 보이지 않지만 서버로 넘겨지는 값을 가집니다.
text	한 줄짜리 텍스트를 입력할 수 있는 텍스트 상자를 넣습니다.
search	검색 상자를 넣습니다.
tel	전화번호 입력 필드를 넣습니다.
url	URL 주소를 입력할 수 있는 필드를 넣습니다.
email	메일 주소를 입력할 수 있는 필드를 넣습니다.
password	비밀번호를 입력할 수 있는 필드를 넣습니다.
datetime	국제 표준시(UTC)로 설정된 날짜와 시간(연, 월, 일, 시, 분, 초, 분할 초)을 넣습니다.
datetime-local	사용자가 있는 지역을 기준으로 날짜와 시간(연, 월, 일, 시, 분, 초, 분할 초)을 넣습니다.
date	사용자 지역을 기준으로 날짜(연, 월, 일)를 넣습니다.
month	사용자 지역을 기준으로 날짜(연, 월)를 넣습니다.
week	사용자 지역을 기준으로 날짜(연, 주)를 넣습니다.

유형	설명
time	사용자 지역을 기준으로 시간(시, 분, 초, 분할 초)을 넣습니다.
number	숫자를 조절할 수 있는 화살표를 넣습니다.
range	숫자를 조절할 수 있는 슬라이드 막대를 넣습니다.
color	색상 표를 넣습니다.
checkbox	주어진 항목에서 2개 이상 선택 가능한 체크박스를 넣습니다.
radio	주어진 항목에서 1개만 선택할 수 있는 라디오 버튼을 넣습니다.
file	파일을 첨부할 수 있는 버튼을 넣습니다.
submit	서버 전송 버튼을 넣습니다.
image	submit 버튼 대신 사용할 이미지를 넣습니다.
reset	리셋 버튼을 넣습니다.
button	버튼을 넣습니다.

# <input> 태그

## type = "hidden"

- 화면 상의 폼에는 보이지 않는다.
- 폼을 서버로 전송할 때 서버로 함께 전송되는 요소

```
기본형 <input type="hidden" name="이름" value="서버로 넘길 값">
```

## type = "text"

- 한 줄짜리 텍스트 입력 필드
- 주로 아이디나 이름, 주소 등 텍스트 입력

```
기본형 <input type="text" [속성="속성 값"]>
```

## type = "password"

- 비밀번호 입력란
- 사용자가 입력하는 내용이 '\*'나 '•'로 표시된다

```
기본형 <input type="password" [속성="속성 값"]>
```

## 텍스트 필드와 패스워드 필드의 속성

(단, 패스워드 필드에는 value 속성이 없음)

속성	설명
name	텍스트 필드를 구별할 수 있도록 이름을 붙입니다.
size	텍스트 필드의 길이를 지정합니다. 즉 화면에 몇 글자가 보이도록 할 것인지를 지정합니다. 예를 들어 최대 입력 가능한 글자 수가 10개인데 size를 5로 지정하면 텍스트 필드 크기가 5개 글자 크기에 맞추어져 나머지 5개 글자는 화면에 보이지 않습니다.
value	텍스트 필드 요소가 화면에 표시될 때 텍스트 필드 부분에 표시될 내용입니다. 이 속성을 사용하지 않으면 빈 텍스트 필드가 표시됩니다.
maxlength	텍스트 필드에 입력할 수 있는 최대 문자 개수를 지정합니다.

## <input> 태그

### type = "search"

- 검색 필드
- 검색창에 x 표시가 되어 검색어 삭제 쉬움

기본형 <input type="search" [속성="속성 값"]>

### type = "url"

- 웹 주소 필드
- http://로 시작하는 사이트 주소 입력

기본형 <input type="url" [속성="속성 값"]>

### type = "email"

- 메일 주소 입력 필드
- 메일 주소 형식 자동 체크

기본형 <input type="email" [속성="속성 값"]>

### type = "tel"

- 전화번호 입력 필드
- 사용자 입력을 체크하지는 않음.

기본형 <input type="tel" [속성="속성 값"]>

# <input> 태그

```

<form>
  <fieldset>
    <legend>로그인 정보</legend>
    <ul>
      <li>
        <label for="user-id">아이디 </label>
        <input type="text" id="user-id">
      </li>
      <li>
        <label for="pwd1">비밀번호 </label>
        <input type="password" id="pwd1">
      </li>
      <li>
        <label for="pwd2">비밀번호 확인 </label>
        <input type="password" id="pwd2">
      </li>
    </ul>
  </fieldset>

```

```

<fieldset>
  <legend>개인 정보</legend>
  <ul>
    <li>
      <label for="user-name">이름 </label>
      <input type="text" id="user-name">
    </li>
    <li>
      <label for="mail">메일 주소</label>
      <input type="email" id="mail">
    </li>
    <li>
      <label for="phone">연락처</label>
      <input type="tel" id="phone">
    </li>
    <li>
      <label for="homep">블로그/홈페이지</label>
      <input type="url" id="homep">
    </li>
  </ul>
</fieldset>
<input type="submit" value="가입하기">
</form>

```

회원 가입

로그인 정보

아이디

비밀번호

비밀번호 확인

개인 정보

이름

메일 주소

연락처

블로그/홈페이지

가입하기

04-2

# <input> 태그

## type = "number"

숫자 입력 필드. 브라우저에 따라 스픈 박스로 표시됨

기본형 <input type="number" [속성="속성 값"]>

<b>주문 개수 : </b> < input type="number" min="1" max="5" value="1"> 개

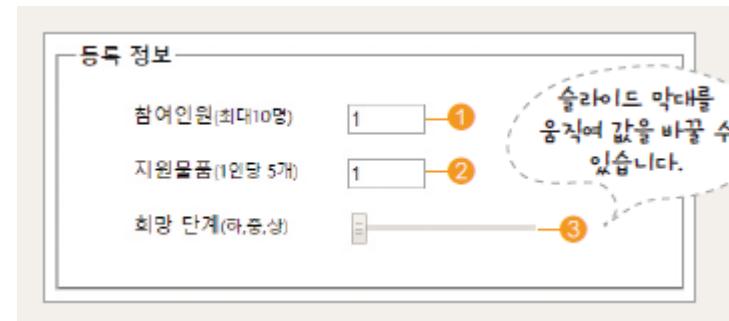


## type = "range"

숫자 입력 필드. 슬라이드 막대를 이용해 숫자 입력

기본형 <input type="range" [속성="속성 값"]>

<b>주문 개수 : </b> < input type="range" min="1" max="5" value="1"> 개



```

<ul>
  <li>
    <label class="reg" for="member">참여인원<small>(최대10명)</small></label>
    <input type="number" id="member" value="1" min="0" max="10" step="1">
  </li>
  <li>
    <label class="reg" for="stuffs">지원물품<small>(1인당 5개)</small></label>
    <input type="number" id="stuffs" value="1" min="0" max="50" step="5" size="5">
  </li>
  <li>
    <label class="reg" for="satis">희망 단계<small>(하, 중, 상)</small></label>
    <input type="range" id="satis" value="1" min="1" max="3">
  </li>
</ul>

```

# <input> 태그

## type = "radio"

여러 항목 중 하나만 선택할 때

기본형 `<input type="radio" [속성="속성 값"]>`

## type = "checkbox"

여러 항목 중 둘 이상을 선택할 때

기본형 `<input type="checkbox" [속성="속성 값"]>`

속성	설명
name	라디오 버튼이나 체크박스가 여러 개 있을 경우, 서버의 폼 프로그램에서 라디오 버튼이나 체크박스를 구분하기 위해 이름을 지정합니다. 라디오 버튼은 여러 개 중에서 한 가지만 선택하는 것이기 때문에 관련 있는 그룹끼리는 name 속성 값을 똑같이 만듭니다.
value	선택한 라디오 버튼이나 체크박스를 서버로 알려 줄 때 넘길 값을 지정합니다. 이 값은 영문이거나 숫자여야 하며 필수 속성입니다.
checked	라디오 버튼의 항목들은 처음에 아무 것도 선택되지 않은 상태로 화면에 표시되는데 기본으로 선택해 놓을 항목이 있다면 checked 속성을 사용합니다.

## type = "color"

색상 표에서 색상 선택

기본형 `<input type="color" [value="기본 색" [속성="속성 값"]]>`

```

<form>
  <fieldset>
    <legend>신청 과목</legend>
    <p>이 달에 신청할 과목을 선택하세요 (1과목만 가능)</p>
    <label><input type="radio" name="subject" value="speaking">회화</label>
    <label><input type="radio" name="subject" value="grammar">문법</label>
    <label><input type="radio" name="subject" value="writing">작문</label>
  </fieldset>
  <fieldset>
    <legend>메일링</legend>
    <p>메일로 받고 싶은 뉴스 주제를 선택해 주세요 (복수 선택 가능)</p>
    <label><input type="checkbox" name="mailing1" value="news">해외 단신</label>
    <label><input type="checkbox" name="mailing2" value="dialog">5분 회화 </label>
    <label><input type="checkbox" name="mailing3" value="pops">모닝팝스</label>
  </fieldset>
</form>

```

신청 과목  
이 달에 신청할 과목을 선택하세요 (1과목만 가능)  
회화 문법 작문

메일링  
메일로 받고 싶은 뉴스 주제를 선택해 주세요 (복수 선택 가능)  
해외 단신 5분 회화 모닝팝스

# <input> 태그

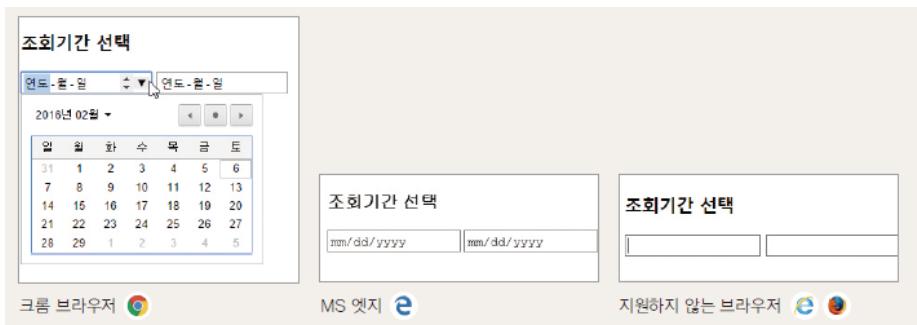
## type = "date", type="month", type="week"

달력 이용해 날짜 입력

기본형 <input type="date | month | week" [value="기본 값" 속성="속성 값"]>

유형	설명
date	날짜를 선택합니다. "yyyy-mm-dd" 형식.
month	월(month)과 연도(year)를 선택합니다. "yyyy-mm" 형식
week	주(week)와 연도(year)를 선택합니다.

```
<label><input type="date" id="start"></label>
<label><input type="date" id="end"></label>
```



## type = "time", type="datetime-local"

시간 입력

기본형 <input type="time | datetime | datetime-local" [value="기본 값" 속성="속성 값"]>

## 사용하는 속성

속성	설명
min	날짜나 시간의 최솟값을 지정합니다.
max	날짜나 시간의 최댓값을 지정합니다.
step	스핀 박스의 화살표를 누를 때마다 날짜나 시간을 얼마나 조절할지를 지정합니다.
value	화면에 표시할 초기값을 지정합니다. type="time"일 경우, 시간은 00:00부터 23:59까지 입력하고 type="datetime"이나 type="datetime-local" 유형일 경우, 날짜 다음에 키워드 T를 쓰고 24시간제로 시간을 지정합니다. 예를 들어 오후 6시를 나타내려면 T18:00라고 하면 됩니다.

```
<label>시작 시간<input type="time" value="09:00" id="start"></label>,
<label>종료 시간<input type="time" value="18:00" id="end"></label>
<label>시작 시간<input type="datetime-local" value="2016-03-02T09:00" id="start"></label>,
<label>종료 시간<input type="datetime-local" value="2016-03-02T18:00" id="end"></label>
```



# <input> 태그

## type = "submit", type="reset"

- 폼 전송/리셋 버튼
- 전송(submit) 버튼 : 사용자 입력 내용을 서버로 전송
- 리셋(reset) 버튼 : 사용자 입력 내용 전부 삭제
- value 속성을 이용해 버튼 표시 내용 지정

```
기본형 <input type="submit | reset" [value="버튼 내용"] [속성="속성 값"]>
```

```
<form action="register.php" method="post">
<label> 메일 주소 <input type="text"></label>
<input type="submit" value="제출">
<input type="reset" value="다시입력">
</form>
```

메일링 리스트 등록

메일 주소  제출

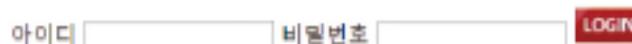
## type = "image"

submit 버튼 대신 이미지 삽입

```
기본형 <input type="image" src="경로" alt="대체 텍스트" [속성="속성 값"]>
```

<form>

```
<label>아이디 <input type="text" size="15"></label>
<label>비밀번호 <input type="password" size="15"></label>
<input type="image" id="butt" src="images/login.jpg" alt="login">
</form>
```



# <input> 태그의 다양한 속성

## autofocus

페이지를 불러오자마자 원하는 폼 요소에 마우스 커서 표시

## placeholder

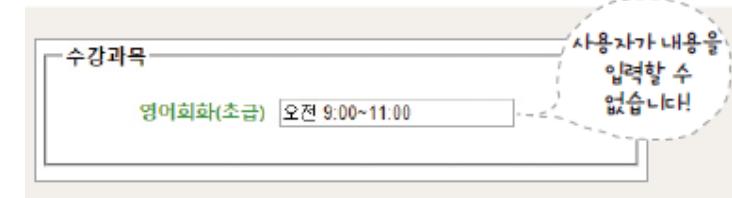
입력란에 표시하는 힌트로, 필드를 클릭하면 사라짐

```
<label class="reg" for="uname">이름
</label>
<input type="text" id="uname">
<label class="req" for="uid">학번</label>
<input type="text" id="uid" placeholder="하이픈없이 입력">
```

## readonly

- 내용을 보기만 하고 입력하지 못하게 함.
- 속성 값 없이 readonly 라고만 쓰면 됨.  
(readonly="readonly", readonly="true"로 표시하기도 함)

```
<label class="req" for="subj">영어회화(초급)</label>
<input type="text" id="subj" value="오전 9:00~11:00" readonly>
```



# <input> 태그의 다양한 속성

## required

- 필수 필드 체크
- 속성 값 없이 required 라고만 입력  
(required="required"라고 해도 됨)
- 필수 필드는 브라우저에서 직접 체크하는 것으로 오류 메시지 내용은 브라우저들마다 다르게 나타남

```
<label class="req" for="uname">이름</label>
<input type="text" id="uname" autofocus required>
```



# <input> 태그의 다양한 속성

## min, max, step

- min, max : 해당 필드의 최솟값, 최댓값
- step : 허용된 범위 내의 숫자 간격
- type이 date, datetime, datetime-local, month, week, time, range, number 일 경우에만 사용

```
<label class="reg" for="group">단체주문</label>
<input type="number" id="qgroup" value="10" min="10" max="100" step="10">
```

## size, minlength, maxlength

- size : 텍스트 관련 필드에서 화면에 몇 글자까지 보이게 할지 결정
- maxlength : 입력 가능한 최대 글자
- minlength : 입력해야 할 최소 글자.  
(크롬과 안드로이드 브라우저에서만 지원)

```
<label class="req" for="uid">학번</label>
<input type="text" id="uid" placeholder="하이픈없이 입력" maxlength="8" required>
```

# 여러 데이터 나열해 보여주기

## <select>, <optgroup>, <option>

- 여러 옵션 중에서 선택 – 드롭다운 목록
- 공간을 최소한으로 사용하면서 여러 옵션 표시 가능

```
기본형 <select 속성="속성 값">
    <option value="값" [속성="속성 값"]> 내용1 </option>
    <option value="값" [속성="속성 값"]> 내용2 </option>
    <option value="값" [속성="속성 값"]> 내용3 </option>
    ...
</select>
```

```
<select id="class">
    <option value="archi">건축공학과</option>
    <option value="mechanic">기계공학과</option>
    <option value="indust">산업공학과</option>
    <option value="elec">전기전자공학과</option>
    <option value="computer" selected>컴퓨터공학과</option>
    <option value="chemical">화학공학과</option>
</select>
```



### <select> 태그의 속성

속성	설명
size	화면에 표시될 드롭다운 메뉴의 항목 개수를 지정합니다. ▶ 크롬 브라우저의 경우, size에서 지정한 개수보다 하나 더 많은 옵션이 표시됩니다. size="3" 으로 지정하면 4개의 옵션이 표시됩니다.
multiple	브라우저 화면에 여러 개의 옵션이 함께 표시되면서 Ctrl 키를 누른 상태로 드롭다운 메뉴에 있는 여러 항목을 선택할 수 있습니다.

### <option> 태그의 속성

속성	설명
value	옵션을 선택했을 때 서버로 넘겨질 값을 지정합니다.
selected	화면에 표시될 때 기본으로 선택되어 있는 옵션을 지정합니다.

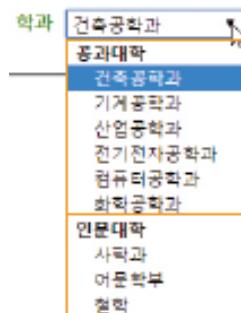
# 여러 데이터 나열해 보여주기

## <select>, <optgroup>, <option>

### <optgroup> 태그

- 여러 항목을 그룹을 묶을 때 사용
- label 속성을 사용해 그룹 제목을 붙임

```
<select id="class">
  <optgroup label="공과대학">
    <option value="archi">건축공학과</option>
    <option value="mechanic">기계공학과</option>
    <option value="indust">산업공학과</option>
    <option value="elec">전기전자공학과</option>
    <option value="computer">컴퓨터공학과</option>
    <option value="chemical">화학공학과</option>
  </optgroup>
  <optgroup label="인문대학">
    <option value="history">사학과</option>
    <option value="lang">어문학부</option>
    <option value="philo">철학</option>
  </optgroup>
</select>
```



## <datalist>, <option>

- 데이터 목록에 제시한 값 중에서 선택하면 그 값이 자동으로 텍스트 필드에 입력됨
- 데이터 목록에 id를 이용해 이름을 붙이고,
- <input> 태그의 list 속성에 데이터 목록 id를 지정함.

```
기본형 <input type="text" list="데이터 목록 id">
          <datalist id="데이터 목록 id">
            <option> ..... </option>
            <option> ..... </option>
            ...
          </datalist>
```

### <option> 태그의 속성

속성	설명
value	사용자가 레이블을 선택했을 때 서버로 넘겨질 값을 지정합니다.
label	사용자를 위해 브라우저에 표시할 레이블입니다. 따로 지정하지 않을 경우, value값을 레이블로 사용합니다.

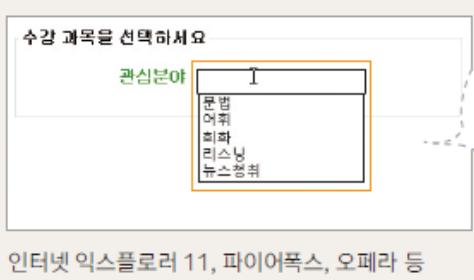
# 여러 데이터 나열해 보여주기

## <datalist>, <option>

```
<input type="text" id="interest" list="choices">
<datalist id="choices">
  <option value="grammar" label="문법"></option>
  <option value="voca" label="어휘"></option>
  <option value="speaking" label="회화"></option>
  <option value="listening" label="리스닝"></option>
  <option value="news" label="뉴스청취"></option>
</datalist>
```



크롬



인터넷 익스플로러 11, 파이어폭스, 오페라 등

## <textarea>

- 텍스트 영역 – 여러 줄의 텍스트 입력
- 게시판의 게시물 입력창, 회원 가입 양식의 약관 등

기본형 <textarea [속성="속성 값"]> 내용 </textarea>

## <textarea> 태그의 속성

속성	설명
name	다른 폼 요소와 구분하기 위해 텍스트 영역의 이름을 지정합니다.
cols	텍스트 영역의 가로 너비를 문자 단위로 지정합니다.
rows	텍스트 영역의 세로 길이를 줄 단위로 지정합니다. 지정한 숫자보다 줄 개수가 많아지면 스크롤 막대가 생깁니다.

# 기타 다양한 폼 요소들

## <button>

- 다양한 형태의 버튼 삽입
- 화면 낭독기에서 버튼임을 정확히 전달할 수 있음
- CSS를 이용해 원하는 형태로 꾸밀 수 있음

기본형 <button [type="submit | reset | button"] 내용 </button>

속성 값	설명
submit	폼을 서버로 전송합니다. <button type="submit">전송하기</button>과 같이 사용합니다.*
reset	폼에 입력한 모든 내용을 초기화시킵니다. <button type="reset">다시 쓰기</button>과 같이 사용합니다.
button	버튼 형태만 만들 뿐 자체 기능은 없습니다. <button type="button">주소 입력</button>과 같이 사용합니다.

<input type="submit" value="전송하기" > [전송하기]  
 <button type="submit">전송하기</button> [전송하기]

```

<style>
.subm{ /* 버튼 스타일 */
  display:block; /* 블록 레벨 요소 */
  background-color:#fff; /* 배경색 */
  border:1px solid #dedede; /* 테두리 */
  cursor:pointer; /* 마우스 포인터 */
  padding:5px 10px 6px 7px; /* 패딩 */
}
.subm img{ /* 버튼 내 왼쪽 이미지 */
  border:0; /* 테두리 없음 */
  padding:0; /* 패딩 없음 */
  width:16px; /* 가로 크기 */
  height:16px; /* 세로 크기 */
}
.subm:hover{ /* 버튼 위로 마우스 포인터 올렸을 때 스타일 */
  background-color:#e6efc2; /* 배경색 */
  border:1px solid #c6d880; /* 테두리 */
  color:#529215; /* 글자색 */
}
</style>
  
```

```

<form>
  <button type="submit" class="subm">
     전송하기
  </button>
</form>
  
```



## <output>

- 계산 결과를 브라우저에 표시
- 브라우저 화면에 다르게 표시되는 것은 아니지만

<output> 태그로 묶인 부분이 일반 텍스트가 아니라 계산의 결과값이라는 점을 웹 브라우저가 정확히 인식할 수 있습니다.

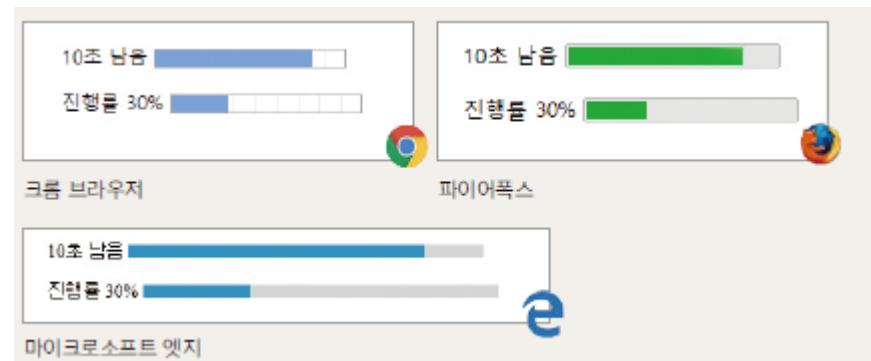
```
<form oninput="result.value=parseInt(num1.value)+  
parseInt(num2.value)">  
  <input type="number" name="num1" value="0">  
  +<input type="number" name="num2" value="0">  
  =<output name="result" for="num"></output>  
</form>
```

## <progress>

- 작업 진행 상태를 브라우저에 표시
- 값에는 특별한 단위가 없고 단위를 표시하지도 않음

속성	설명
value	작업 진행 상태를 나타내며 부동 소수점으로 표현합니다. 이 값은 0보다 크거나 같고 max 값보다 작거나 같아야 합니다. 만약 max 값이 지정되지 않았다면 이 값은 1.0보다 작아야 합니다.
max	작업이 완료되려면 얼마나 많은 작업을 해야 하는지 부동 소수점으로 표현합니다. 이 값은 0보다 커야 합니다.

```
<label>10초 남음</label>  
<progress value="50" max="60"> </progress>  
  
<label>진행률 30%</label>  
<progress value="30" max="100"></progress>
```



## <meter>

- 전체 크기 중에서 얼마나 차지하는지를 표현할 때 사용  
예) 하드 디스크 사용량, 유권자 투표율 등

### 사용할 수 있는 속성

속성	설명
min, max	범위의 최솟값과 최댓값을 나타냅니다. 값을 정하지 않으면 0과 1로 간주합니다.
value	범위 내에서 차지하는 값을 나타냅니다.
low	"이 정도면 낮다."라고 할 정도의 값을 지정합니다.
high	"이 정도면 높다."라고 할 정도의 값을 지정합니다.
optimum	"이 정도면 적당하다."라고 할 정도의 범위를 지정합니다. optimum값이 high 값보다 크다면 value 값이 클수록 좋고 optimum 값이 low 값보다 작다면 value 값이 작을수록 좋습니다.

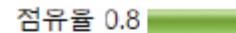
```
<label>점유율 0.8 </label>
<meter value="0.8"></meter>
```

```
<label>사용량 64%</label>
<meter min="0" max="100" value="64"></meter>
```

```
<label>트래픽 초과</label>
<meter min="1024" max="10240" low="2048" high="8192"
value="9216"></meter>
```

```
<label>적절한 트래픽</label>
<meter value="0.5" optimum="0.8"></meter>
```

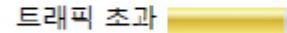
점유율 0.8



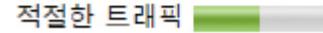
사용량 64%



트래픽 초과



적절한 트래픽

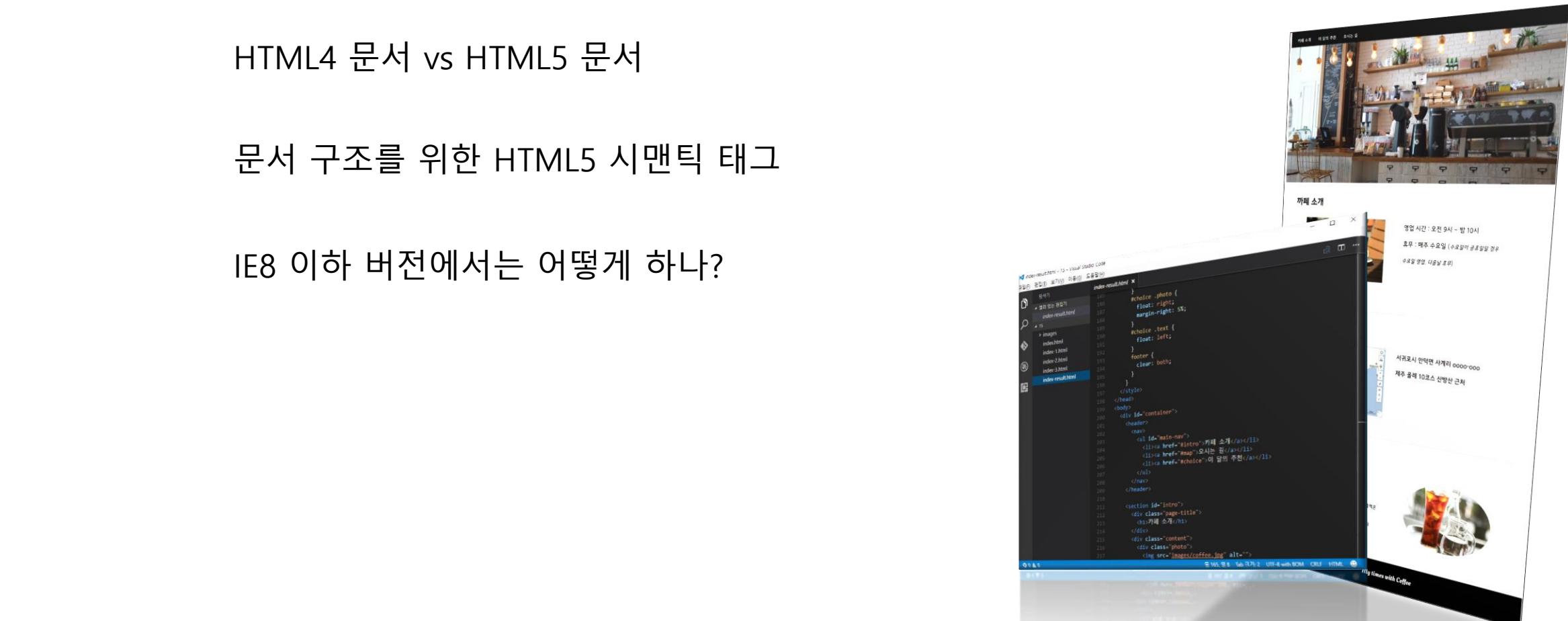


# 5. HTML5와 시맨틱 태그

HTML4 문서 vs HTML5 문서

문서 구조를 위한 HTML5 시맨틱 태그

IE8 이하 버전에서는 어떻게 하나?



# HTML4 문서 vs HTML5 문서

## HTML4 문서

화면을 구성하는 역할은 대부분 `<div>` 태그에게 맡겨 놓았고, 수많은 `<div>` 태그들을 구분하는 것은 id 속성을 이용했다.

```
<div id="header"> Header </div>
<div id="content"> Content </div>
<div id="footer"> Footer </div>
```

`<div>` 태그로 묶기만 하면 되었고 그 이름에는 제약이 없었기 때문에 홈페이지 제작자 자신이 나중에 보더라도 이해하기 어려웠을 뿐 아니라, 담당자가 바뀌게 되면 홈페이지 구조를 이해하는 데 많은 시간이 걸리곤 했다.

실제로 웹사이트를 만들 때는 수천 줄의 소스를 일일이 살펴보면서 헤더 부분이 어디인지, 메뉴는 어디에 있는지, 사이드 바 내용이 있는 부분은 어디인지 등 찾는 것이 쉽지 않다.

## HTML5 문서



HTML5의 웹 표준 문서 구조

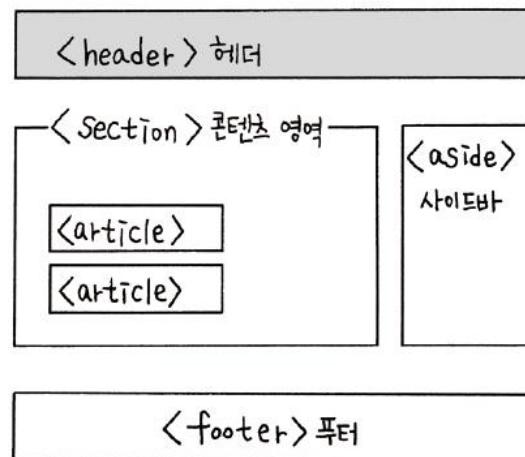
WHATWG에서는 웹 문서의 레이아웃을 어느 정도 표준화할 수 있겠다고 생각하고, 레이아웃을 만들 때 사용하는 주요 태그를 미리 약속해 놓고 해당 태그만 보면 어느 부분이 헤더이고 어느 부분이 실제 내용인지 쉽게 알 수 있도록 했다.

→ 표준화된 시맨틱 태그를 이용해 웹 문서를 작성하면 어떤 기기에서도 문서 구조를 정확히 이해할 수 있다.

# 문서 구조를 위한 HTML5 시맨틱 태그

## <header> 태그 – 머리말 지정하기

- 사이트 전체의 제목 부분이 될 수도 있고, 본문의 제목 부분이 될 수도 있다.
- 주로 페이지 맨 위쪽이나 왼쪽에 삽입
- <form> 태그를 사용해 검색 창을 넣거나 <nav> 태그를 사용해 사이트 메뉴를 넣음



# 문서 구조를 위한 HTML5 시맨틱 태그

## <nav> 태그 – 내비게이션 링크

- 같은 사이트 안의 문서나 다른 사이트의 문서로 연결하는 링크를 나타냄
- 내비게이션 메뉴뿐만 아니라 푸터에 있는 사이트 링크 모음 부분에도 사용됨



```

<div class="container">
  <header>
    <h1>Joandora</h1>
    <h2>가장 제주다운 수산리집</h2>
  </header>
  <nav class="navi">
    <ul>
      <li><a href="#">이용 안내</a></li>
      <li><a href="#">객실 소개</a></li>
      <li><a href="#">예약 방법</a></li>
      <li><a href="#">예약하기</a></li>
    </ul>
  </nav>
</div>

```



# 문서 구조를 위한 HTML5 시맨틱 태그

## <section> 태그 – 주제별 콘텐츠 영역

- 문서에서 주제별로 콘텐츠를 묶을 때 사용
- 섹션 제목을 나타내는 <h<sub>n</sub>> 태그가 함께 사용됨

## <article> 태그 – 콘텐츠 내용

- 웹 상의 실제 내용
- 태그 적용 부분을 떼어내 독립적으로 배포하거나 재사용하더라도 완전히 하나의 콘텐츠

## <aside> 태그 – 본문 이외의 내용

- 본문 내용 외에 주변에 표시되는 기타 내용들
- 필수 요소가 아니므로 광고나 링크 모음 등 문서의 메인 내용에 영향을 미치지 않는 내용들을 넣을 때 사용

```

<article>
  <section class="content">
    <h2>요안도라 소개</h2>
    <h3>▣ 요안도라는 ..... 농어촌 민박입니다</h3>
    <p>성산의 날씨는 ..... 많아질듯 합니다.</p>
    <p>오늘은 사진에 보이는 ..... 달았습니다.</p>
    <p>내일은 ..... 예정입니다. <br> ..... 알리려고 합니다.</p>
    <div class="banner">
      
    </div>
  </section>
</article>
<aside class="sidebar">
  <h3>▣ 알립니다</h3>
  <p>게스트하우스 예약은 ..... 부과되지 않습니다. </p>
  
  
  
</aside>

```



# 문서 구조를 위한 HTML5 시맨틱 태그

## <iframe> 태그 – 외부 문서 삽입

웹 문서 안에 다른 외부 문서 삽입 – 인라인 프레임

기본형    <iframe src="삽입할 문서 주소" [속성 = "속성 값"]> </iframe>

속성	설명
width	인라인 프레임의 너비입니다. 픽셀이나 백분율 값으로 표시합니다.
height	인라인 프레임의 높이입니다. 픽셀이나 백분율 값으로 표시합니다.
name	인라인 프레임의 이름입니다.
src	프레임에 표시할 문서의 주소를 지정합니다.
seamless	프레임의 테두리를 없애 마치 본문의 일부처럼 보이도록 만들며 속성 값 없이 seamless라고 쓰면 됩니다. 이 속성은 아직 크롬과 사파리에서만 지원합니다.

<h2>iframe 태그를 이용해 외부 문서(사이트) 포함시키기</h2>

```
<div class="content">
  <iframe src="http://www.easyspub.co.kr/" width="95%" height="500"></iframe>
</div>
```



# 문서 구조를 위한 HTML5 시맨틱 태그

## <footer> 태그 – 제작 정보와 저작권 정보

사이트 제작자의 연락처 정보와 저작권 정보를 표시

## <address> 태그 – 제작자 정보와 연락처 정보

- 사이트 제작자의 이름이나 제작자 의 웹 페이지 또는 피드백을 위한 연락처 정보
- 웹 사이트와 관련된 우편 주소  
(단순 우편 주소는 <p> 태그 사용)



```

<div class="container">
  <header>
    <h1>Joandora</h1>
    <h2>가장 제주다운 수산리집</h2>
  </header>
  <nav class="navi">
    <ul>
      <li><a href="#">이용 안내</a></li>
      <li><a href="#">객실 소개</a></li>
      <li><a href="#">예약 방법</a></li>
      <li><a href="#">예약하기</a></li>
    </ul>
  </nav>
  <section class="content"> ... </section>
  <aside class="sidebar"> ... </aside>
  <footer>
    <address>
      <p>제주특별자치도 남제주군 성산읍 수산리 000 번지 요안도라</p>
      <p>yoan##@naver.com</p>
    </address>
    <p> Copyright ©. All rights reserved.</p>
  </footer>
</div>

```

# 문서 구조를 위한 HTML5 시맨틱 태그

## <div> 태그

주로 콘텐츠를 묶어 시각적 효과를 적용할 때 즉 콘텐츠에 CSS를 적용할 때 사용

```
<style>
#wrapper{
  width:800px;
  margin: 0 auto;
}
</style>
```

```
<body>
  <div id="wrapper">
    <header> ..... </header>
    .....
    <div>
  </body>
```



# IE8 이하 버전에서는 어떻게 하나

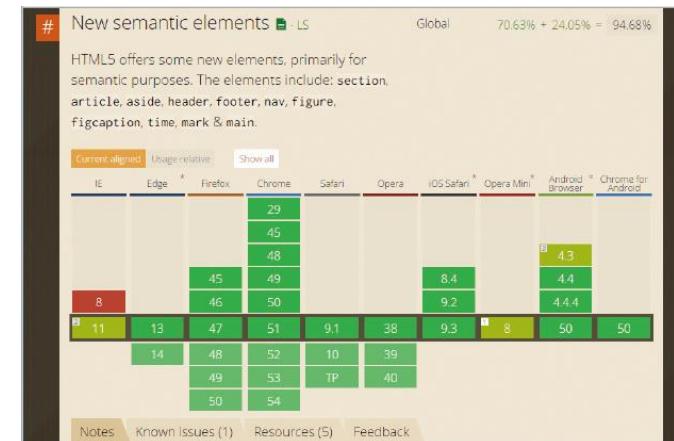
## 새로운 시맨틱 태그 지원 상황

HTML5의 새로운 시맨틱 태그는 대부분의 웹 브라우저에서는 사용할 수 있지만 IE8 이하에서는 지원하지 않음

<http://caniuse.com> 사이트 태그나 속성을 선택하면 그 항목을 모던 브라우저에서 얼마나 지원하는지 직접 확인

The screenshot shows the Can I use website interface. In the search bar, 'form' is entered. Below the search results, there's a section titled 'Index of features' with a dropdown menu set to 'HTML5'. Under 'HTML5', the 'New semantic elements' section is highlighted. Other sections shown include 'CSS', 'JS API', and 'SVG'. The 'New semantic elements' section lists: section, article, aside, header, footer, nav, figure, figcaption, time, mark, and main.

This screenshot shows the same 'Can I use' interface as the previous one, but with the 'HTML5' tab selected in the dropdown menu. The 'New semantic elements' section is now the active tab, displaying a list of semantic elements: section, article, aside, header, footer, nav, figure, figcaption, time, mark, and main.



# IE8 이하 버전에서는 어떻게 하나

## IE8 이하에서 시맨틱 태그를 사용하려면 – html5shiv

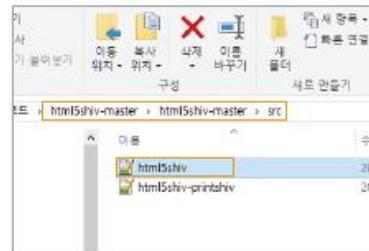
1. <https://github.com/aFarkas/html5shiv>에서 js 파일 다운로드

**Manual installation**

Download and extract the [latest zip package](#) from this repository and copy the two files `dist/html5shiv.js` and `dist/html5shiv-printshiv.js` into your project. Then include one of them into your `<head>`, as above.

**HTML5 Shiv API**

HTML5 Shiv works as a simple drop-in solution. In most cases there is no need to configure HTML5 Shiv or use methods provided by HTML5 Shiv.



2. `<head>`와 `</head>` 사이에 js 파일 삽입

```
<!-- [if lt IE 9]
  <script src="js/html5shiv.js"></script>
<![endif] -->
```

# 6. HTML5와 멀티미디어

웹과 멀티미디어

오디오 & 비디오 재생하기



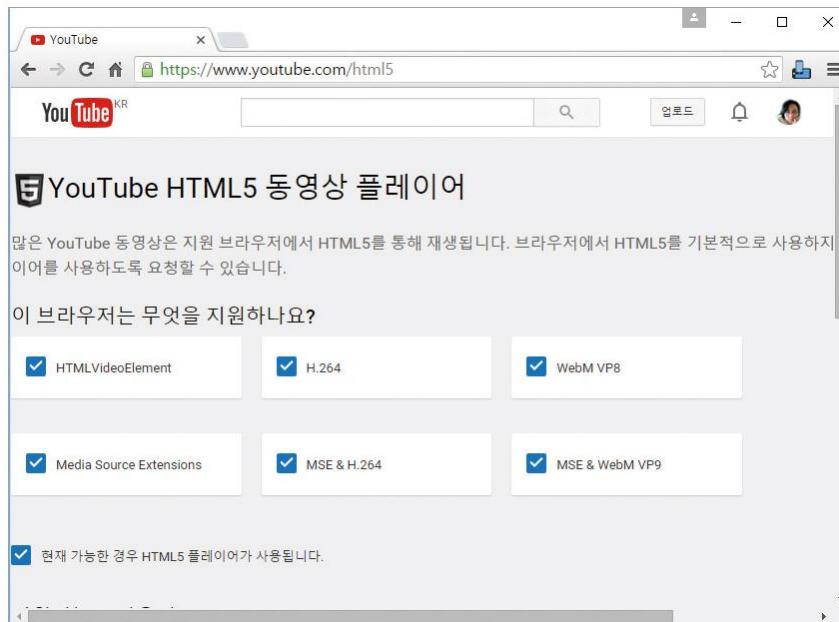
A screenshot of a code editor window showing the source code for a website. The code includes HTML structure like a header with navigation links and a main content section with a photo, along with corresponding CSS styles for styling elements like the photo and footer.

```
<html>
  <head>
    <title>My Times with Coffee</title>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@100;300;400;500;700;900&display=swap" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
    <link href="css/reset.css" rel="stylesheet" />
    <link href="css/main.css" rel="stylesheet" />
  </head>
  <body>
    <header>
      <div id="main-nav">
        <ul>
          <li><a href="#intro">My Times</a></li>
          <li><a href="#map">오시는 길</a></li>
          <li><a href="#choice">이 달의 추천</a></li>
        </ul>
      </div>
    </header>
    <section id="intro">
      <div class="page-title">
        <h1>커피 소믈리에</h1>
      </div>
      <div class="content">
        <div class="photo">
          
        </div>
      </div>
    </section>
  </body>
</html>
```

# 웹과 멀티미디어

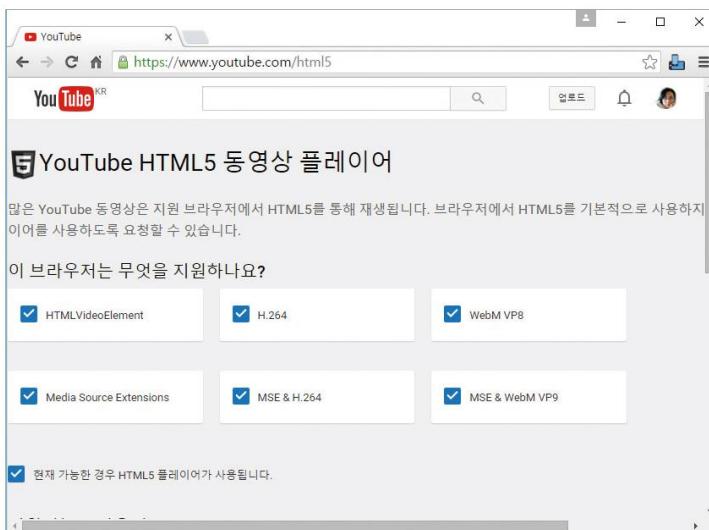
## 플러그인 프로그램

- HTML4까지는 웹 브라우저에서 멀티미디어를 직접 재생할 수 없기 때문에 플러그인 프로그램 연결해서 사용
- HTML5 웹 표준 이후 웹 브라우저에서 직접 멀티미디어 재생
- 크롬 브라우저, 마이크로소프트 엣지 브라우저 등에서는 플래시 플레이어 차단
- 유튜브나 비메오 등 대부분의 비디오 사이트에서 HTML5 플레이어 사용



## 플러그인 프로그램

- HTML4까지는 웹 브라우저에서 멀티미디어를 직접 재생할 수 없기 때문에 플러그인 프로그램 연결해서 사용
- HTML5 웹 표준 이후 웹 브라우저에서 직접 멀티미디어 재생
- 크롬 브라우저, MS 엣지 등에서는 플래시 플레이어 차단
- 유튜브나 비메오 등 비디오 사이트에서 HTML5 플레이어 사용



## <object>, <embed> - 외부 파일 삽입

- 웹 브라우저에서 직접 재생할 수 없는 자바 애플릿이나 PDF, 플래시 무비 등 삽입

**기본형**

```
<object data="경로" type="유형" name="이름" width="너비" height="높이">
</object>
```

- <object> 태그를 지원하지 않는 브라우저에서는 <embed> 태그 사용

**기본형**

```
<embed src="경로" type="유형" width="너비" height="높이">
```

# 웹과 멀티미디어

## 멀티미디어 웹 표준화

- 플러그인 프로그램 없이 웹 브라우저 자체에서 멀티미디어 재생
- 단, 브라우저마다 재생할 수 있는 멀티미디어 파일 종류가 다르다.

브라우저	비디오			오디오	
	mp4	webm	ogv	mp3	ogg
인터넷 익스플로러	O (9+)	X	X	O (9+)	X
크롬	O (all)	O (25+)	O (all)	O (all)	O (all)
파이어폭스	O (35+)	O (28+)	O (3.5+)	O (22+)	O (3.5+)
사파리	O (3.2+)	X	X	O 4+)	X
오페라	O (25+)	O (16+)	O (11.5+)	O (15+)	O (11.5+)
iOS 사파리	O (all)	X	X	O (4.1+)	X
안드로이드 브라우저	O (4.4+)	X	X	O (2.3+)	O (2.3)

최신 모던 브라우저를 사용한다면 mp4와 mp3를 사용하면 간단하지만, 사용자들의 브라우저는 다양하기 때문에 여러 종류의 파일 형식을 지정해야 한다.

웹에서 지원되는 비디오/오디오 파일에는 여러 종류가 있습니다. 뒤이어 배울 코덱으로 비디오/오디오 파일을 제작할 수 있으며 파일 확장자도 바꿀 수 있습니다.

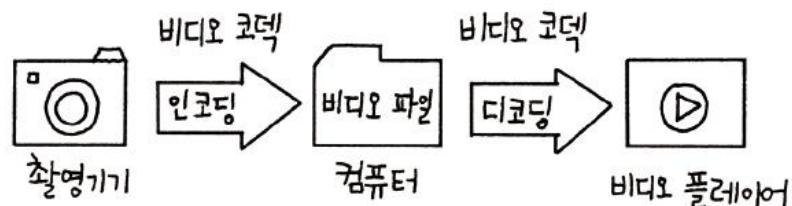
종류	확장자	설명
비디오	mp4	고화질 영상을 지원해 많은 사이트에서 사용되고 있습니다. 라이선스가 있으므로 웹 용을 제외하면 유료로 사용해야 합니다.
	webm	화질이 우수하고 무료로 제공되어 최근 많이 사용되고 있습니다.
	ogv	화질은 다른 비디오 유형보다 떨어지지만 무료라는 장점 때문에 webm 형식이 등장하기 전에 많이 사용되었습니다.
오디오	mp3	대부분의 음원에서 사용되지만 라이선스 때문에 유료로 사용해야 합니다.
	ogg	무료이므로 게임 등에서 많이 사용합니다.



# 웹과 멀티미디어

## HTML5와 비디오 코덱

- 인코딩(encoding) : 원본 비디오를 컴퓨터에서 사용할 수 있는 비디오 파일로 변환
- 디코딩(decoding) : 컴퓨터 비디오 파일에 있는 비디오 정보를 가져와 플레이어에 보여주는 과정
- 비디오 코덱 : 인코딩과 디코딩 수행. HTML5에서는 브라우저에서 직접 재생할 수 있는 비디오 코덱만 허용.



### ① H.264/AVC

- 고화질의 영상. mp4 파일, mov 파일 등에서 사용
- 유료 코덱이지만 온라인에서 사용할 경우 무료로 사용 가능.
- 대부분 모던 브라우저에서 지원

### ② 오그 테오라(Ogg Theora)

- 공개 코덱. ogv 파일 형식에서 사용
- 파이어폭스와 오페라, 크롬에서 지원.

### ③ v8, v9

- 오픈 소스로 공개한 코덱. webm 파일에서 사용
- 화질이 좋고 무료로 제공
- 파이어폭스와 오페라, 크롬 등에서 지원.

- 대부분의 브라우저에 H.264 코덱을 지원하므로 mp4 파일을 기본적으로 사용
- 무료이면서 최신 코덱인 v9 코덱을 이용한 webm 파일도 함께 사용
- 이전 모던 브라우저 중에는 ogv 파일만 지원하는 경우도 있으므로 ogv 파일도 사용

# 웹과 멀티미디어

## HTML5와 오디오 코덱

### ① MPEG-1 AUDIO Layer3 (MP3 코덱)

- 가장 많이 사용하는 오디오 코덱
- mp3 파일에서 사용
- 특허권이 등록되어 있어 유료

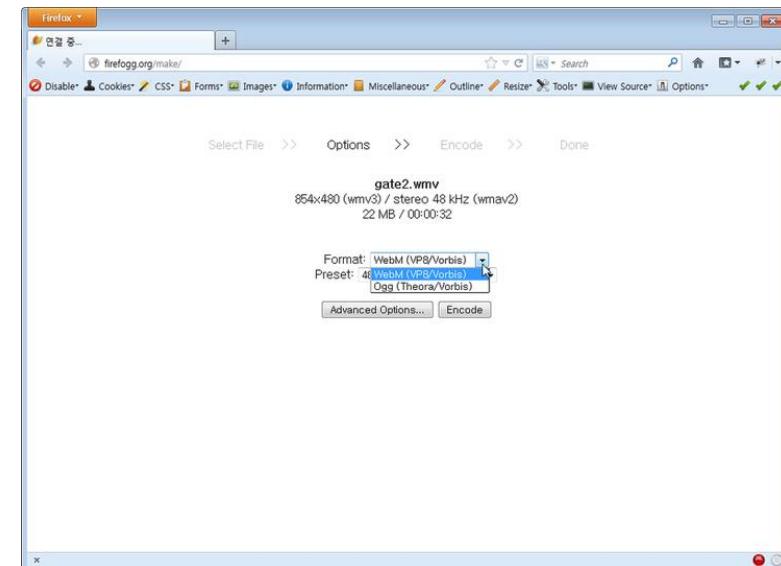
### ② 오그 보비스(Ogg Vorbis)

- 공개 코덱. ogg 파일 형식에서 사용
  - 재생 플레이어가 적고 인코딩 시간이 더 걸린다는 단점.
- but, 무료라서 PC 게임 등에 많이 사용됨

# 웹과 멀티미디어

## HTML5 비디오 변환

- 스마트폰 등에서 찍은 동영상을 webm과 mp4, ogv 파일로 변환.
- 인코딩 프로그램(인코더)을 이용해 비디오 파일 형식 변환  
ex) 카카오 인코더, 다음 팟인코더 등
- 파이어폭스 브라우저의 확장 프로그램 'firefogg'를 이용해 webm 파일과 ogv 파일로 변환



# 오디오 & 비디오 재생하기

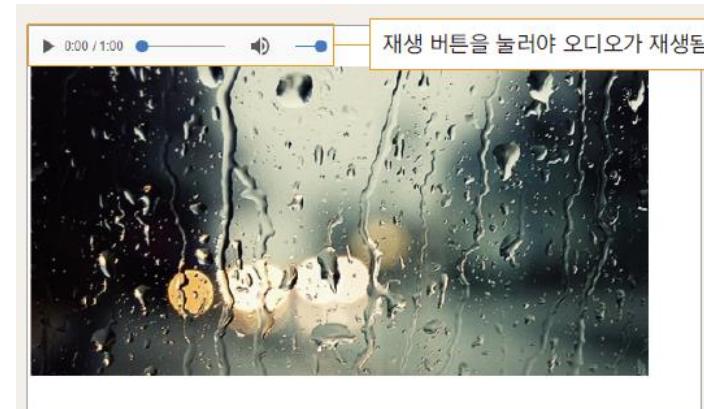
## <audio> 태그

- 배경 음악이나 효과음 등 오디오 재생
- 대부분 브라우저에서 mp3 지원하므로 mp3 파일만 사용

기본형    <audio src="오디오 파일 경로" [속성] [속성="속성 값"]></audio>

속성	설명
autoplay	오디오를 자동 재생합니다.
controls	웹 화면에 컨트롤 막대를 표시합니다. 컨트롤 막대에는 재생/멈춤, 진행 바, 볼륨 등이 표시됩니다.
loop	오디오를 반복 재생합니다.
muted	오디오를 재생해 진행하지만 소리는 끕니다.
preload	재생 버튼을 눌러 재생하기 전에 오디오 파일을 다운로드해 준비해 둡니다.

<audio src="media/bgsound.mp3" controls ></audio>



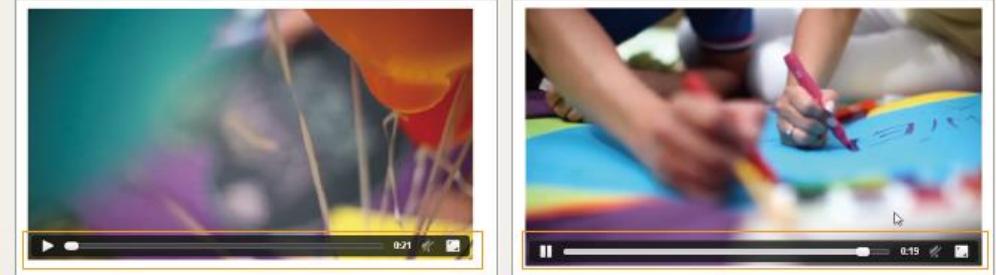
# 오디오 & 비디오 재생하기

## <video> 태그

웹 문서에 비디오 파일 삽입

기본형    <video src="비디오 파일 경로" [속성] [속성="속성 값"]></video>

<video src="media/Painting.mp4" controls></video>



<video src="media/Painting.mp4"></video>



크롬 바로가기 메뉴에서 컨트롤 표시

파이어폭스 바로가기 메뉴에서 메뉴 선택

# 오디오 & 비디오 재생하기

## <source> 태그

사용자들의 브라우저 환경을 고려해서 최신 브라우저와 이전 브라우저에서 모두 재생할 수 있도록 여러 코덱의 파일을 함께 지정

```
<source src="video.ogv" type="video/ogg; codecs='theora,vorbis'">
```

속성	설명
src	미디어 파일의 경로를 지정하는 필수 속성으로 파일 경로를 지정할 때는 경로에 공백이 있으면 안 됩니다.
type	웹 브라우저가 해당 미디어 파일을 재생할 수 있는지 여부를 확인할 수 있도록 미디어 파일의 유형을 알려 줍니다.
codecs	비디오 코덱을 지정합니다.

```
<video controls>
  <source src="media/Painting.ogv" type="video/ogg">
  <source src="media/Painting.mp4" type="video/mp4">
  <source src="media/Painting.webm" type="video/webm">
</video>
```

## HTML5를 지원하지 않는 브라우저에서는?

- HTML5 지원 브라우저가 필요하다는 대체 텍스트 표시

```
<video controls>
```

```
  <source src="media/Painting.mp4" type="video/mp4">
  <source src="media/Painting.webm" type="video/webm">
  <source src="media/Painting.ogv" type="video/ogg">
```

이 영상을 보기 위해서는 HTML5를 지원하는 브라우저가 필요합니다.

```
</video>
```

- 플래시 무비로 변환한 후 <embed> 태그나 <object> 태그로 삽입

```
<video controls>
```

```
  <source src="media/Painting.mp4" type="video/mp4">
  <source src="media/Painting.webm" type="video/webm">
  <source src="media/Painting.ogv" type="video/ogg">
  <object data="media/Painting.swf" type="application/x-
shockwave-flash"></object>
```

```
</video>
```

# 오디오 & 비디오 재생하기

## <audio>, <video> 태그의 속성

- **width, height** : 비디오 크기 조절
- **controls** : 미디어 파일에 컨트롤 막대 표시. 표시되는 컨트롤 막대는 웹 브라우저마다 다르며, 사용할 수 있는 명령도 조금씩 다르다.  
`<video src="media/Painting.mp4" controls></video>`
- **preload** : 재생하기 전에 비디오 파일을 모두 다운로드 할 것인지 또는 일부 정보만 다운로드 할 것인지 여부를 지정한다.
  - none – 재생 버튼을 눌러야 다운로드하기 시작 `<video src="media/Painting.mp4" controls preload="none"></video>`
  - metadata – 미디어 파일 전체를 다운로드하지 않고 메타 정보만 다운로드 `<video src="media/Painting.mp4" controls preload="metadata"></video>`
  - auto – 웹 문서를 로드할 때 미디어 파일도 모두 다운로드. 기본값 `<video src="media/Painting.mp4" controls preload="auto"></video>`
- **muted** : 비디오를 재생할 때 소리는 끄고 화면만 재생. 비디오를 문서 배경으로 사용하거나 소리가 필요하지 않을 때 사용
- **autoplay** : 미디어 파일을 다운로드 하자 마자 자동으로 재생. 모바일 기기에서는 자동 재생되지 않는다.
- **loop** : 미디어 파일 반복 재생. 속성 값 없이 loop 라고 하면 된다.
- **poster** : 포스터 이미지(비디오를 재생할 수 없을 경우 비디오 화면 자리에 대신 표시하는 이미지) 지정  
`<video src="media/Painting.mp4" controls poster="fireworks.jpg"></video>`

# 오디오 & 비디오 재생하기

## <track> 태그

- 비디오에 외부 자막 파일을 연결하는 태그
- 청각 장애인 뿐만 아니라 주변 소음이나 소리를 들을 수 없는 상황에서 비디오 내용을 이해하는데 도움이 됨

기본형    <track kind="자막 종류" src="경로" srclang="언어" label="제목" default>

### ① kind 속성 - 자막 종류 지정

속성 값	설명
subtitles	자막입니다. 소리를 볼 수 있지만 이해할 수 없는 경우, 다른 언어로 번역한 자막일 경우에 사용하며 비디오 화면에 표시됩니다.★
captions	캡션입니다. 청각장애인용 자막이거나 소리를 들을 수 없거나 볼 수 없는 경우에 사용합니다. 비디오 화면에 표시됩니다.
descriptions	비디오 콘텐츠에 대한 설명입니다. 비디오 화면에는 표시되지 않습니다.
chapters	비디오 탐색을 위한 장 제목입니다. 비디오 화면에 표시되지 않습니다.
metadata	비디오 콘텐츠 정보입니다. 비디오 화면에 표시되지 않습니다.

### ② src 속성 - 자막 텍스트의 파일 경로 지정    ③ srclang 속성 - 사용 언어 지정

### ④ label 속성 - 자막이 여러 개일 경우, 자막 식별 제목 표시

### ⑤ default 속성 - 자막 파일이 여러 개일 경우, 기본으로 사용할 자막을 default로 지정

예)

<track kind="subtitles" src="wildlife.vtt" srclang="ko" label="korean" default>

# 오디오 & 비디오 재생하기

## WebVTT 자막 파일

- 웹 비디오 텍스트 트랙(**Web Video Text Track**)
  - 모든 브라우저에서 공식적으로 지원하는 자막 파일 형식
  - 자막 내용과 시간 정보를 함께 담고 있음.
  - 비디오에서 자막을 확인하려면 서버에 올린 후 확인

기본형      시작 시간 --> 종료 시간  
                자막 내용

                시작 시간 --> 종료 시간  
                자막 내용

                ...

WEBVTT

예) 00:00.000 --> 00:02.172  
예술이란

00:02.172 --> 00:05.719  
자연이 인간에 투영된 것입니다

00:05.719 --> 00:07.716  
중요한 것은

00:07.716 --> 00:11.061  
깨끗하게 투영될 수 있도록

00:11.061 --> 00:20.879  
늘 깨끗하게 거울을 닦는 일입니다

```
<video autoplay>
  <source src="media/Painting.ogv" type="video/ogg">
  <source src="media/Painting.mp4" type="video/mp4">
  <source src="media/Painting.webm" type="video/webm">
  <track src="media/Painting.vtt" srclang="ko" label="Korean"
default>
</video>
```



크롬 브라우저의 자막



MS 엣지 브라우저의 자막

# 07. CSS 기초

스타일과 스타일 시트

주요 선택자

캐스케이딩 스타일 시트(CSS)

CSS3와 CSS 모듈



# 스타일과 스타일 시트

## 스타일과 스타일 시트

- 스타일(style) : HTML 문서에서 자주 사용하는 글꼴이나 색상, 정렬, 각 요소들의 배치 방법 등 문서의 겉모습을 결정짓는 내용들
- 스타일 시트(style sheet) : 스타일을 관리하기 쉽도록 한 군데 모아놓은 것

## 왜 스타일을 사용할까?

- 웹 문서의 내용과 상관없이 디자인만 바꿀 수 있다.
  - 내용과 디자인의 분리 – 웹 표준의 시작
  - 디자인에 영향 없이 내용 수정하거나 내용은 건드리지 않고 디자인만 바꾸는게 가능
  - [CSS Zen Garden](#) 사이트 참고
- 다양한 기기에 맞춰 탄력적으로 바꿔는 문서를 만들 수 있다.
  - 내용은 그대로 두고, 프린터나 스마트폰 브라우저 등 다양한 기기에 맞는 레이아웃을 만들 수 있다.

# 스타일과 스타일 시트

## 스타일 형식



- 선택자 : { 와 } 사이에 정의한 스타일 규칙이 적용될 대상.
- 속성과 속성 값 : '속성 : 속성 값'과 같은 형식으로 함께 표시하며, 속성/속성 값 쌍이 여럿일 경우에 세미콜론(:)으로 구분

```

p { color: blue; font-size: 16px; } /* 텍스트 단락의 글자 색 파란색, 글자 크기 16px */

h2 {font-size:20px; color:orange; } /* 2단계 제목의 글자 크기 20px, 글자 색상 주황색 */
p {color:blue; } /* 텍스트 단락의 글자 색 파란색 */
  
```

## 스타일을 표기하는 방법

```
h2{font-size:20px; color:orange;}
```

```

h2{
  font-size:20px;
  color:orange;
}
  
```

```

h2
{
  font-size:20px;
  color:orange;
}
  
```

모두 가능

## 스타일 주석

- /\*와 \*/ 사이에 주석 내용 입력
- 한 줄 또는 여러 줄을 입력 가능

# 스타일과 스타일 시트

## 내부 스타일 시트

- 웹 문서 안에서 사용할 스타일을 문서 안에 정리한 것
- 모든 스타일 정보는 <head> 태그와 </head> 태그 안에서 정의
- <style> 태그와 </style> 태그 사이에 작성



**세계 3대 미항**

- 시드니(Sydney), 호주
- 리우데자네이루(Rio de Janeiro), 브라질
- 나폴리(Naples), 이탈리아

```

<head>
  <meta charset="utf-8">
  <title>내부 스타일시트</title>
  <style>
    ul {
      color:blue; /* 목록의 글자색 */
      list-style-type:square; /* 불릿 형태 : 사각형 */
    }
  </style>
</head>
<body>
  
  <h1>세계 3대 미항</h1>
  <ul>
    <li>시드니(Sydney), 호주</li>
    <li>리우데자네이루(Rio de Janeiro), 브라질</li>
    <li>나폴리(Naples), 이탈리아</li>
  </ul>
</body>

```

# 스타일과 스타일 시트

## 외부 스타일 시트

- 여러 웹 문서에서 사용할 스타일을 별도 파일로 저장해 놓고 필요할 때마다 파일에서 가져와 사용
- <style> 태그 없이 <link> 태그만 사용해 미리 만들어 놓은 외부 스타일 시트 파일 연결

```

ul {
  color:blue;
  list-style-type:square;
}

```

세계 3대 미항

- 시드니(Sydney), 호주
- 리우데자네이루(Rio de Janeiro), 브라질
- 나폴리(Naples), 이탈리아

```

<head>
  <meta charset="utf-8">
  <title>외부 스타일시트</title>
  <link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
  
  <h1>세계 3대 미항</h1>
  <ul>
    <li>시드니(Sydney), 호주</li>
    <li>리우데자네이루(Rio de Janeiro), 브라질</li>
    <li>나폴리(Naples), 이탈리아</li>
  </ul>
</body>

```

# 스타일과 스타일 시트

## 인라인 스타일

- 스타일 시트를 사용하지 않고 스타일을 적용할 대상에 직접 표시
- 스타일을 적용하고 싶은 태그에 style 속성을 사용해 **style="속성: 속성 값;"** 형태로 스타일 적용

### 블루베리와 항산화 효능

블루베리는 항산화제인 안토시아닌과 폴리페놀을 다량 포함하고 있습니다.

매사추세츠 보스톤에 있는 USDA 노화에 관한 인류 영양 연구센터 (the USDA Human Nutrition Research Center on Aging) 의 자료에 의하면 블루베리는 과일 중에서 가장 항산화 작용이 뛰어난 과일이라고 합니다.

```
<body>
  <h1>블루베리와 항산화 효능</h1>
  <p style="color:blue;"> 블루베리는 항산화제인 안토시아닌과 폴리페놀을 다량 포함하고 있습니다.</p>
  <p> 매사추세츠 보스톤에 있는 USDA 노화에 관한 인류 영양 연구센터 (the USDA Human Nutrition Research Center on Aging) 의 자료에 의하면 블루베리는 과일 중에서 가장 항산화 작용이 뛰어난 과일이라고 합니다. </p>
</body>
```

## 전체 선택자

- 페이지에 있는 모든 요소를 대상으로 스타일을 적용할 때 사용
- 다른 선택자와 함께 모든 하위 요소에 한꺼번에 스타일을 적용하려고 할 때 주로 사용

기본형 \* {속성:속성 값; 속성:속성 값; ...}

예) \* {  
 margin:0;  
 padding:0;  
}

## 태그 선택자

- 문서에서 특정 태그를 사용한 모든 요소에 스타일이 적용됨

기본형      태그 { 스타일 }

### 블루베리에 관한 연구

#### 블루베리와 항산화 효능

블루베리는 항산화제인 안토시아닌과 폴리페놀을 다량 포함하고 있습니다.

매사추세츠 보스턴에 있는 USDA 노화에 관한 인류 영양 연구센터(The USDA Human Nutrition Research Center on Aging)의 자료에 의하면 블루베리는 과일 중에서 가장 항산화 작용이 뛰어난 과일이라고 합니다.

#### 블루베리와 노화

USDA 인류 영양 연구 센터(The USDA Human Nutrition Research Center) 실험실에서 신경과학자들은 블루베리를 연구하면서 노화에 대한 인지능력의 손실을 예방해 끝나는 사실을 발견하였습니다.

```

<head>
  <title>블루베리(Blueberry)</title>
  <style>
    h2 {
      color:blue;
    }
    p{
      font-size:12px;
      margin-left:20px;
    }
  </style>
</head>
<body>
  <h1>블루베리에 관한 연구</h1>
  <h2>블루베리와 항산화 효능</h2>
  <p>블루베리는 항산화제인 안토시아닌과 폴리페놀을 다량 포함하고 있습니다.</p>
  <p>매사추세츠 보스턴에 있는 ..... 과일이라고 합니다. </p>
  <h2>블루베리와 노화</h2>
  <p> USDA 인류 영양 연구센터..... 사실을 발견하였습니다. </p>
</body>

```

# 주요 선택자

## 클래스(class) 선택자, id 선택자

- **공통점** : 요소의 특정 부분에만 스타일 적용
- **차이점**
  - 클래스 선택자 : 문서 안에서 여러 번 반복할 스타일이라면 클래스 선택자로 정의.  
마침표(.) 다음에 클래스 이름 지정
  - id 선택자 : 문서 안에서 한번만 사용한다면 id 선택자로 정의.  
파운드(#) 다음에 id 이름 지정

### 클래스 선택자

```
.bluetext {  
  color:blue; /* 글자색 */  
}
```

### id 선택자

```
#container {  
  background: #ff6a00; /* 배경색 */  
  width:400px; /* 너비 */  
  height:200px; /* 높이 */  
  margin:0 auto; /* 가로로 중앙에 배치 */  
}
```

# 주요 선택자

## 클래스(class) 선택자

### 블루베리에 관한 연구

#### 블루베리와 항산화 효능

블루베리는 항산화제인 안토시아닌과 폴리페놀을 다량 포함하고 있습니다.

매사추세츠 보스톤에 있는 USDA 노화에 관한 인류 영양 연구센터 (the USDA Human Nutrition Research Center on Aging) 의 자료에 의하면 블루베리는 과일 중에서 가장 항산화 작용이 뛰어난 과일이라고 합니다.

#### 블루베리와 노화

USDA 인류 영양 연구센터(the USDA Human Nutrition Research Center) 신경과학자들은 쥐들에게 블루베리를 먹일 때 노화에 의한 인지능적의 손실을 예방해 준다는 사실을 발견하였습니다.

```

<head>
  <title>블루베리(Blueberry)</title>
  <style>
    h2.bluetext {
      color:blue; /* 글자색 */
    }
    .redtext {
      color: red; /* 글자색 */
    }
    p{
      font-size:12px; /* 글자 크기 */
      margin-left:20px; /* 왼쪽 마진 여백 */
    }
  </style>
</head>
<body>
  <h1>블루베리에 관한 연구</h1>
  <h2 class="bluetext">블루베리와 항산화 효능</h2>
  <p class="bluetext">블루베리는 항산화제인 ..... 있습니다.</p>
  <p>매사추세츠 보스톤에 있는 <span class="redtext">USDA 노화에 관한  
인류 영양 연구센터 (the USDA Human Nutrition Research Center on  
Aging)</span> 의 자료에 의하면 ..... 뛰어난 과일이라고 합니다. </p>
  <h2>블루베리와 노화</h2>
  <p><span class="redtext"> USDA 인류 영양 연구센터(the USDA Human  
Nutrition Research Center) </span> 신경과학자들은 ..... 발견하였습니다.
  </p>
</body>
```

# 주요 선택자

## id 선택자



```
<head>
  <meta charset="utf-8">
  <title>id 선택자</title>
  <style>
    #container {
      background: #ff6a00; /* 배경색 */
      width:400px; /* 너비 */
      height:200px; /* 높이 */
      margin:0 auto; /* 가로로 중앙에 배치 */
    }
  </style>
</head>
<body>
  <div id="container"></div>
</body>
```

## 그룹 선택자

- 같은 스타일을 사용하는 선택자를 한꺼번에 정의
- 쉼표(,)로 구분해 여러 선택자를 나열

기본형    이름1, 이름2, ...

```
h1 {  
    text-align:center; /* 가운데 정렬 */  
}  
h2 {  
    text-align:center; /* 가운데 정렬 */  
}
```

→ 

```
h1, h2 {  
    text-align:center; /* 가운데 정렬 */  
}
```

# 캐스캐이딩 스타일 시트(CSS)

## 캐스캐이딩의 의미

- 캐스캐이딩(Cascading) : '위에서 아래로 흐른다'는 뜻
- 선택자에 여러 스타일이 적용될 때 스타일 충돌을 막기 위해 '위에서 아래로 흐르며 적용되는' 방법을 선택

## 캐스캐이딩의 원칙

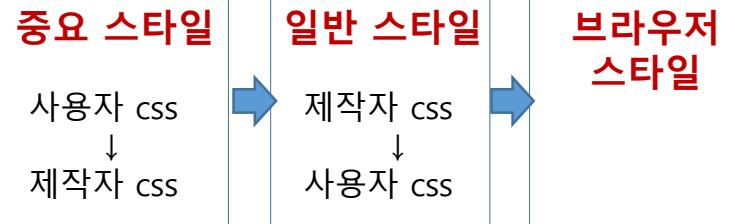
- ① 스타일 우선순위 – 스타일 규칙의 중요도, 적용 범위에 따라 우선순위가 결정되고 그 우선순위에 따라 위에서 아래로 스타일 적용
- ② 스타일 상속 – 태그들의 포함 관계에 따라 부모 요소의 스타일을 자식 요소로, 위에서 아래로 전달

※ 스타일 시트에서 '캐스캐이딩'은 가장 기본적인 개념이기 때문에 일반적으로 '스타일 시트'는 '캐스캐이딩 스타일 시트(CSS)'와 같은 의미로 사용됨

# 캐스케이딩 스타일 시트(CSS)

## 원칙 1: 스타일 우선 순위

### 1. 얼마나 중요한가에 따라



```

<head>
  <style>
    p {
      font-style:italic; /* 글자 스타일 */
      color:blue; /* 글자색 */
    }
    p {
      color:red !important; /* 글자색, 우선 적용 */
    }
  </style>
</head>
<body>
  <h1>세계 3대 미항</h1>
  
  <p>시드니(Sydney), 호주</p>
  <p>리우데자네이루(Rio de Janeiro), 브라질</p>
  <p>나폴리(Naples), 이탈리아</p>
</body>
  
```

### 세계 3대 미항



시드니(Sydney), 호주

리우데자네이루(Rio de Janeiro), 브라질

나폴리(Naples), 이탈리아

# 캐스케이딩 스타일 시트(CSS)

## 원칙 1: 스타일 우선 순위

### 2. 얼마나 한정지를 수 있는가에 따라



### 3. 소스 순서에 따라

- 중요도와 명시도가 같다면 소스 순서에 따라 결정
- 소스에서 나중에 온 스타일이 먼저 온 스타일을 덮어씀

```

<head>
  <style>
    h1 { color:blue; /* 글자색 */ }
    #habor{
      border:1px solid gray; /* 테두리 - 1픽셀짜리 회색 실선 */
      padding:10px; /* 패딩 여백 - 테두리와 내용 사이의 여백 */
      color:green; /* 글자색 */
    }
    .heading { color:red; /* 글자색 */ }
  </style>
</head>

<h1 class="heading" id="habor">세계 3대 미항</h1>

<p>시드니(Sydney), 호주</p>
<p>리우데자네이루(Rio de Janeiro), 브라질</p>
<p>나폴리(Naples), 이탈리아</p>
  
```

### 세계 3대 미항



시드니(Sydney), 호주

리우데자네이루(Rio de Janeiro), 브라질

나폴리(Naples), 이탈리아

# 캐스캐이딩 스타일 시트(CSS)

## 원칙 2: 스타일 상속

- 자식 요소에서 별도로 스타일을 지정하지 않으면 부모 요소에 있는 스타일 속성들이 자식 요소로 전달됨.
- 상속을 이용하면 스타일 시트를 효과적으로 만들 수 있다
- 주의할 것은 스타일의 모든 속성이 부모 요소에서 자식 요소로 상속되는 것은 아니라는 점  
(예, 글자 색은 상속되지만 배경 색은 상속되지 않음)
- 부모 요소로부터 스타일이 상속되는데 자식 요소에서 다른 스타일을 사용하고자 한다면, 즉, 스타일 충돌이 생긴다면 '중요도'나 '명시도', '소스순서' 등에 따라 우선 순위 결정.

```

<style>
  body {
    font-family:"돋움"; /* 글꼴 */
    color:darkgreen; /* 글자색 */
  }
</style>

<body>
  <h1>뉴욕 타임즈에서 발표한 세계 10대 슈퍼푸드
  </h1>
  <ul>
    <li>블루베리</li>
    <li>귀리</li>
    <li>토마토</li>
    <li>시금치</li>
    <li>적포도주</li>
    <li>건과류</li>
    <li>브로콜리</li>
    <li>연어</li>
    <li>마늘</li>
    <li>녹차</li>
  </ul>
</body>

```

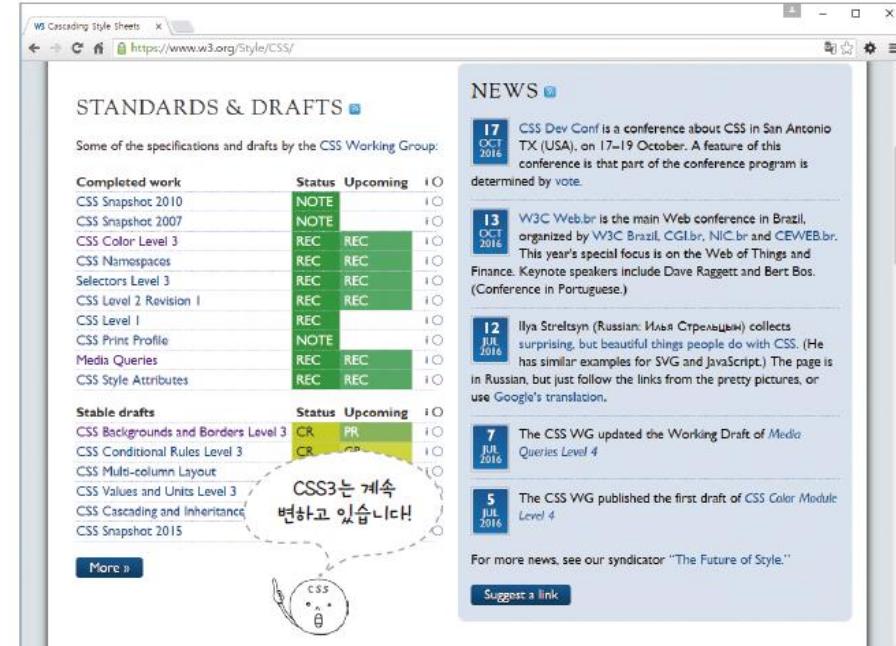
뉴욕 타임즈에서 발표한 세계 10대 슈퍼푸드

- 블루베리
- 귀리
- 토마토
- 시금치
- 적포도주
- 건과류
- 브로콜리
- 연어
- 마늘
- 녹차

## CSS3란

- CSS1 → CSS2 → **CSS3**
- CSS2를 기본으로 새로운 규약들을 추가한 것이 CSS3
- CSS2 규약 안에는 모든 스타일 규약이 담겨있어서 덩치가 크고 복잡해서 한꺼번에 업데이트하기 어려움  
→ CSS3부터는 배경이나 글꼴, 박스 모델 등 수십 개 기능을 주제 별로 규약을 따로 만듦 → “CSS 모듈”이라고 부름
- CSS 모듈별로 개발 진행 속도도 다르고 필요에 따라 계속 새로운 모듈이 생김  
→ CSS3는 한번에 표준 규약이 결정되지 않음.

[W3C CSS 사이트로 이동하기](#)



W3 CSS 사이트

## CSS3와 브라우저 접두사

- CSS3 모듈이 계속 개발되고 있는데 표준 규약이 아닌 속성들은 브라우저에 따라 다른 방식으로 지원됨  
→ 속성 이름 앞에 접두사(prefix)를 붙여 브라우저별로 구분해야

IE를 제외하고 크롬, 파이어폭스, 오페라, 사파리 등 웹 표준을 지원하는 브라우저.

- 표준 규약이 완성된 속성도 옛날 버전의 **모던 브라우저** 사용자를 고려하기 위해 브라우저 접두사를 붙여 사용하기도 함

접두사	설명
-webkit-	웹킷 방식 브라우저용(사파리, 크롬 등)
-moz-	게코 방식 브라우저용(모질라, 파이어폭스 등)
-o-	오페라 브라우저
-ms-	마이크로소프트 인터넷 익스플로러

```
<style>
.box:hover {
    -webkit-transform: rotate(15deg);
    -moz-transform: rotate(15deg);
    -o-transform: rotate(15deg);
    -ms-transform: rotate(15deg);
    transform: rotate(15deg);
}
</style>
```

```
<body>
<div class="box">Mouse Over</div>
</body>
```



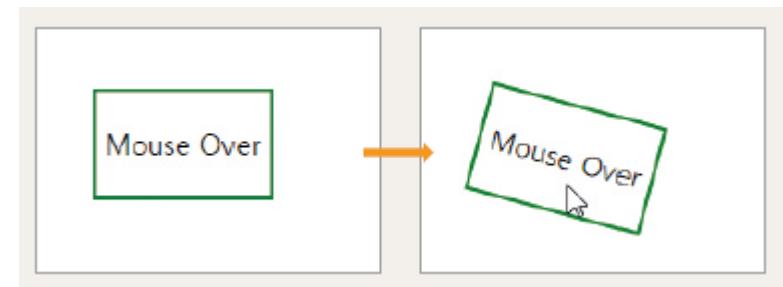
## 브라우저 접두사를 자동으로 붙여준다

-**prefix-free.js** : 브라우저 벤더 접두사를 자동으로 붙여줌.

1. <http://leaverou.github.io/prefixfree/>에서 파일 다운로드
2. prefixfree.min.js 파일을 원하는 곳으로 복사하거나 옮김
3. <script> 태그를 이용해 웹 문서에 삽입  
`<script src="prefixfree.min.js"></script>`
4. 이제부터는 브라우저 접두사 없이 CSS3 속성 사용

```
<head>
  <script src="prefixfree.min.js"></script>
  <style>
    .box:hover {
      transform: rotate(15deg);
    }
  </style>
</head>

<body>
  <div class="box">Mouse Over</div>
</body>
```



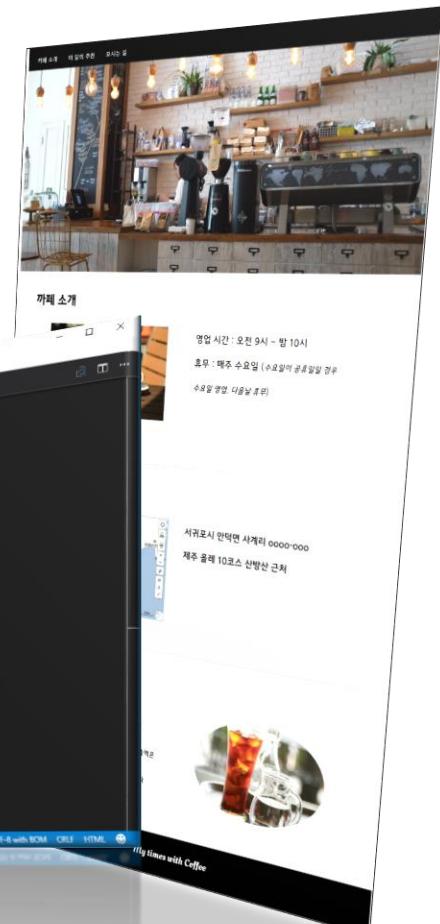
# 08. 텍스트 관련 스타일

## 글꼴 관련 스타일

# 텍스트 스타일

문단 스타일

## 목록과 링크 스타일



# 글꼴 관련 스타일

## font-family 속성

기본형 font-family:<글꼴 이름>[,<글꼴 이름>, <글꼴 이름>]

- 웹 문서에서 사용할 글꼴 지정
- <body> 태그를 비롯해 <p> 태그나 <hn> 태그처럼 텍스트를 사용하는 요소들에서 사용
- 웹 문서에서 글꼴을 지정할 때는 한 가지 글꼴만 지정하기도 하지만
- 지정한 글꼴이 없을 경우에 대비해 두 번째, 세 번째 글꼴까지 지정함.
- 둘 이상의 글꼴 이름을 지정할 때는 쉼표(,)로 글꼴 구분

`body { font-family:"맑은 고딕", 돋움, 굴림 }`

- font-family 속성은 상속되기 때문에 <body> 태그 스타일에서 한 번 정의하면 문서 전체에 적용되고 문서 안의 모든 자식 요소에 계속 같은 글꼴이 사용됨.
- 부모 요소와 다른 글꼴을 사용하고 싶다면 태그 스타일이나 클래스 스타일을 이용해 해당 요소에서 다른 글꼴을 정의한다.

웹 문서 전체에 "맑은 고딕"이라는 글꼴을 적용하는데 만일 "맑은 고딕" 글꼴이 없다면 "돋움" 글꼴로 적용하고 그 글꼴도 없다면 "굴림" 글꼴로 적용하라는 뜻

## @font-face 속성

- 웹 폰트(web-font) : 웹 문서 안에 글꼴 정보도 함께 저장했다가 사용자가 웹 문서에 접속하면 글꼴을 사용자 시스템으로 다운로드시켜 사용하는 글꼴.
- 사용자 시스템에 없는 글꼴이더라도 웹 제작자가 의도한 대로 텍스트를 표시할 수 있다.

### 구글 웹 폰트 사용하기

- <https://fonts.googleapis.com/earlyaccess>로 접속
- 한글 폰트 검색
- Link 항목에 있는 소스 복사 & 글꼴 이름 기억
- 웹 문서의 <style> 태그 안에 붙여 넣음
- font-family 속성에서 웹 폰트 글꼴 이름 사용

```

<style>
  @import
    url(http://fonts.googleapis.com/earlyaccess/nanumgothic.css);
    /* 구글 웹 폰트 */
  .ng-font {
    font-family:'Nanum Gothic', 돋움; /* 웹 폰트 지정 */
  }
  p {
    font-size:30px; /* 글자 크기 */
  }
</style>

<body>
  <p>브라우저 기본 글꼴 사용</p>
  <p class="ng-font">나눔고딕 웹 폰트 사용</p>
</body>

```

브라우저 기본 글꼴 사용

나눔고딕 웹 폰트 사용

## @font-face 속성

### 직접 웹 폰트 업로드해 사용하기

1. 웹 폰트 파일 준비
  - eot 파일, woff 파일
  - 기존 ttf 파일을 변환해서 사용할 수도 있음
2. 다운로드하기 전에 사용자 시스템에 있는지 확인
  - local(글꼴이름)
3. IE8 이하 버전을 위해 eot 파일 먼저 선언
4. woff 파일 선언
5. 용량이 큰 ttf 파일을 마지막에 선언

```

<style>
  @font-face {
    font-family: 'trana'; /* 글꼴 */
    src: local('trana'),
        url('trana.eot'),
        url('trana.woff') format('woff'),
        url('trana.ttf') format('truetype');
  }
  .w-font {
    font-family:'trana', sans-serif; /* 웹 폰트 지정 */
  }
  p {
    font-size:30px; /* 글자 크기 */
  }
</style>

<p>Using Default Fonts</p>
<p class= "w-font">Using Trana Fonts</p>

```

Using Default Fonts

USING TRANA FONTS

# 글꼴 관련 스타일

## font-size 속성

- 글자 크기를 조절하는 속성
- 사용할 수 있는 값: 절대 크기, 상대 크기, 숫자, 백분율
- 기본 값은 상대 크기인 medium
- font-size 속성은 상속된다.

기본형 font-size: <절대 크기> | <상대 크기> | <크기> | <백분율>

속성 값	설명
<절대 크기>	브라우저에서 지정한 글자 크기입니다. 사용할 수 있는 값은 xx-small   x-small   small   medium   large   x-large   xx-large입니다.
<상대 크기>	부모 요소의 글자 크기(font-size)를 기준으로 더 크게 표시하거나 더 작게 표시합니다. 사용할 수 있는 값은 larger   smaller입니다.
<크기>	브라우저와 상관없이 글자 크기를 직접 지정합니다.
<백분율>	부모 요소의 글자 크기를 기준으로 해당되는 %를 계산해 표시합니다(%와 함께 표기해야 합니다).

## <크기> 값에서 사용하는 단위

단위	설명
em	해당 글꼴의 대문자 M의 너비를 기준으로 크기를 조절합니다.
ex	x-height(엑스 하이트). 해당 글꼴의 소문자 x의 높이를 기준으로 크기를 조절합니다.
px	픽셀. 모니터에 따라 상대적 크기가 됩니다.
pt	포인트. 일반 문서에서 많이 사용하는 단위입니다.

### px 단위

- px 단위를 사용하면 폰트 크기가 고정됨
- 모바일 기기로 볼 때도 같은 크기로 화면에 표시되기 때문에 작은 화면 안에 작은 글씨로 표시됨

### em 단위

- 사용하는 글꼴의 대문자 M을 기준으로 한다.
- 대문자 M의 너비를 1em으로 놓고 상대적 값을 계산해 다른 요소들의 글자 크기를 조절함

# 글꼴 관련 스타일

## font-weight 속성

글자 굵기를 조절하는 속성

기본형 font-weight: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900

속성 값	설명
normal	일반적인 형태로 기본 값입니다.★
bold   lighter   bolder	굵게   원래 굵기보다 더 가늘게   원래 굵기보다 더 굵게 나타냅니다.
100~900 사이의 수치	400은 normal, 700은 bold에 해당하며 숫자 값을 조절해 좀 더 세밀하게 글꼴의 두께를 조절할 수 있습니다.

## font-variant 속성

대문자를 소문자 크기에 맞추어 작게 표시

기본형 font-variant: normal | small-caps

속성 값	설명
normal	일반적인 형태로 표시합니다.★
small-caps	작은 대문자로 표시합니다.

```
<style>
.accent {
  font-variant:small-caps; /* 작은 대문자 */
  font-weight:bold; /* 굵게 */
}
</style>

<h1>세계 3대 미항</h1>
<p><span class="accent">시드니(Sydney)</span>, 호주</p>
<p><span class="accent">리우데자네이루(Rio de Janeiro)</span>, 브라질</p>
<p><span class="accent">나폴리(Naples)</span>, 이탈리아</p>
```

### 세계 3대 미항

시드니(SYDNEY), 호주

리우데자네이루(RIO DE JANEIRO), 브라질

나폴리(NAPLES), 이탈리아

# 글꼴 관련 스타일

## font-style 속성

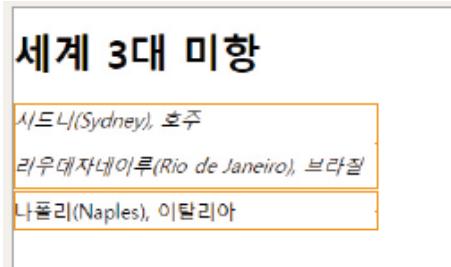
글자를 이탤릭체로 표시하는 속성

기본형 font-variant: normal | small-caps

속성 값	설명
normal	일반적인 형태로 표시합니다.*
small-caps	작은 대문자로 표시합니다.

```
<style>
  p {font-style:italic;}
  p#txt {font-style:normal;}
</style>
```

```
<h1>세계 3대 미항</h1>
<p>시드니(Sydney), 호주</p>
<p>리우데자네이루(Rio de Janeiro), 브라질</p>
<p id="txt">나폴리(Naples), 이탈리아</p>
```



## font 속성

글꼴 관련 속성들을 한꺼번에 묶어 표기

기본형 font: <font-style><font-variant><font-weight><font-size/line-height><font-family> | caption | icon | menu | message-box | small-caption | status-bar

속성 값	설명
font-*	font-로 시작하는 글꼴 관련 속성을 한꺼번에 나열합니다.
caption	캡션에 어울리는 글꼴 스타일로 표시합니다.
icon	아이콘에 어울리는 글꼴 스타일로 표시합니다.
menu	드롭다운 메뉴에 어울리는 글꼴 스타일로 표시합니다.
message-box	대화상자에 어울리는 글꼴 스타일로 표시합니다.
small-caption	작은 캡션에 어울리는 글꼴 스타일로 표시합니다.
status-bar	상태 표시줄에 어울리는 글꼴 스타일로 표시합니다.

line-height는 줄 간격 조절 속성인데 font 속성은 아니지만 글자 크기와 줄 간격이 밀접한 관련이 있기 때문에 font-size/line-height처럼 하나의 속성처럼 사용하기도 함

## color 속성

- 글자 색 지정
- 16진수 값이나 rgb 값, hsl 값, 색상 이름 중에서 사용

기본형 color : <색상>

```
<style>
    h1 {color:rgb(0,200,0);} /* rgb 값 사용 - 녹색 계열 */
    h2 {color:blue;} /* 색상 이름 사용 - 파랑 */
    .accent {color:#ff0000;} /* 16진수 사용 - 빨강 */
</style>
```

```
<h1>세계 10대 슈퍼푸드</h1>

<h2>마늘(Garlic)</h2>
<p>일해백리(一害百利)는 ..... 준다는 것이다. </p>
<p>마늘 특유의 아린 맛은
    <span class="accent">알리신</span>이라는..... 준다.
</p>
```

**세계 10대 슈퍼푸드**

**마늘(Garlic)**

일해백리(一害百利)는 마늘의 별명이다. 한가지 해가 있고 백가지 이로움이 있다는 뜻이다. 그 한 가지 해란, 아린 맛으로 인해 위에 부담을 준다는 것이다.

마늘 특유의 아린 맛은 **알리신**이라는 성분 때문으로, 살균 및 항균 작용을 하며 마늘에 들어 있는 아연이 피로 회복에 도움을 준다.



## text-decoration 속성

- 텍스트에 밑줄을 긋거나 가로지르는 줄 표시
- 텍스트 링크의 밑줄을 없앨 때도 사용

기본형 text-decoration : none | underline | overline | line-through

속성 값	설명
none	밑줄을 표시하지 않습니다.*
underline	밑줄을 표시합니다.
overline	영역 위로 선을 그립니다.
line-through	영역을 가로지르는 선(취소 선)을 그립니다.

```
<style>
  a {text-decoration:none;} /* 밑줄 없앰 */
  .edited {text-decoration:line-through;} /* 취소선 */
</style>
```

```
<h2>토마토(Tomato)</h2>
<p>[<a href="https://www.fitbug.com/q/Superfoods-tomatoes" target="_blank">외부 링크</a>]</p>
<p>토마토는 비타민A, C가 풍부한
  <span class="edited">과일이다.</span>
  채소다. ..... 음식이기도 하다.
</p>
```

### 토마토(Tomato)

[외부 링크]

토마토는 비타민A, C가 풍부한 과일이다. 채소다. 칼륨과 같은 각종 미네랄은 혈액의 산성도를 낮추는 역할을 해주며 혈압을 내리고 혈관을 튼튼하게 해준다. 토마토에 들어있는 피코펜이라는 성분은 뛰어난 항암 작용을 보이며 잘 알려져 있듯이 블루베리와 함께 대표적인 항산화 음식이기도 하다.

## text-transform 속성

영문 텍스트의 대문자나 소문자를 바꾸는 속성

기본형 text-transform : none | capitalize | uppercase | lowercase | full-width;

속성 값	설명
none	변환하지 않습니다.*
capitalize	시작하는 첫 번째 글자를 대문자로 변환합니다.
uppercase	모든 글자를 대문자로 변환합니다.
lowercase	모든 글자를 소문자로 변환합니다.
full-width	가능한 모든 문자를 전각 문자로 변환합니다.

```
<style>
  .trans1 {text-transform:uppercase;} /* 대문자로 */
  .trans2 {text-transform:capitalize;} /* 첫글자만 대문자로 */
</style>
```

```
<h1>Have to study</h1>
<ul>
  <li class="trans1">html</li>
  <li class="trans1">css</li>
  <li class="trans2">javascript</li>
</ul>
```

Have to study

- HTML
- CSS
- Javascript

## text-shadow 속성

텍스트에 그림자 효과를 추가하는 속성

기본형 text-shadow : none | <가로 거리> <세로 거리> <번짐 정도> <색상>

속성 값	설명
<가로 거리>	텍스트부터 그림자까지의 가로 거리를 입력합니다. 양수 값은 글자 오른쪽, 음수 값은 글자 왼쪽에 그림자를 만듭니다. 필수 속성입니다.
<세로 거리>	텍스트부터 그림자까지의 세로 거리를 입력합니다. 양수 값은 글자 아래쪽, 음수 값은 글자 위쪽에 그림자를 만듭니다. 필수 속성입니다.
<번짐 정도>	그림자가 얼마나 번지는지를 나타냅니다. 양수 값을 사용하면 그림자가 모든 방향으로 퍼져 나가기 때문에 그림자가 크게 표시됩니다. 반대로 음수 값은 그림자가 모든 방향으로 축소되어 보입니다. 기본 값은 0입니다.
<색상>	그림자 색상을 지정합니다. 한 가지만 지정할 수도 있고 공백으로 구분해 여러 색상을 지정할 수도 있습니다. 기본 값은 현재 글자 색입니다.

```

<style>
h1 {
    font-size:100px; /* 글자 크기 */
    font-family:"Arial Rounded MT"; /* 글꼴 */
}
.shadow1{
    color:orange; /* 글자색 */
    text-shadow:1px 1px; /* 텍스트 그림자 */
}
.shadow2 {
    text-shadow: 5px 5px 3px #f00; /* 텍스트 그림자 */
}
.shadow3 {
    color:#fff; /* 글자색 */
    text-shadow: 7px -7px 5px #000; /* 텍스트 그림자 */
}
</style>

<h1 class="shadow1">HTML5</h1>
<h1 class="shadow2">HTML5</h1>
<h1 class="shadow3">HTML5</h1>

```

HTML5

HTML5

HTML5

## white-space 속성

공백 처리 방법 지정

기본형 white-space : normal | nowrap | pre | pre-line | pre-wrap

속성 값	설명
normal	기본 값으로 연속하는 공백을 하나로 처리합니다.*
nowrap	연속하는 공백을 하나로 처리하고 줄을 바꾸지 않습니다.
pre	연속하는 공백을 소스 그대로 표시하고 소스에서 줄을 바꾼 부분만 줄을 바꾸어 표시하며 영역 너비를 넘어가는 부분의 줄은 자동으로 바꾸지 않습니다.
pre-line	연속하는 공백은 하나로 취급하고 소스에서 줄을 바꾼 부분은 줄을 바꾸어 표시하며 영역 너비를 넘어가는 부분이 있다면 자동으로 줄을 바꿉니다.
pre-wrap	연속하는 공백을 소스 그대로 표시하고 소스에서 줄을 바꾼 부분은 줄을 바꾸어 표시하며 영역 너비를 넘어가는 부분이 있다면 자동으로 줄을 바꿉니다.

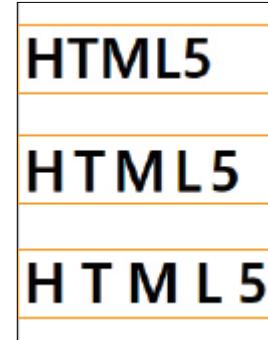
## letter-spacing, word-spacing 속성

글자간 간격, 단어간 간격

기본형 letter-spacing : normal | <크기>  
word-spacing : normal | <크기>

```
<style>
  h1 {font-size:40px;} /* 글자 크기 */
  .letter1 { letter-spacing: 0.2em; } /* 자간 */
  .letter2{ letter-spacing:0.5em; } /* 자간 */
</style>
```

```
<h1>HTML5</h1>
<h1 class="letter1">HTML5</h1>
<h1 class="letter2">HTML5</h1>
```



# 문단 스타일

## direction 속성

- 텍스트를 쓰는 방향 지정.
- 아랍어처럼 오른쪽에서 왼쪽으로 쓰는 언어일 경우 direction 속성으로 right to left 로 지정

기본형 direction : ltr | rtl

속성 값	설명
ltr	왼쪽에서 오른쪽으로(left-to-right) 텍스트를 표시합니다. 기본 값입니다.★
rtl	오른쪽에서 왼쪽으로(right-to-left) 텍스트를 표시합니다.

## text-align 속성

### 텍스트 정렬 방법 지정

기본형 text-align : start | end | left | right | center | justify | match-parent

속성 값	설명
start	현재 텍스트 줄의 시작 위치에 맞추어 문단을 정렬합니다. left-to-right 언어라면 왼쪽으로, right-to-left 언어라면 오른쪽으로 맞추어 정렬합니다.
end	현재 텍스트 줄의 끝 위치에 맞추어 문단을 정렬합니다. left-to-right 언어라면 오른쪽으로, right-to-left 언어라면 왼쪽으로 맞추어 정렬합니다.
left	왼쪽에 맞추어 문단을 정렬합니다.
right	오른쪽에 맞추어 문단을 정렬합니다.
center	가운데에 맞추어 문단을 정렬합니다.
justify	양쪽에 맞추어 문단을 정렬합니다.
match-parent	부모 요소부모 요소를 따라 문단을 정렬합니다. 다만 부모 요소의 속성 값이 start나 end일 경우, 부모 요소가 left-to-right인지 right-to-left인지에 따라 left나 right 값으로 계산해 적용합니다.

## text-align 속성

```
<style>
p {
    border: 1px solid #ccc; /* 테두리 - 1픽셀짜리 회색 실선 */
    padding: 10px; /* 테두리와 내용 사이의 패딩 여백 */
    margin: 10px; /* 단락 주변의 마진 여백 */
}
.align-left {text-align:left;} /* 왼쪽 정렬 */
.align-right {text-align:right;} /* 오른쪽 정렬 */
.align-center {text-align:center;} /* 가운데 정렬 */
.align-justify {text-align:justify;} /* 양쪽 정렬 */
</style>

<p class="align-left"> Integer elementum ..... </p>
<p class="align-right"> Integer elementum ..... </p>
<p class="align-center"> Integer elementum ..... </p>
<p class="align-justify"> Integer elementum ..... </p>
```

## text-justify 속성

text-align="justify"일 경우 양쪽 끝에 맞춰 정렬할 때 글자와 단어 사이의 간격을 조절하는 속성

기본형 text-justify : auto | none | inter-word | distribute

속성 값	설명	예시
auto	웹 브라우저에서 자동으로 지정합니다.	日本語 Latin ॥ ॥ ॥
none	정렬하지 않습니다.	
inter-word	단어 사이의 공백을 조절해 정렬합니다.	日本語 Latin ॥ ॥ ॥
distribute	인접한 글자 사이의 공백을 똑같이 맞추어서 정렬합니다.	日本語 Latin ॥ ॥ ॥

## text-indent 속성

문단의 첫 글자를 얼마나 들여 쓸지 지정

기본형 `text-indent : <크기> | <백분율>`

속성 값	설명
<크기>	단위와 함께 들여 쓸 크기를 지정합니다. 음수 값도 사용할 수 있습니다.
<백분율>	부모 요소의 너비를 기준으로 상대적 크기를 지정합니다.

```
<style>
  .indent1 {text-indent:15px;} /* 15px만큼 들여쓰기 */
  .indent2 {text-indent:5%;} /* 5%만큼 들여쓰기 */
</style>
```

`<p>`블루베리는 비타민A, .... `</p>`

`<p class="indent1">`블루베리는 비타민A, .... `</p>`

`<p class="indent2">`블루베리는 비타민A, .... `</p>`

블루베리는 비타민A, C, E가 풍부하고 안토니시아민, 페놀 등이 활성 산소를 없애 노화를 억제하고 블루베리의 안토니시아민은 눈의 피로와 시력 저하를 회복시키는 효능을 가지고 한다.

블루베리는 비타민A, C, E가 풍부하고 안토니시아민, 페놀 등이 활성 산소를 없애 노화를 억제하고 블루베리의 안토니시아민은 눈의 피로와 시력 저하를 회복시키는 효능을 가지고 한다.

블루베리는 비타민A, C, E가 풍부하고 안토니시아민, 페놀 등이 활성 산소를 없애 노화를 억제하고 블루베리의 안토니시아민은 눈의 피로와 시력 저하를 회복시키는 효능을 가지고 한다.

## line-height 속성

- 문단의 줄 간격 지정
- <숫자>와 <백분율>은 부모 요소를 기준으로 몇 배인지 지정
- 보통 글자 크기의 1.5~2배 정도면 적당

기본형 `line-height : normal | <숫자> | <크기> | <백분율> | inherit`

```
<style>
  .big-line { line-height:2;} /* 글자 크기 2배 */
  .small-line { line-height: 0.7;} /* 글자 크기 0.7배 */
</style>
```

`<p>`블루베리의 대표적인 기능은 .... `</p>`

`<p class="small-line">`블루베리의 대표적인 기능은 .... `</p>`

`<p class="bi">`블루베리의 대표적인 기능은 .... `</p>`

블루베리의 대표적인 기능은 항산화로 비타민A, C, E가 풍부하고 안토니시아민, 페놀 등이 활성 산소를 없애 노화를 억제한다. 이밖에 블루베리의 안토니시아민은 눈의 피로와 시력 저하를 회복시키는 효능을 가지고 있다.

블루베리의 대표적인 기능은 항산화로 비타민A, C, E가 풍부하고 안토니시아민, 페놀 등이 활성 산소를 없애 노화를 억제한다. 이밖에 블루베리의 안토니시아민은 눈의 피로와 시력 저하를 회복시키는 효능을 가지고 있다.

블루베리의 대표적인 기능은 항산화로 비타민A, C, E가 풍부하고 안토니시아민, 페놀 등이 활성 산소를 없애 노화를 억제한다. 이밖에 블루베리의 안토니시아민은 눈의 피로와 시력 저하를 회복시키는 효능을 가지고 있다.

## text-overflow 속성

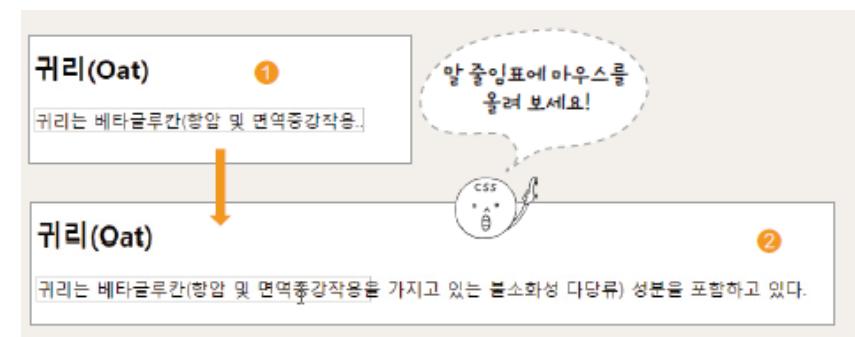
- 지정한 영역을 벗어나는 텍스트를 어떻게 할지 지정
- 해당 요소의 overflow 속성 값이 hidden일 때, 또는 overflow가 scroll, auto 이면서 white-space:nowrap일 때만 적용됨.

기본형 text-overflow: clip | ellipsis

속성 값	설명
clip	넘치는 텍스트를 잘라 버립니다.*
ellipsis	말 줄임표(...)로 잘린 텍스트가 있다고 표시합니다.

```
<style>
.content {
    border:1px solid #ccc; /* 테두리 */
    width:300px; /* 단락의 너비 */
    white-space:nowrap; /* 줄바꿈 없음 */
    overflow:hidden; /* 넘치는 부분 감춤 */
    text-overflow:ellipsis; /* 말줄임표 */
}
.content:hover {
    overflow:visible; /* 넘치는 부분 보여줌*/
}
</style>
```

<p class="content">귀리는 베타글루칸(항암 및 면역증강작용을 가지고 있는 불소화성 다당류) 성분을 포함하고 있다.</p>



## list-style-type 속성

순서 없는 목록의 불릿이나 순서 목록의 숫자를 바꾸는 속성

```
기본형 list-style-type : none | <순서 없는 목록의 불릿> | <순서 목록의 번호>
```

### 1. 순서 없는 목록의 불릿 바꾸기

속성 값	설명
disc (●)	채운 원 ★
circle (○)	빈 원
square (■)	채운 사각형
none	불릿 없애기

### 2. 순서 없는 목록의 불릿 없애기

```
ul { list-style-type:none; }
```

### 3. 순서 목록의 숫자 바꾸기

속성 값	설명	예시
decimal ★	1로 시작하는 십진수	1, 2, 3, ..., 10, 11
decimal-leading-zero	앞에 0이 붙는 십진수	01, 02, 03, ..., 10, 11
lower-roman	소문자 로마 숫자	i, ii, iii, iv
upper-roman	대문자 로마 숫자	I, II, III, IV
lower-alpha 또는 lower-latin	소문자 알파벳	a, b, c, d, e
upper-alpha 또는 upper-latin	대문자 알파벳	A, B, C, D, E
armenian	아르메니아 숫자	
georgian	조지 왕조시대의 숫자	an, ban, gan

# 목록과 링크 스타일

## list-style-type 속성

```

<style>
  .book1 {
    list-style-type: lower-alpha; /* 소문자 알파벳 */
  }
  .book2 {
    list-style-type: upper-roman; /* 대문자 로마 숫자 */
  }
</style>

<ol class="book1">
  <li>Do it! 시리즈</li>
  <li>된다 시리즈</li>
  <li>DCM 프로 사진가</li>
  <li>데이터과학 시리즈</li>
</ol>

<ol class="book2">
  <li>Do it! 시리즈</li>
  <li>된다 시리즈</li>
  <li>DCM 프로 사진가</li>
  <li>데이터과학 시리즈</li>
</ol>

```



## list-style-image 속성

순서 없는 목록의 불릿을 이미지로 바꾸는 속성

기본형    list-style-image : <이미지> | none  
<이미지> = url(이미지 파일 경로)

속성 값	설명
none	이미지를 사용하지 않고 list-style-type 속성에서 지정한 형태로 표시합니다.*
<이미지>	url("images.jpg") 처럼 url() 키워드 안에 이미지 파일 경로를 지정합니다.

```

<style>
  ul {
    list-style-image:url('images/dot.png');
    /* 불릿으로 사용할 이미지 */
  }
</style>

```



# 목록과 링크 스타일

## list-style-position 속성

불릿이나 번호를 들여쓰거나 내어쓸 수 있음

기본형 `list-style-position : inside | outside;`

속성 값	설명
inside	불릿이나 숫자를 안쪽으로 들여 씁니다.
outside	기본 값으로 불릿이나 숫자를 밖으로 내어 씁니다.★

```
<style>
  .inside {
    list-style-position:inside;
  }
</style>
```

```
<h3> list-style-position을 지정하지 않음</h3>
<ul> ..... </ul>
<h3> list-style-position :
inside</h3>
<ul class=inside> ..... </ul>
```



## list-style 속성

list-style-type, list-style-position, list-style-image 속성을 한꺼번에 표기

`ol {list-style-type:none;}`

속성을 대체해 사용

`ol {list-style:none;}`

`ul {`

`list-style-type:lower-alpha;  
list-style-position:inside;`

`}`

속성 값을 나열

`ul {`

`list-style: lower-alpha, inside ;`

`}`

# 09. 색상과 배경을 위한 스타일

## 웹에서 색상 표현하기

## 배경 색과 배경 이미지

## 그레이데이션 효과로 배경 꾸미기



```
<head>
    <style>
        #choice {
            float: right;
            margin-right: 5%;
        }
        #choice_text {
            float: left;
        }
        footer {
            clear: both;
        }
    </style>
</head>
<body>
    <div id="container">
        <header>
            <nav>
                <ul id="main-nav">
                    <li><a href="#intro">카페 소개</a></li>
                    <li><a href="#map">오늘은 끌어다녀요</a></li>
                    <li><a href="#choice">이 달의 주제</a></li>
                </ul>
            </nav>
        </header>
        <section id="intro">
            <div class="page-title">
                <h1>카페 소개</h1>
            </div>
            <div class="content">
                
            </div>
        </section>
    </div>
</body>

```

영업 시간: 오전 9시 ~ 밤 10시  
호모: 해주 수요일 (수요일이 공휴일인 경우  
수요일 영업, 다음날 휴무)  
서귀포시 애덕면 서계리 0000-0000  
제주 올레 103스 산방산 근처

# 웹에서 색상 표현하기

## 16진수 표기법

- #ffffff 처럼 #과 함께 6자리의 16진수로 표시.
- 앞에서부터 두자리씩 묶어 빨강, 초록, 파랑의 양.
- 하나도 섞이지 않았을 때는 00, 가득 섞였을 때는 ff로 표시.
- 000000(검은색) ~ ffffff(흰색)
- 두자리씩 중복될 경우 줄여 사용할 수 있음.

예) #ffff00 → #ff0, #cccccc → #ccc

## hsl/hsla 표기법

- color:hsl(240, 100%, 50%) 처럼 세 자리의 숫자로 표시.
- 앞의 숫자부터 색상(hue), 채도(saturation), 밝기(light)의 양.
- 투명도를 조절할 때는 hsla(240,100%,50%,0.3)처럼 마지막에 알파값 추가
- 알파값은 불투명도를 나타내는 값으로 0~1 값 중에서 사용  
(1은 불투명, 0은 완전 투명)

## rgb/rgba 표기법

- color:rgb(255,0,0)처럼 세 자리의 숫자로 표시.
- 앞의 숫자부터 빨강, 초록, 파랑의 양.
- 하나도 섞이지 않았을 때는 0, 가득 섞였을 때는 255
- 투명도를 조절할 때는 color:rgba(255,0,0,.3) 처럼 마지막에 알파값 추가
- 알파값은 불투명도를 나타내는 값으로 0~1 값 중에서 사용  
(1은 불투명, 0은 완전 투명)

## 색상 이름 표기법

- 잘 알려진 색상 이름으로 표시
- 기본 색상 16가지
- 모든 브라우저에서 표현할 수 있는 색상을 웹 안전 색상(web-safe color)라고 하는데 기본 16가지 색상을 포함해 모두 216가지입니다.

# 배경 색과 배경 이미지

## background-color 속성

기본형 background-color: <색상>

- 웹 문서의 요소에 배경색 지정
- 예) 웹 문서 전체의 지정하려면 body 요소에 사용.
- 16진수나 rgb 값, rgba 값 또는 색상 이름 사용

background-color:#00ff00;

① 16진수: 세밀히 색상 조절

background-color:rgb(0,255,0);

② rgba: 필요하면 투명도도 함께 조절 가능

background-color:green;

③ 색상 이름: 원색 사용

- 배경 색은 상속되지 않는다

```
<style>
  body {
    background-color:#0094ff; /* 문서 전체 배경색 */
  }
  div {
    background-color:white; /* div 영역 배경색 */
    width:90%; /* 너비 */
    padding:15px; /* 패딩 */
    border:1px solid black; /* 테두리 */
  }
  p {
    line-height:30px;
  }
</style>
```

### 초콜릿(Chocolate)

초콜릿은 카카오 콩을 재료로 가공한 식품이다. 숙성한 카카오 콩을 볶은 뒤 이를 갈아서 만든 카카오 맥스와 지망 성분만으로 만들어진 코코아 버터를 혼합하여 만드는데, 설탕 등의 다른 재료를 더 넣어 만들기도 한다. 카카오 맥스의 향광에 따라 다크 초콜릿, 밀크 초콜릿, 화이트 초콜릿으로 구분한다.

# 배경 색과 배경 이미지

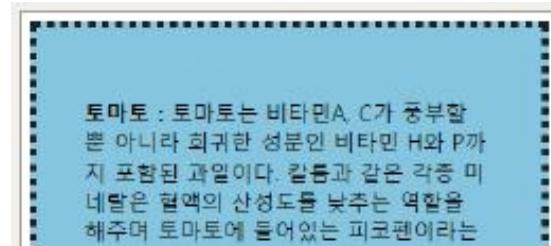
## background-clip 속성

- 배경을 어디까지 적용할지 지정
- 박스 모델 기준

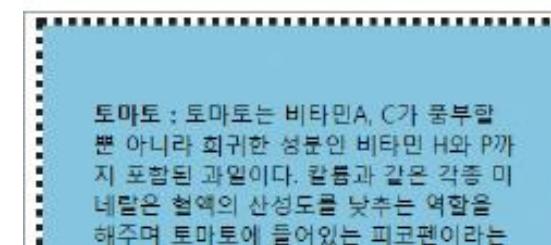
기본형 background-clip: border-box | padding-box | content-box

속성 값	설명
border-box	박스 모델의 가장 외곽인 테두리(border)까지 적용합니다.
padding-box	박스 모델에서 테두리를 뺀 패딩(padding) 범위까지 적용합니다.
content-box	박스 모델에서 내용 부분에만 적용합니다.

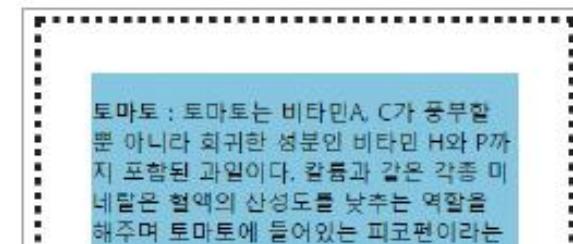
.bg1 { background-clip: border-box; }



.bg2 { background-clip: padding-box; }



.bg3 { background-clip: content-box; }



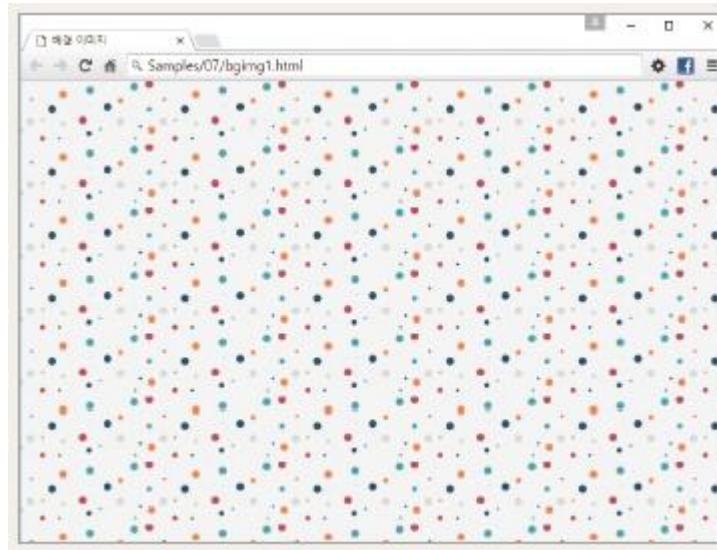
# 배경 색과 배경 이미지

## background-image 속성

- 배경 이미지 지정하기
- 배경 이미지 파일 경로 지정

**기본형** background-image: url(파일 경로)

```
<style>
  body { background-image: url('images/bg1.png'); }
</style>
```



문서 전체에 가로와 세로로 반복 표시된 배경 이미지

## background-repeat 속성

배경 이미지 반복 여부 및 반복 방향 지정

**기본형** background-repeat : repeat | repeat-x | repeat-y | no-repeat

속성 값	설명
repeat	브라우저 화면에 가득 찰 때까지 배경 이미지를 가로와 세로로 반복합니다. ★
repeat-x	브라우저 창 너비와 같아질 때까지 배경 이미지를 가로로 반복합니다.
repeat-y	브라우저 창 높이와 같아질 때까지 배경 이미지를 세로로 반복합니다.
no-repeat	배경 이미지를 한 번만 표시하고 반복하지 않습니다.

# 배경 색과 배경 이미지

## background-size 속성

배경 이미지 크기 조절

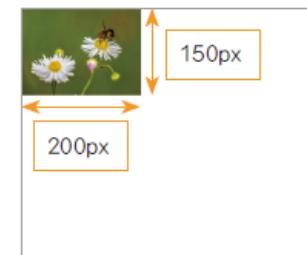
기본형 background-size: auto | contain | cover | <크기 값> | <백분율>

속성 값	설명
auto	원래 배경 이미지 크기만큼 표시됩니다.
contain	요소 안에 배경 이미지가 다 들어오도록 확대/축소
cover	배경 이미지로 요소를 모두 덮도록 확대/축소
<크기 값>	너비 값과 높이 값을 지정합니다. 너비 값만 지정할 경우, 원래 배경 이미지 크기를 기준으로 축소/확대 비율을 자동으로 계산해 높이 값을 지정합니다.
<백분율>	원래 배경 이미지 크기를 기준으로 확대하거나 축소합니다.

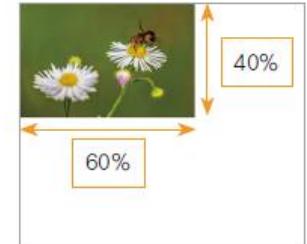
.bg1{ background-size:auto; }



.bg2 { background-size:200px 150px; }



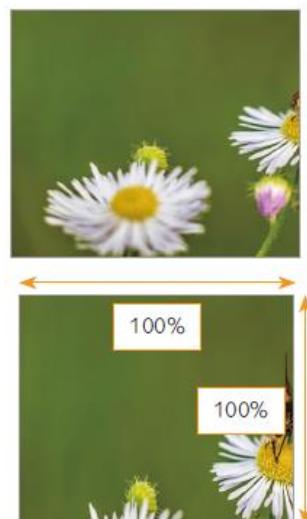
.bg3 { background-size:60% 40%; }



.bg4 { background-size:contain; }



.bg5 { background-size:cover; }



.bg6{ background-size:100% 100%; }

# 배경 색과 배경 이미지

## background-position 속성

배경 이미지를 반복하지 않을 경우, 배경 이미지를 표시할 위치 지정

**기본형** background-position: <수평 위치> <수직 위치>;  
 수평 위치 : left | center | right | <백분율> | <길이 값>  
 수직 위치 : top | center | bottom | <백분율> | <길이 값>

① **백분율** : 배경 이미지의 가로 위치와 세로 위치를 %로 나타냄.

예) background-position: 0% 0% , background-position : 30% 60%

② **길이 길이** : 배경 이미지의 위치를 직접 길이로 지정

예) background-position:30px 20px;

③ **키워드** - top, left, center, right, top, middle, bottom

가로 배치는 **left와 center, right** 중에서 선택

세로 배치는 **top과 center, bottom** 중에서 선택

예) background-position:center bottom

```
<style>
  #bg1 {
    background-image:url('images/bg2.jpg');
    background-repeat:no-repeat;
    background-position:left top;
    padding-top:150px;
  }
  #bg2 {
    background-image:url('images/bg2.jpg');
    background-repeat:no-repeat;
    background-position:center;
  }
</style>
```

```
<div id="bg1"> ..... </div>
<div id="bg2"> ..... </div>
```



id="bg1"일 때

초콜릿(Chocolate)

초콜릿은 카카오 씨앗을 재료로 가공한 식품이다. 숙성한 카카오 씨앗을 볶은 뒤 이를 갈아서 만든 카카오 맥스와 지방 성분만으로 만들어진 코코아 버터를 혼합하여 만드는데, 설탕 등의 다른 재료를 더 넣어 만들기도 한다. 카카오 맥스의 함량에 따라 다크 초콜릿, 밀크 초콜릿, 화이트 초콜릿으로 구분한다.

id="bg2"일 때

# 배경 색과 배경 이미지

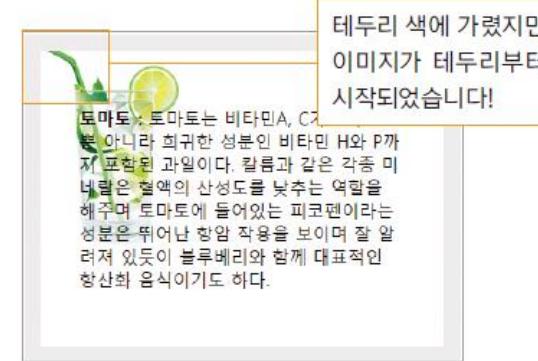
## background-origin 속성

배경 이미지를 배치하기 위한 기준 설정

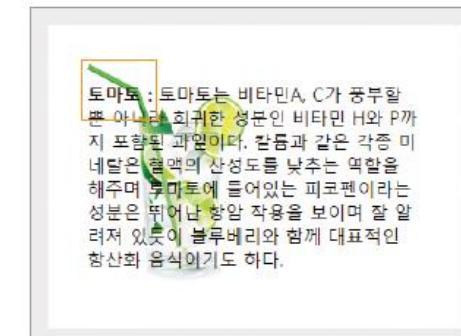
기본형    background-origin: border-box | padding-box | content-box

속성 값	설명
border-box	박스 모델의 가장 외곽인 테두리(border)가 기준이 됩니다.
padding-box	박스 모델에서 테두리를 뺀 패딩(padding)이 기준이 됩니다.
content-box	박스 모델에서 내용 부분이 기준이 됩니다.

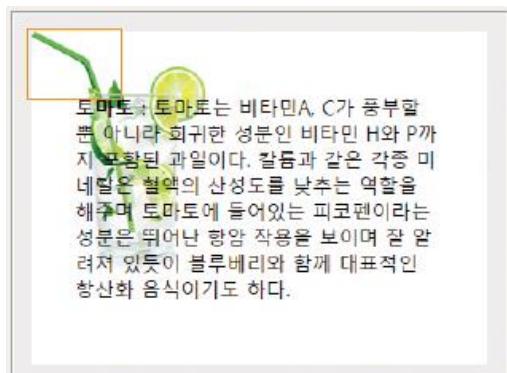
.bg2 { background-origin:border-box; }



.bg3 { background-origin:content-box; }



.bg1 { background-origin:padding-box; }



# 배경 색과 배경 이미지

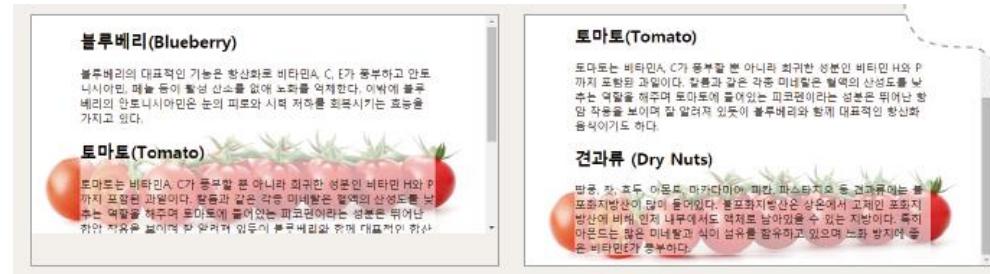
## background-attachment 속성

배경 이미지를 고정하는 속성

기본형 background-attachment: scroll | fixed;

속성	설명
scroll	화면 스크롤과 함께 배경 이미지도 스크롤됩니다. 기본 값입니다.★
fixed	화면이 스크롤되더라도 배경 이미지는 고정됩니다.

```
body {
  background-image:url('images/bg3.jpg');
  background-repeat:no-repeat;
  background-position:right bottom;
  background-attachment:fixed;
}
```

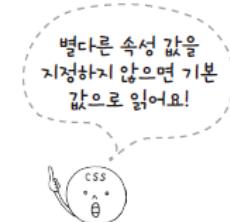


## background 속성

- 배경 관련 속성을 줄여서 표기
- 각 속성 값이 다르므로 표기 순서는 상관없음

background:url('images/bg3.jpg') no-repeat fixed right bottom;

속성	속성 값
background-image	url('images/bg3.jpg')
background-repeat	no-repeat
background-attachment	fixed
background-position	right bottom
background-clip	border-box
background-origin	padding-box
background-size	auto



# 그레이디언트 효과로 배경 꾸미기

## 그레이디언트와 브라우저 접두사

- 그레이디언트는 크기가 없는 배경 이미지이므로 background-image나 background 속성에서 사용
- 그레이디언트 속성은 표준화 됨
- 하지만 구형 모던 브라우저에서는 브라우저 접두사를 붙여야 동작함

접두사	브라우저 버전
-webkit-	사파리 5.1 ~ 6.0
-moz-	파이어폭스 3.6 ~ 15
-o-	오페라 11.1 ~ 12.0

```

<style>
  .grad {
    background : blue; /* 그레이디언트를 지원하지 않는 브라우저용 */
    background : -webkit-linear-gradient(left top, blue, white);
    background : -moz-linear-gradient(right bottom, blue, white);
    background : -o-linear-gradient(right bottom, blue, white);
    background : linear-gradient(to right bottom, blue, white); /* 표준 구문 */
  }
</style>

```

# 그레이디언 효과로 배경 꾸미기

## 선형 그레이디션

- 수직 방향이나 수평 방향으로, 혹은 대각선 방향으로 색상이 일정하게 변하는 것
- 선형 그레이디션을 지정할 때는 방향과 색상이 필요하다

기본형    `linear-gradient( <각도> to <방향>, color-stop, [color-stop,...]);`

- 위 구문이 표준 구문이지만 '위치'와 '각도'를 표시하는 방법이 중간에 몇 번 바뀌다 보니 브라우저별, 버전별 사용법이 조금씩 다름.

# 그라데이션 효과로 배경 꾸미기

## 선형 그라데이션

### 1) 방향

- [표준 구문] 끝 지점을 기준으로 'to' 키워드와 함께 사용

속성 값	설명
to top	아래에서 시작해 위로 그라데이션이 만들어집니다.
to left	오른쪽에서 시작해 왼쪽으로 그라데이션이 만들어집니다.
to right	왼쪽에서 시작해 오른쪽으로 그라데이션이 만들어집니다.
to bottom	위에서 시작해 아래로 그라데이션이 만들어집니다.

- [접두사 구문] 예전 모던 브라우저에서는 접두사를 사용해야 하는데, 접두사마다 방향 속성 값을 사용하는 기준이 다름.

접두사	브라우저 버전	'위치' 속성 값
-webkit-	사파리 5.1 ~ 6.0	그라데이션 시작 위치 기준
-moz-	파이어폭스 3.6 ~ 15	그라데이션 끝 위치 기준. 키워드 to 사용하지 않음
-o-	오페라 11.1 ~ 12.0	그라데이션 끝 위치 기준. 키워드 to 사용하지 않음

```

<style>
  .qgrad {
    background: blue; /* css3 미지원 브라우저 */
    background: -webkit-linear-gradient(left top, blue, white);
    /* 초기 모던 브라우저 */
    background: -moz-linear-gradient(right bottom, blue, white);
    /* 초기 모던 브라우저 */
    background: -o-linear-gradient(right bottom, blue, white);
    /* 초기 모던 브라우저 */
    background: linear-gradient(to right bottom, blue, white);
    /* 최신 모던 브라우저 */
  }
</style>

<div class="grad"></div>

```

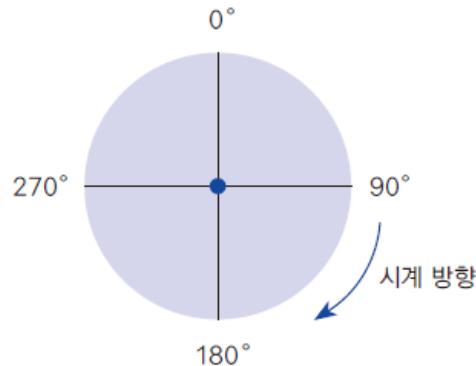


# 그레이디션 효과로 배경 꾸미기

## 선형 그레이디션

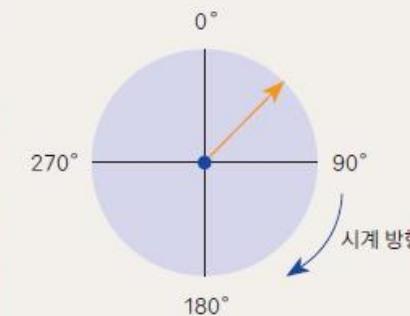
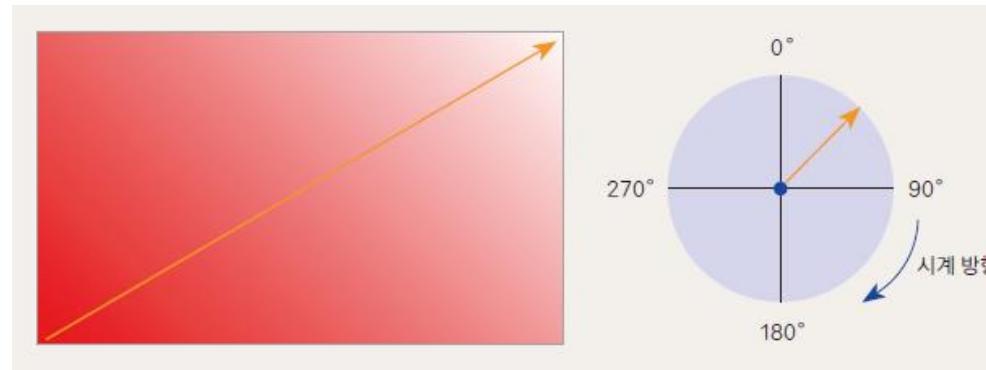
### 2) 각도

- 그레이디션이 끝나는 각도
- 단위는 deg



```
<style>
  .grad { /* 45도 (오른쪽 위) 방향으로, 빨강에서 흰색으로 */
    background: #ff0000; /* CSS3 미지원 브라우저 */
    background: -webkit-linear-gradient(45deg, #ff0000, #ffffff); /* 초기 모던 브라우저 */
    background: -moz-linear-gradient(45deg, #ff0000, #ffffff); /* 초기 모던 브라우저 */
    background: -o-linear-gradient(45deg, #ff0000, #ffffff); /* 초기 모던 브라우저 */
    background: linear-gradient(45deg, #ff0000, #ffffff); /* 최신 모던 브라우저 */
  }
</style>

<div class="grad"></div>
```



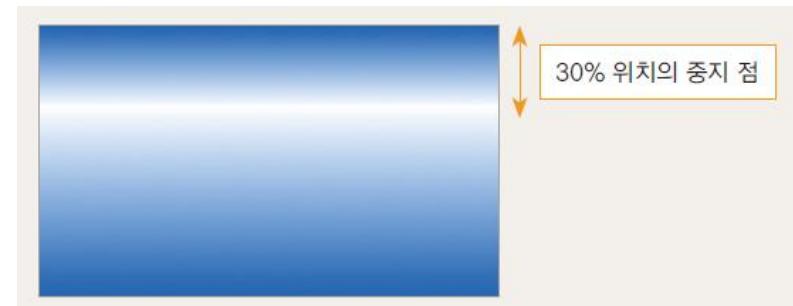
# 그레이디언 효과로 배경 꾸미기

## 선형 그레이디션

### 3) 색상 중지점

- 색상이 바뀌는 지점
- 색상만 지정할 수도 있고 색상과 함께 중지 점의 위치도 함께 지정할 수도 있음.

```
<style>
.grad {
  background: #06f; /* css3 미지원 브라우저 */
  background-image: -webkit-linear-gradient(top, #06f, white 30%, #06f); /* 초기 모던 브라우저 */
  background-image: -moz-linear-gradient(bottom, #06f, white 30%, #06f); /* 초기 모던 브라우저 */
  background: -o-linear-gradient(bottom, #06f, white 30%, #06f); /* 초기 모던 브라우저 */
  background: linear-gradient(to bottom, #06f, white 30%, #06f); /* 최신 모던 브라우저 */
}
</style>
```



# 그라데이션 효과로 배경 꾸미기

## 원형 그라데이션

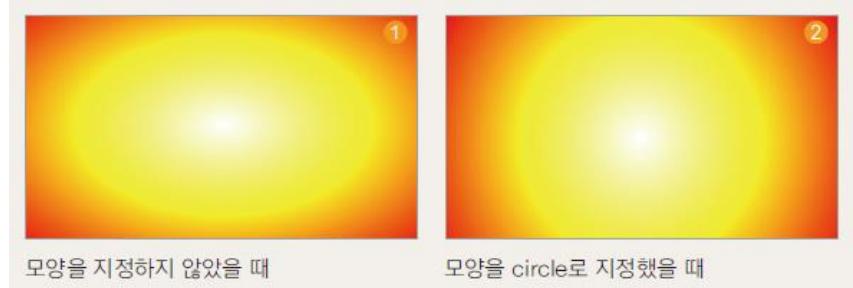
- 원이나 타원의 중심부터 동심원을 그리며 바깥 방향으로 색상이 바뀌는 그라데이션
- 색상이 바뀌기 시작하는 원의 중심과 크기를 지정하고 그라데이션의 모양을 지정해야 함.

기본형 radial-gradient( <최종 모양> <크기> at <위치>, color-stop, [color-stop...] )

```
<style>
.grad1{
  background:red;
  background:-webkit-radial-gradient(white, yellow, red);
  background:-moz-radial-gradient(white, yellow, red);
  background:-o-radial-gradient(white, yellow, red);
  background:radial-gradient(white, yellow, red);
}
.grad2{
  background:red;
  background:-webkit-radial-gradient(circle, white, yellow, red);
  background:-moz-radial-gradient(circle, white, yellow, red);
  background:-o-radial-gradient(circle, white, yellow, red);
  background:radial-gradient(circle, white, yellow, red);
}
</style>
```

### 1) 모양

- 원형 그라데이션에서 만들어지는 모양은 circle(원형)과 ellipse(타원형)
- 따로 지정하지 않으면 ellipse로 인식



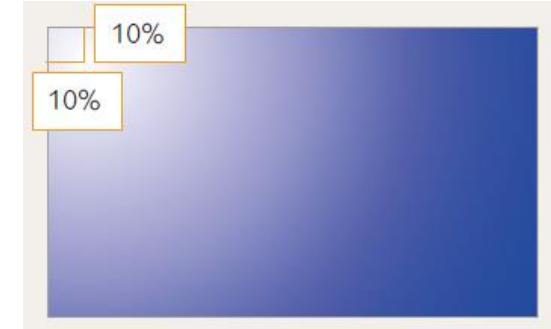
# 그레이디언트 효과로 배경 꾸미기

## 원형 그레이디션

### 2) 위치

- 그레이디션이 시작하는 원의 중심 지정
- [표준 구문] '모양'과 '크기' 속성 다음에 at 키워드와 함께 위치 값 지정
- [접두사 구문] at 키워드 없이 구문의 맨 앞에 위치 값 지정
- 사용할 수 있는 값 : 키워드(left, center, right 중 하나, top, center, bottom 중 하나)나 백분율

```
<style>
.grad {
  background: blue;
  background: -webkit-radial-gradient(10% 10%, circle, white,blue);
  background: -moz-radial-gradient(10% 10%, circle,white,blue);
  background: -o-radial-gradient(10% 10%, circle, white,blue);
  background: radial-gradient(circle at 10% 10%,white,blue);
}
</style>
```

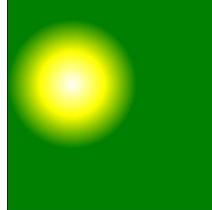
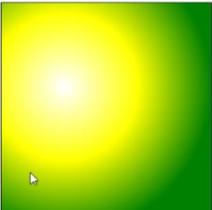
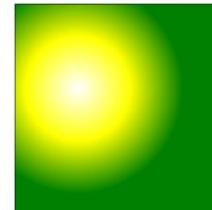
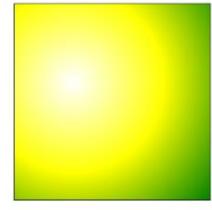


# 그러데이션 효과로 배경 꾸미기

## 원형 그레이데이션

### 3) 크기

그레이데이션 원의 크기 지정

속성 값	설명	예	속성 값	설명	예
closest-side	[원] 그레이데이션 가장자리가 그레이데이션 중심에서 가장 가까운 요소의 모서리와 만남 [타원] 그레이데이션 중심에서 가장 가까운 요소의 수평축이나 수직축과 만남		farthest-side	[원] 그레이데이션 가장자리가 그레이데이션 중심에서 가장 먼 모서리와 만남 [타원] 그레이데이션 가장자리가 그레이데이션 중심에서 가장 먼 모서리와 만남	
closest-corner	그레이데이션 가장자리가 그레이데이션 중심에서 가장 가까운 요소의 코너에 닿도록 함		farthest-corner	그레이데이션 가장자리가 그레이데이션 중심에서 가장 먼 코너에 닿도록 함. 기본값.	

# 그레이디언트 효과로 배경 꾸미기

## 원형 그레이디언트

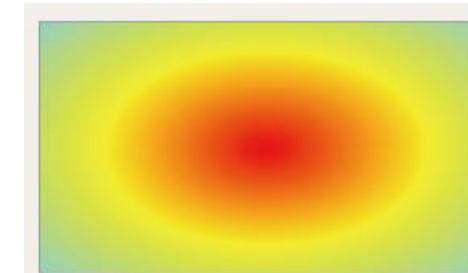
### 4) 색상 중지점

- 색상이 바뀌는 지점
- 색상만 지정할 수도 있고 색상과 함께 중지 점의 위치도 함께 지정할 수도 있음.

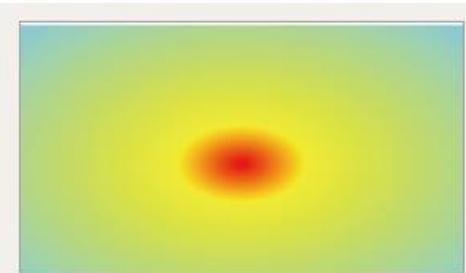
```

<style>
.grad1{
  background:skyblue;
  background:-webkit-radial-gradient(red, yellow, skyblue);
  background:-moz-radial-gradient(red, yellow, skyblue);
  background:-o-radial-gradient(red, yellow, skyblue);
  background:radial-gradient(red, yellow,skyblue);
}
.grad2{
  background:skyblue;
  background:-webkit-radial-gradient(red, yellow 20%, skyblue);
  background:-moz-radial-gradient(red, yellow 20%, skyblue);
  background:-o-radial-gradient(red, yellow 20%, skyblue);
  background:radial-gradient(red, yellow 20%, skyblue);
}
</style>

```



색상 중지 점의 위치를 지정하지 않았을 때



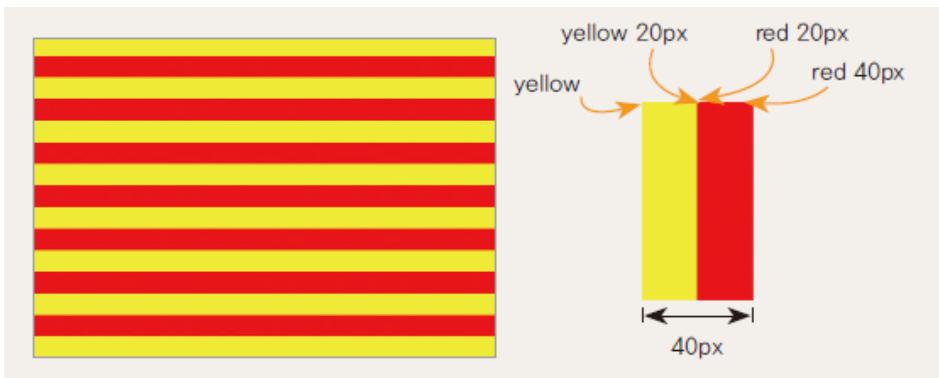
색상 중지 점의 위치를 지정했을 때

# 그레이디언트 효과로 배경 꾸미기

## 그레이디션 반복

단순히 그레이디션을 반복하는 것이 아니라 '패턴'을 만들어 반복시킴

```
<style>
.qrad2 {
  background: red; /* css3 미지원 브라우저 */
  background: -webkit-repeating-linear-gradient(yellow, yellow 20px, red 20px, red 40px); /* 초기 모던 브라우저 */
  background: -moz-repeating-linear-gradient(yellow, yellow 20px, red 20px, red 40px); /* 초기 모던 브라우저 */
  background: -o-repeating-linear-gradient(yellow, yellow 20px, red 20px, red 40px); /* 초기 모던 브라우저 */
  background: repeating-linear-gradient(yellow, yellow 20px, red 20px, red 40px); /* 최신 모던 브라우저 */
}
</style>
```



# 10. 레이아웃을 위한 스타일

CSS와 박스 모델

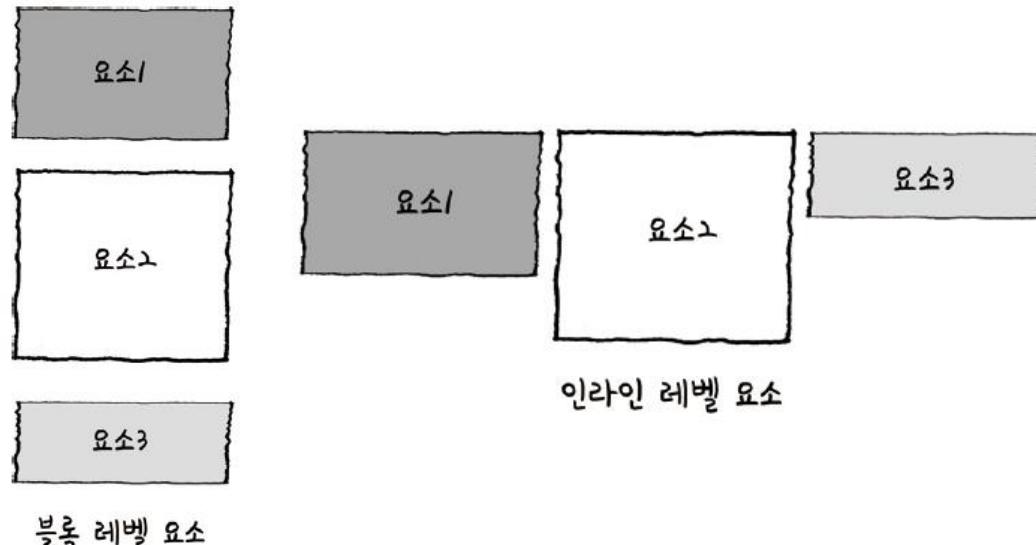
테두리 관련 속성들

여백을 조절하는 속성들



## 블록 레벨 요소

- 요소를 삽입했을 때 혼자 한 줄을 차지하는 요소
- 요소의 너비가 100%
- 예) <div>, <p> 등



## 인라인 레벨 요소

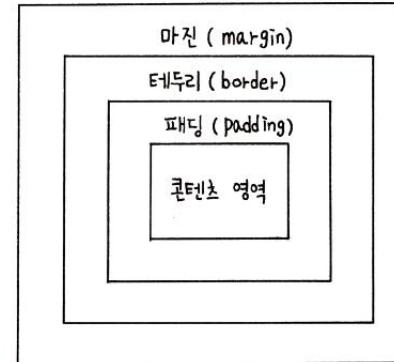
- 줄을 차지하지 않는 요소
- 화면에 표시되는 콘텐츠만큼만 영역을 차지하고 나머지 공간에는 다른 요소가 올 수 있음
- 예) <img>, <strong> 등

**<h3>**시간이란..</h3>  
**<p>**내일 죽을 것처럼 오늘을 살고</p>  
**<p>**영원히 살 것처럼 **<span>**내일을 꿈꾸어라.</span></p>



## 박스 모델

실제 콘텐츠 영역, 패딩(padding), 박스의 테두리(border), 그리고 마진(margin) 등의 요소로 구성됨.



개발자 도구 창에서 박스 모델 확인 가능

The screenshot shows the Google Chrome DevTools interface. The left pane displays the DOM tree with the following structure:

```
<!DOCTYPE html>
<html lang="ko">
  <head>..</head>
  <body>
    <div id="wrapper">
      <header>..</header>
      <section>
        <h2>강아지 용품 준비하기</h2>
        <article class="at1">
          <h3>강아지 집 </h3>
          <p>강아지 집</p>
        </article>
        <article class="at2">..</article>
        <article class="at3">..</article>
        <article class="at4">..</article>
      </section>
      <aside>..</aside>
    </body>
  </html>
```

The right pane shows the Computed tab of the Styles panel. It highlights the styling for a paragraph element with a bounding box of 250px by 120px. The styling includes:

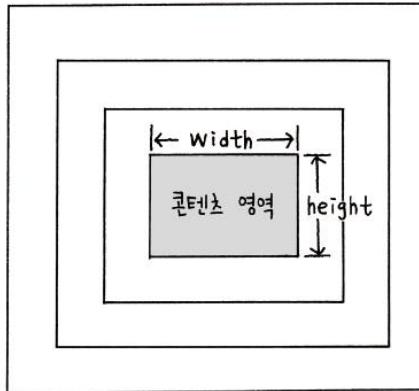
- margin: 12px
- border: none
- padding: 0px
- display: block
- font-family: inherit

# CSS와 박스 모델

## width, height 속성

실제 콘텐츠 영역의 크기 지정

```
기본형 width: <크기> | <백분율> | auto  
height: <크기> | <백분율> | auto
```



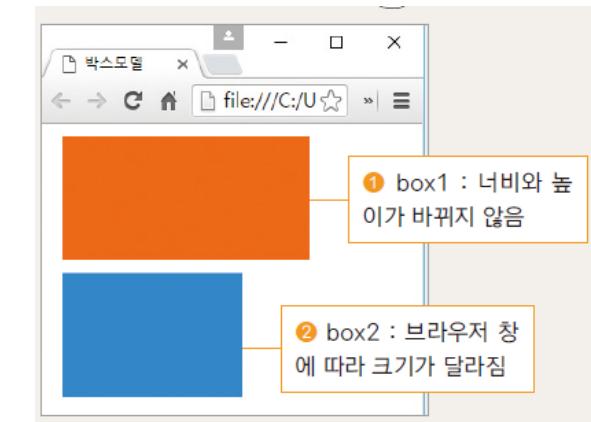
속성 값	설명
<크기>	너비나 높이 값을 px(픽셀)이나 cm(센티미터) 같은 단위와 함께 수치로 지정합니다.
<백분율>	박스 모델을 포함하는 부모 요소를 기준으로 너비나 높이 값을 백분율(%)로 지정합니다.
auto	박스 모델의 너비와 높이 값이 콘텐츠 양에 따라 자동으로 결정됩니다. 기본 값입니다.★

## 실제 콘텐츠 너비 계산하기

모던 브라우저에서 박스 모델의 전체 너비 = width 값 + 좌우 패딩 + 좌우 테두리

인터넷 익스플로러 6에서 박스 모델의 width 값 = 콘텐츠 너비 + 좌우 패딩 + 좌우 테두리

```
<style>  
.box1{  
    width:200px; /* 고정 너비 */  
    height:100px; /* 높이 */  
    background:#ff6a00; /* 배경색 */  
}  
.box2{  
    width:50%; /* 가변 너비 - 브라우저 창 너비의 50% */  
    height:100px; /* 높이 */  
    background:#0094ff; /* 배경색 */  
}  
div {  
    margin:10px; /* div 간의 여백 */  
}  
</style>
```



## display 속성

블록 레벨 요소를 인라인 레벨 요소로 바꾸거나 인라인 레벨 요소를 블록 레벨 요소로 바꿈

기본형    `display: none | contents | block | inline | inline-block | table | table-cell 등`

### 1) display:block

해당 요소를 블록 레벨로 지정

```
<style>
#block img {
  display: block;
  margin: 10px;
}
</style>
```



### 2) display:inline

블록 레벨 요소를 인라인 레벨로 지정

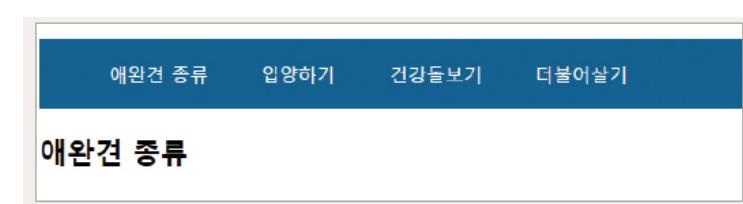
```
<style>
nav ul li {
  display: inline;
}
</style>
```

[애완견 종류](#) [입양하기](#) [건강돌보기](#) [더불어살기](#)

### 3) display:inline-block

요소를 인라인 레벨로 배치하면서 내용에는 블록 레벨 속성을 지정

```
<style>
nav ul li {
  display: inline-block;
  margin: 20px;
}
</style>
```



## display 속성

### 4) display:none

해당 요소를 화면에 표시하지 않음  
화면에서 공간도 차지하지 않음

### 5) 기타 display 속성 값

속성 값	설명
inherit	상위 요소의 display 속성을 상속받습니다.
table	블록 레벨의 표로 만듭니다.
inline-table	인라인 레벨의 표로 만듭니다(<table> 태그 사용한 것처럼).
table-row	표의 행으로 만듭니다(<tr> 태그 사용한 것처럼).
table-row-group	표의 행 그룹으로 만듭니다(<tbody> 태그 사용한 것처럼).
table-header-group	표의 제목 영역(header) 그룹으로 만듭니다(<thead> 태그 사용한 것처럼).
table-footer-group	표의 요약 영역(footer) 그룹으로 만듭니다(<tfoot> 태그 사용한 것처럼).
table-column	표의 열로 만듭니다(<col> 태그 사용한 것처럼).
table-column-group	표의 열 그룹으로 만듭니다(<colgroup> 태그 사용한 것처럼).
table-cell	표에서 하나의 셀로 만듭니다(<td>나 <th> 태그 사용한 것처럼).
table-caption	표의 캡션을 만듭니다(<caption> 태그 사용한 것처럼).
list-item	목록의 항목을 표시할 수 있도록 기본적인 블록 박스와 표시자 박스를 만듭니다(<li> 태그 사용한 것처럼). ▶ ‘기본적인 블록 박스’란 항목의 내용이 표시되는 부분이며 ‘표시자 박스’란 불릿이 표시되는 부분입니다.

# 테두리 관련 속성들

## border-style 속성

- 기본 값이 none → 화면에 테두리 표시안됨
- 테두리를 그리기 위해서는 맨 먼저 테두리 스타일부터 지정

속성 값	설명	예시
none	테두리가 나타나지 않습니다. 기본 값입니다.★	
hidden	테두리가 나타나지 않습니다. border-collapse:collapse일 경우, 다른 테두리도 표시되지 않습니다.	
dashed	테두리를 짧은 선(직선으로 된 점선)으로 표시합니다.	
dotted	테두리를 점선으로 표시합니다.	
double	테두리를 이중선(겹선)으로 표시합니다. 두 선 사이의 간격은 border-width 값으로 지정합니다.	
groove	테두리를 창에 조각한 것처럼 표시합니다. 흙이 파인 듯 입체적으로 보입니다.	
inset	border-collapse:separate일 경우, 전체 박스 테두리가 창에 박혀 있는 것처럼 표시되고 border-collapse:collapse일 경우, groove와 똑같이 표시됩니다.	
outset	border-collapse:separate일 경우, 전체 박스 테두리가 창에서 튀어나온 것처럼 표시되고 border-collapse:collapse일 경우, ridge와 똑같이 표시됩니다.	
ridge	테두리를 창에서 튀어나온 것처럼 표시합니다.	
solid	테두리를 실선으로 표시합니다.	

기본형 border-style: none | hidden | dashed | dotted | double | groove | inset | outset | ridge | solid

```
<style>
.box1 { border-style:solid; } /* 실선 */
.box2 { border-style:dotted; } /* 점선 */
.box3 { border-style:dashed; } /* 선으로 된 점선 */
</style>
```



# 테두리 관련 속성들

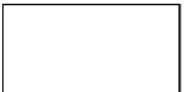
## border-width 속성

테두리 두께 지정

```
기본형 border-top-width: <크기> | thin | medium | thick
border-right-width: <크기> | thin | medium | thick
border-bottom-width: <크기> | thin | medium | thick
border-left-width: <크기> | thin | medium | thick
border-width: <크기> | thin | medium | thick
```

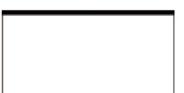
border-width 속성 값이

1) 1개라면 : 네 방향에 모두 같은 두께



```
.box1 { border-width:2px; }
```

2) 2개라면 : 위아래, 좌우 둘어서



```
.box2 { border-width:thick thin; }
```

3) 4개라면 : top → right → bottom → left



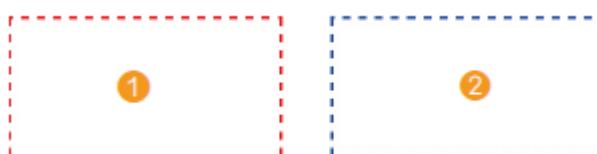
```
.box3 { border-width:5px 10px 15px 20px; }
```

## border-color 속성

테두리 색상 지정

```
기본형 border-top-color: <색상>
border-right-color: <색상>
border-bottom-color: <색상>
border-left-color: <색상>
border-color: <색상>
```

```
<style>
div {
    ...
    border-style:dashed; /* 테두리 스타일 - 선으로 된 점선 */
    border-width:2px;      /* 테두리 굵기 - 2px */
}
.box1 { border-color:red; } /* 색상 - 빨강 */
.box2 { border-color:blue; } /* 색상 - 파랑 */
</style>
```



# 테두리 관련 속성들

## border 속성

- 테두리 스타일과 두께, 색상 등을 둘어 표기
- 순서는 상관없음

기본형 border-top: <두께> | <색상> | <스타일>  
 border-right: <두께> | <색상> | <스타일>  
 border-bottom: <두께> | <색상> | <스타일>  
 border-left: <두께> | <색상> | <스타일>  
 border: <두께> | <색상> | <스타일>

```
<style>
h1 {
  padding-bottom: 5px;
  border-bottom: 3px solid #ccc; /* 아랫 부분 - 3px짜리 회색 실선*/
}
p {
  padding: 10px;
  border: 2px dotted black;      /* 모든 방향 - 3px짜리 검정 점선 */
}
</style>

<h1>박스 모델</h1>
<p>박스 모델은 실제 콘텐츠 영역 ..... 있습니다. </p>
```

## 박스 모델

박스 모델은 실제 콘텐츠 영역, 박스와 콘텐츠 영역 사이의 여백인 패딩(padding), 박스의 테두리(border), 그리고 여러 박스 모델 간의 여백인 마진(margin) 등의 요소로 구성되어 있습니다.

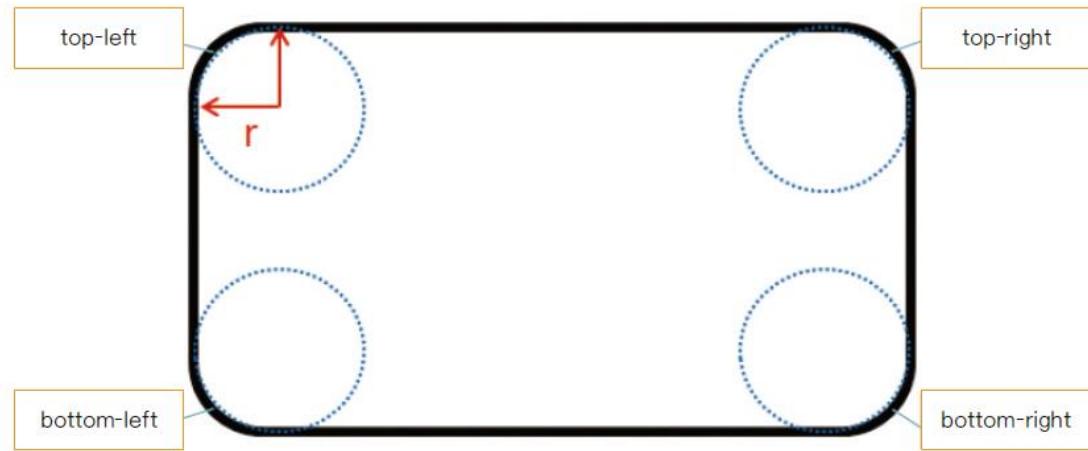
# 테두리 관련 속성들

## border-radius 속성

박스 모서리 부분을 둥글게 처리

기본형

- border-top-left-radius: <크기> | <백분율>
- border-top-right-radius: <크기> | <백분율>
- border-bottom-right-radius: <크기> | <백분율>
- border-bottom-left-radius: <크기> | <백분율>
- border-radius: <크기> | <백분율>



```
<style>
.round {
    border:2px solid red; /* 2px짜리 빨강 실선 */
    border-radius:20px; /* 모서리 20px 만큼 라운딩 */
}
#bg {
    background:url(images/pic1.jpg) no-repeat; /* 배경 이미지 */
    background-size:cover; /* 영역을 다 채우게*/
}
</style>
```



# 테두리 관련 속성들

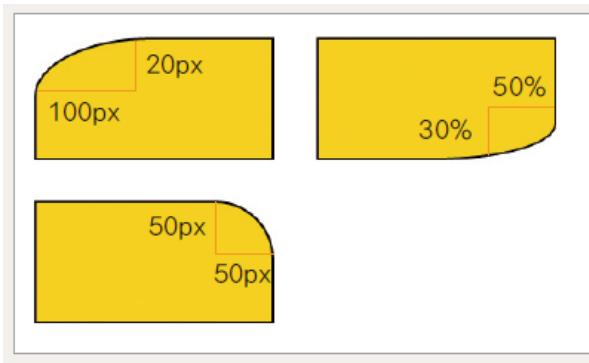
## border-radius 속성

가로 반지름 크기와 세로 반지름 크기를 함께 지정

기본형

```
border-top-left-radius: <가로 크기> <세로 크기>
border-top-right-radius: <가로 크기> <세로 크기>
border-bottom-right-radius: <가로 크기> <세로 크기>
border-bottom-left-radius: <가로 크기> <세로 크기>
border-radius: <가로 크기> / <세로 크기>
```

```
<style>
.round1 { border-top-left-radius:100px 20px; } /* 왼쪽 위 라운딩 */
.round2 { border-bottom-right-radius:50% 30%; }
.round3 { border-top-right-radius:50px; }
</style>
```



# 테두리 관련 속성들

## box-shadow 속성

선택한 요소에 그림자 효과 내기

```
기본형 box-shadow : none | <그림자 값> [, <그림자 값>]*;
<그림자 값> = <수평 거리> <수직 거리> <흐림 정도>
<번짐 정도> <색상> inset
```

수평 거리와 수직 거리는 필수, 기타 속성 같은 옵션.

속성 값	설명
<수평 거리>	그림자의 수평 옵셋 거리(수평으로 얼마나 떨어져 있는지)입니다. 양수 값은 요소의 오른쪽, 음수 값은 요소의 왼쪽에 그림자를 만듭니다. 필수 속성입니다.
<수직 거리>	그림자의 수직 옵셋 거리(세로로 얼마나 떨어져 있는지)입니다. 양수 값은 요소의 아래쪽, 음수 값은 요소의 위쪽에 그림자를 만듭니다. 필수 속성입니다.
<흐림 정도>	그림자의 흐림 정도(blur radius)를 지정합니다. 이 값을 생략하면 0을 기본 값으로 해 진한 그림자를 표시합니다. 이 값이 커질수록 부드러운 그림자를 표시하며 음수 값은 사용할 수 없습니다.
<번짐 정도>	그림자의 번지는 정도를 나타냅니다. 양수 값을 사용하면 그림자가 모든 방향으로 퍼져 나가기 때문에 그림자가 박스보다 크게 표시됩니다. 반대로 음수 값은 그림자가 모든 방향으로 축소되어 보입니다. 기본 값은 0입니다.
<색상>	그림자의 색상을 지정합니다. 한 가지만 지정할 수도 있고 공백으로 구분해 여러 개의 색상을 지정할 수도 있습니다. 필요한 경우에만 사용하는 옵션 값이며 기본 값은 현재 글자 색입니다.
inset	이 키워드를 함께 표시하면 안쪽 그림자로 그립니다. 필요한 경우에만 사용하는 옵션 값입니다.

```
<style>
.box1{ box-shadow:2px -2px 5px 0px black;}
.box2{ box-shadow:5px 5px 15px 5px gray;}
</style>
```



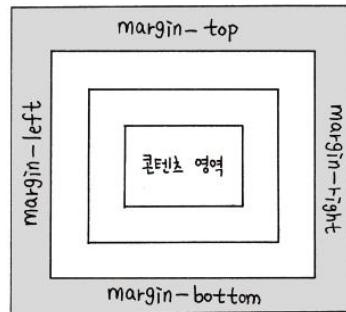
# 여백을 조절하는 속성들

## margin 속성

- 현재 요소 주변의 여백
- 마진을 이용하면 요소와 요소 간의 간격 조절 가능

기본형

```
margin-top: <크기> | <백분율> | auto
margin-right: <크기> | <백분율> | auto
margin-bottom: <크기> | <백분율> | auto
margin-left: <크기> | <백분율> | auto
margin: <크기> | <백분율> | auto
```



속성 값	설명
<크기>	너비나 높이 값을 px(픽셀)이나 cm(센티미터) 같은 단위와 함께 수치로 지정합니다. 예 margin:10px;
<백분율>	박스 모델을 포함하고 있는 부모 요소를 기준으로 너비나 높이 값을 %로 지정합니다. 예 margin:0.1%;
auto	display 속성에서 지정한 값에 맞게 적절한 값을 자동으로 지정합니다.

- 1) margin 속성 값이 1개라면

→ 네 방향 모두에 같은 값 적용

```
p { margin: 50px; } /* 네 방향 마진 모두 50px */
```

- 2) margin 속성 값이 2개라면

→ 첫번째 값은 위아래, 두번째 값은 좌우 마진 값

```
p { margin: 30px 50px; } /* 위아래 마진 - 30px, 좌우 마진 - 50px */
```

- 3) margin 속성 값이 3개라면

→ 빠진 값은 마주 보는 방향의 속성 값 사용

```
p { margin: 30px 20px 50px; } /* 위 마진 - 30px, 좌우 마진 - 20px, 아래 마진 - 50px */
```

- 4) margin 속성 값이 4개라면

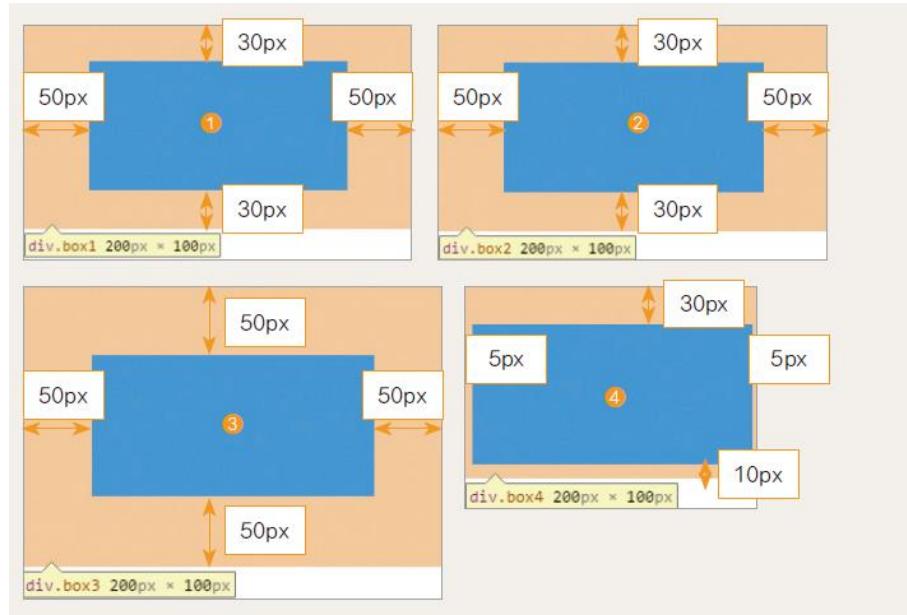
→ top, right, bottom, left 순으로 적용

```
p { margin: 30px 50px 30px 50px; } /* 위아래 마진 - 30px, 좌우 마진 - 50px */
```

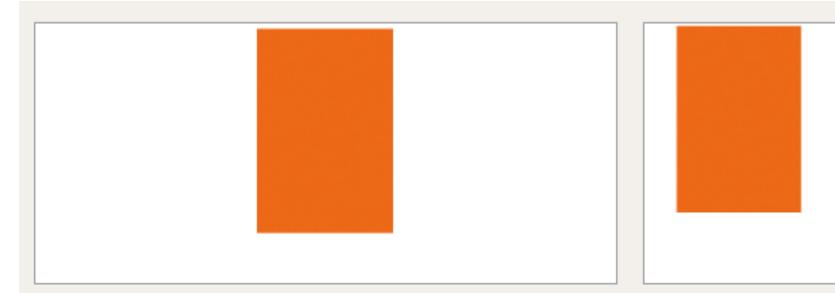
# 여백을 조절하는 속성들

## margin 속성

```
<style>
.box1 { margin:30px 50px 30px 50px; }
.box2 { margin:30px 50px; }
.box3 { margin:50px; }
.box4 { margin:30px 5px 10px; }
</style>
```



```
<style>
.box {
    width:200px;           /* 너비 */
    height:300px;          /* 높이 */
    background:#ff6a00;     /* 배경색 */
    margin:0 auto;          /* 마진 - 0 auto 0 auto */
}
</style>
```



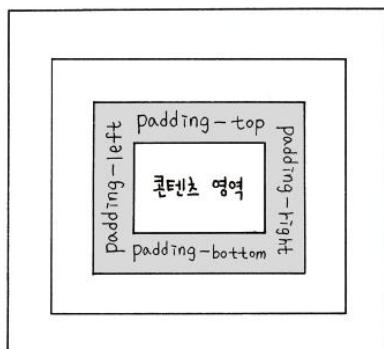
# 여백을 조절하는 속성들

## padding 속성

- 콘텐츠 영역과 테두리 사이의 여백(테두리 안쪽 여백)

기본형

```
padding-top: <크기> | <백분율> | auto
padding-right: <크기> | <백분율> | auto
padding-bottom: <크기> | <백분율> | auto
padding-left: <크기> | <백분율> | auto
padding: <크기> | <백분율> | auto
```



```
<style>
div {
    width:200px;           /* 너비 */
    height:auto;           /* 높이 */
    background:#0094ff;    /* 배경색 */
    display:inline-block;   /* 가로로 배치 */
    margin:15px;           /* 마진 - 15px 15px 15px 15px */
    color:white;           /* 글자색 */
}
.box1 { padding:10px 30px 10px 30px; }
.box2 { padding:10px 30px; }
.box3 { padding:10px; }
</style>
```

패딩(padding)이란 콘텐츠 영역과 테두리 사이의 여백을 말합니다.

패딩(padding)이란 콘텐츠 영역과 테두리 사이의 여백을 말합니다.

패딩(padding)이란 콘텐츠 영역과 테두리 사이의 여백을 말합니다.

파란색 부분은 콘텐츠 영역이고 초록색 부분은 패딩 영역입니다.

# 11. CSS 포지셔닝

CSS 포지셔닝과 주요 속성들

다단으로 편집하기

표 스타일



A screenshot of Visual Studio Code showing the CSS code for a website. The code includes styles for a header, navigation menu, and content section. The code is annotated with numbers from 100 to 337, likely indicating line numbers or specific styling rules.

```
100 .choice .photo {  
101     float: right;  
102     margin-right: 5%;  
103 }  
104 .choice .text {  
105     float: left;  
106 }  
107 .footer {  
108     clear: both;  
109 }  
110 }  
111 </style>  
112 </head>  
113 <body>  
114 <div id="container">  
115     <header>  
116         <nav>  
117             <ul id="main-nav">  
118                 <li><a href="#intro">끼제 소개</a></li>  
119                 <li><a href="#map">오시는 길</a></li>  
120                 <li><a href="#choice">이 달의 추천</a></li>  
121             </ul>  
122         </nav>  
123     </header>  
124     <section id="intro">  
125         <div class="page-title">  
126             <h1>끼제 소개</h1>  
127         </div>  
128         <div class="content">  
129             <img class="photo">  
130               
131     </section>
```

# CSS 포지셔닝과 주요 속성들

## CSS 포지셔닝이란

CSS를 웹 문서 요소를 적절히 배치하는 것

### HTML 마크업으로 작성한 웹 문서



## CSS 포지셔닝

### 우리가 보는 웹 사이트



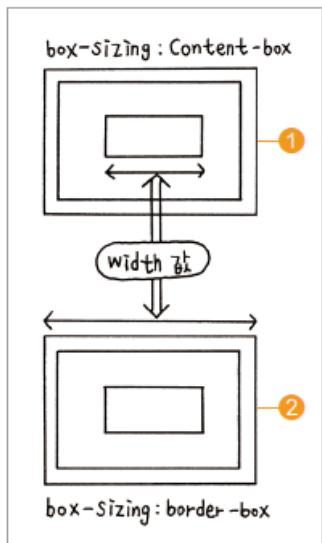
# CSS 포지셔닝과 주요 속성들

## box-sizing 속성

박스 모델의 너비 값 기준 지정

기본형 box-sizing: content-box | border-box

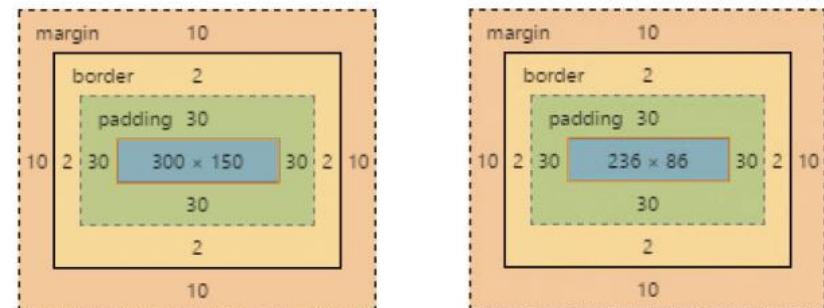
속성 값	설명
content-box	width 속성 값을 콘텐츠 영역 너비 값으로 사용합니다. 기본 값입니다.★
border-box	width 속성 값을 콘텐츠 영역에 테두리까지 포함한 박스 모델 전체 너비 값으로 사용합니다.



```
<style>
.box1 {
  box-sizing:content-box; /* 콘텐츠 영역 기준 */
  width: 300px;           /* 콘텐츠 영역 너비 300px */
  height: 150px;          /* 높이 */
  margin: 10px;           /* 마진 */
  padding: 30px;          /* 패딩 */
  border:2px solid red;   /* 테두리 */
}

.box2 {
  box-sizing:border-box;  /* 테두리까지(박스 전체) 기준*/
  width: 300px;           /* 박스 모델 전체 너비 */
  height: 150px;          /* 박스 높이 */
  margin: 10px;           /* 마진 */
  padding: 30px;          /* 패딩 */
  border: 2px solid red;  /* 테두리 */
}

</style>
```



# CSS 포지셔닝과 주요 속성들

## float 속성

요소를 왼쪽이나 오른쪽에 떠 있게 만듦

기본형 float: left | right | none

속성 값	설명
left	해당 요소를 문서의 왼쪽으로 배치합니다.
right	해당 요소를 문서의 오른쪽으로 배치합니다.
none	좌우 어느 쪽으로도 배치하지 않습니다.

```
<style>
  .box1 { float:left; }
  .box2 { float:left; }
  .box3 { float:left; }
  .box4 { float:right; }
</style>
```



## clear 속성

float 속성을 무효화 시키는 속성

기본형 clear: none | left | right | both

```
<style>
  .box1 { float:left; }
  .box2 { float:left; }
  .box3 { }
  .box4 { clear:both; }
</style>
```



# CSS 포지셔닝과 주요 속성들

## position 속성

웹 문서 안에 요소들을 배치하기 위한 속성

기본형 position: static | relative | absolute | fixed

속성 값	설명
static	요소를 문서의 흐름에 맞추어 배치합니다.★
relative	이전 요소에 자연스럽게 연결해 배치하되 위치를 지정할 수 있습니다.
absolute	원하는 위치를 지정해 배치합니다.
fixed	지정한 위치에 고정해 배치합니다. 화면에서 요소가 잘릴 수도 있습니다.

### 1) static

- 문서의 흐름대로 배치한다
- left 속성이나 top 속성을 지정할 수 없다.

```
<style>
.box1{
  float:left;
  width:100px;
  top:300px;
  background:#ffd800;
  margin-right:10px;
  padding:20px;
}
.box2 {
  width:300px;
  background:#0094ff;
  float:left;
  padding:20px;
}
</style>
```



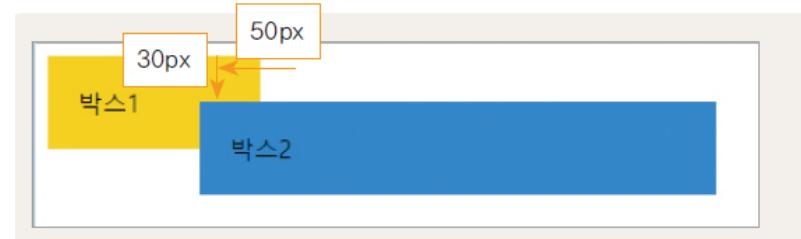
# CSS 포지셔닝과 주요 속성들

## position 속성

### 2) relative

- 자연스럽게 배치
- 고정되어 있지 않고 다른 요소에 의해 바뀔 수 있다.
- 상대적인 위치를 사용하기 때문에 다른 요소와 조화를 이룬다.
- left나 top 속성을 이용해 요소의 위치를 옮길 수도 있다.

```
<style>
.box1{
    float:left;
    width:100px;
    background:#ffd800;
    margin-right:10px;
    padding:20px;
}
.box2 {
    position:relative;
    left:-50px;
    top:30px;
    width:300px;
    background:#0094ff;
    float:left;
    padding:20px;
}
</style>
```

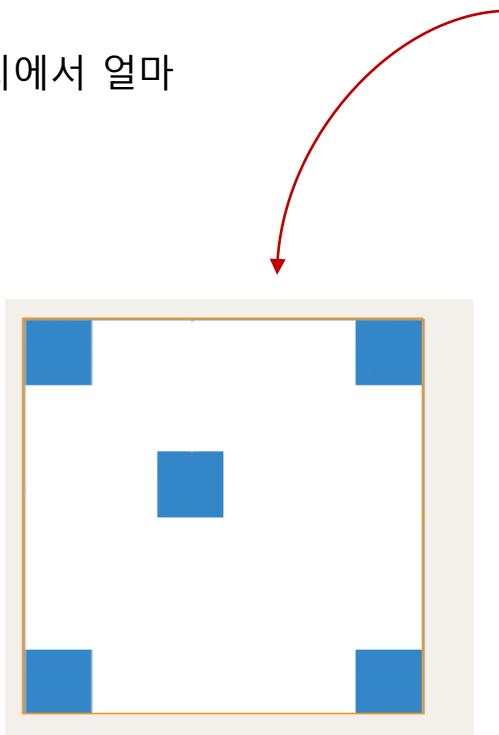


# CSS 포지셔닝과 주요 속성들

## position 속성

### 3) absolute

- 문서의 흐름과는 상관없이 원하는 위치에 요소를 배치
- 요소의 위치는 가장 가까운 부모 요소나 조상 요소 중 position:relative인 요소
- left, top, right, bottom 속성을 사용해 네 모서리에서 얼마나 떨어져 있는지 지정



```
<style>
#wrap{
    position: relative;
    width: 300px;
    height: 300px;
    border: 1px solid #ccc;
}
.box{
    position: absolute;
    width: 50px;
    height: 50px;
    background: #0094ff;
}
.crd1 { top:0; left:0; }
.crd2 { top:0; right:0; }
.crd3 { bottom:0; left:0; }
.crd4 { bottom:0; right:0; }
.crd5 { top:100px; left:100px; }
</style>
```

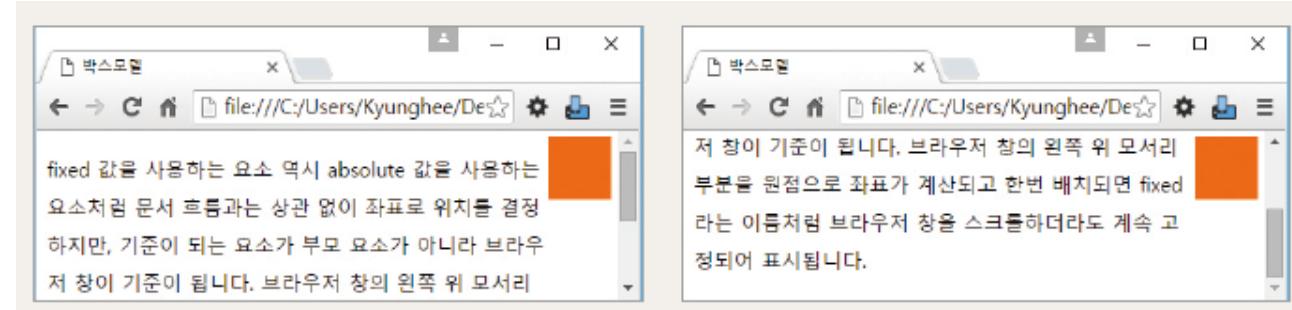
# CSS 포지셔닝과 주요 속성들

## position 속성

### 4) fixed

- 문서의 흐름과는 상관없이 원하는 위치에 요소를 배치
- 부모 요소가 아닌 브라우저 창 기준  
→ 브라우저 창 왼쪽 위 꼭지점(0,0) 기준으로 좌표 계산
- 브라우저 창 화면을 스크롤하더라도 계속 같은 위치에 고정

```
<style>
#fx{
  position:fixed;
  top:5px;
  right:5px;
  width:50px;
  height:50px;
  background:#ff6a00;
}
</style>
```



# CSS 포지셔닝과 주요 속성들

## visibility 속성

특정 요소를 화면에 보이거나 보이지 않게 설정하는 속성

기본형    visibility: visible | hidden | collapse

속성 값	설명
visible	화면에 요소를 표시합니다. 기본 값입니다.★
hidden	화면에서 요소를 감춥니다. 하지만 크기는 그대로 유지하기 때문에 배치에 영향을 미칩니다.
collapse	표의 행, 열, 행 그룹, 열 그룹 등에서 지정하면 서로 겹치도록 조절합니다. 그 외의 영역에서 사용하면 'hidden'처럼 처리합니다.

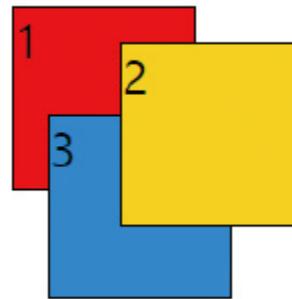
```
<style>
img {
  margin:10px;
  padding:5px;
  border:1px solid black;
}
.invisible {
  visibility:hidden;
}
</style>
```



## z-index 속성

- 요소 쌓는 순서 정하기
- z-index 값이 크면 값이 작은 요소보다 위에 쌓인다.
- z-index 값을 명시하지 않으면 1부터 시작해서 1씩 커진다

```
<style>
div#wrapper {
  position: relative;
}
.b1 { z-index:1; }
.b2 { z-index:3; }
.b3 { z-index:1; }
</style>
```



# 다단으로 편집하기

## column-width

- 단의 너비를 고정해 놓고 화면 분할
- 화면이 커지면 단의 개수가 많아진다.

기본형    column-width: <크기> | auto

```
<style>
.multi {
    -webkit-column-count: 3;
    -moz-column-count: 3;
    column-count: 3;
}
</style>
<div class="multi"> ..... </div>
```



## column-count

- 단의 개수를 먼저 정해 놓고 화면 분할
- 화면이 커질수록 단의 너비가 넓어진다.

기본형    column-count: <숫자> | auto

```
<style>
div#wrapper {
    position: relative;
}
.b1 { z-index:1; }
.b2 { z-index:3; }
.b3 { z-index:1; }
</style>
```

# 다단으로 편집하기

## 다단 위치 지정

특정 요소의 앞이나 뒤 (주로 인쇄 목적)

- break-before:column,
  - break-before:avoid-column
  - break-after:column,
  - bread-after:avoid-column

건강에 좋은 식품 - Super Food	
<b>코코넛 오일</b>	남미 파나마에서는 “코코넛오일 한 컵이면 모든 병이 낫는다”라고 이야기할 정도로 그 효능이 뛰어나다. 코코넛 오일에 들어있는 중성지방산은 글리세롤을 수지에 나쁜 영향을 주지 않는 건강한 지방산이다. 또 라우르산이 풍부해 세균을 물리치고 염증을 가라앉히는 효과가 있다. 하루 권장량은 성인기준 2~4스푼(30~40mL) 정도이다.
<b>아보카도</b>	<b>케일</b>
<pre> &lt;style&gt; .multi {     column-count: 4;     column-rule: 2px dotted #ccc;     text-align: justify; } .multi h3 {     break-before: column; } &lt;/style&gt;</pre>	<p>나무에서 올리는 비단과 불풀 만두 속에 좋은 지방과 단백질, 비타민 E와 같은 항산화 성분이 풍부한 과일이다. 알파-카르跤이 풍부하게 들어 있어 여러 항암 효과가 기대되는 것으로 알려져 있다. 그뿐만 아니라 세포의 물리적 합성을 도와 피부 단력을 높이는 데 도움을 준다. 하루 권장량은 성인기준 1회이며, 꾸준히 섭취하면 글리세롤을 수지를 낮춰준다.</p> <p>케일은 브로콜리, 양배추, 유채, 콩, 경과 같은 배추과 씨식물로써 세계적인 기구에서 최고의 녹색채소로 선정되었다. 베타카로틴과 면역소가 풍부하게 체내 유전자를 회복하도록 하는 항암효과가 주어진다. 또한, 위암 예방과 치료에 좋은 비타민 U와 유산균을 피하는 베타민 M과 같은 성분이 영양소를 활용하고 있다. 하루 권장량은 성인기준 150g 정도이다.</p>
<b>블루베리</b>	<b>아몬드</b>
	<p>블루베리는 우리 몸이 신체하는 형상을 막아주는 항산화물질이 가장 많은 식품으로 알려졌다. 미국 농무부에서 빛깔, 맛, 향기, 비타민과 같은 영양소로 들어 있었던 아몬드는 블루베리의 “안토시아닌”이라는 항암효과가 주어진다. 시력 향상에 뛰어난 효과가 있다. 그뿐만 아니라 다이어트에도 좋다. 블루베리의 페놀파울리페놀 성분은 체내 지방세포를 줄여주는 역할을 한다. 하루 권장량은 성인기준 50g 전후로, 20~30일 정도이다.</p>

# column-span

## 여러 단을 하나로 합치기

기본형 column-span: 1 | all

속성 값	설명
1	단을 하나만 합치는 것이므로 합치지 않는 것과 같습니다. 기본 값입니다.★
all	전체 단을 하나로 합쳐 표현합니다. 단의 일부만 합칠 수는 없습니다.

```
<style>
.multi {
    column-count: 3;
    column-rule: 2px dotted #00C
}
.no-col{
    column-span:all;
}
</style>
<div class="multi">
.....
<div class="no-col"> ....</div>
</div>
```

# 표 스타일

## caption-side

- 캡션(설명글)은 기본으로 표 위쪽에 표시됨.
- 이 속성을 이용해 아래쪽에 표시 가능

기본형 caption-side: top | bottom

## border

표의 바깥 테두리와 셀 테두리 모두 지정해야 함.

```
<style>
.table1 {
  border:1px solid black;
}
.table1 td {
  border:1px dotted black;
}
</style>
```

프로축구 경기 일정	
울산	울산 vs 인천
부산	부산 vs 대전
서울	서울 vs 강원

## border-collapse

표 테두리와 셀 테두리를 합칠 것인지 설정

기본형 border-collapse: collapse | separate

속성 값	설명
collapse	테두리를 하나로 합쳐 표시합니다.
separate	테두리를 따로 표시합니다. 기본 값입니다.★

```
<style>
.table1 {
  border:1px solid black;
  border-collapse:collapse;
}
.table1 td {
  border:1px dashed black;
}
</style>
```

프로축구 경기 일정	
울산	울산 vs 인천
부산	부산 vs 대전
서울	서울 vs 강원

# 표 스타일

## border-spacing

- border-collapse:separate를 사용해 셀들을 분리했을 경우, 인접한 셀 테두리 사이의 거리를 지정
- 값이 1개 : 수평 거리 & 수직 거리를 같게
- 값이 2개 : 첫번째 값은 수평 거리, 두번째 값은 수직 거리

기본형 border-spacing: <크기>

```
<style>
.table1 {
  border:1px solid black;
  border-collapse:separate;
  border-spacing:20px 10px;
}
.table1 td {
  border:1px solid black;
}
</style>
```

프로축구 경기 일정		
울산	울산 vs 인천	
부산	부산 vs 대전	
서울	서울 vs 강원	

## empty-cell

border-collapse:separate를 사용해 셀들을 분리했을 경우, 내용이 없는 빈 셀들의 표시 여부를 지정

기본형 empty-cells: show | hide

```
<style>
.schedule { border-collapse:separate; }
td { border:1px solid black; }
#tb1 td{ empty-cells:show; }
#tb2 td { empty-cells:hide; }
</style>
```

```
<table class="schedule" id="tb1"> .... </table>
<table class="schedule" id="tb2"> .... </table>
```

프로축구 경기 일정		
울산	울산 vs 인천	TV 중계
부산	부산 vs 대전	
서울	서울 vs 강원	

프로축구 경기 일정		
울산	울산 vs 인천	TV 중계
부산	부산 vs 대전	
서울	서울 vs 강원	

# 표 스타일

## width, height

- 너비나 높이를 지정하지 않으면 셀 안의 내용이 표시될 만큼만 표시된다.
- width 값을 지정할 경우 padding 속성을 이용해 여백을 넣어 주면 보기 좋게 꾸밀 수 있다.

```
<style>
table {
border-collapse:collapse;
width:300px;
}
td {
padding:10px;
}
</style>
```

프로축구 경기 일정	
울산	울산 vs 인천
부산	부산 vs 대전
서울	서울 vs 강원

## table-layout

셀 안의 내용 양에 따라 셀 너비를 변하게 할지, 고정시킬지 결정

기본형 table-layout: fixed | auto

속성 값	설명
fixed	셀 너비를 고정합니다. 즉, 셀 내용에 따라 셀의 너비가 달라지지 않습니다.
auto	셀 내용에 따라 셀의 너비가 달라집니다. 기본 값입니다.*

```
<style>
.table1 {
border-collapse:collapse;
width:300px;
table-layout:fixed;
word-break:break-all;
height:auto;
}
.table1 td {
width:150px;
border:1px solid black;
padding:5px;
}
</style>
```

한글로 띄어쓰기 없기	long_description_without_space
-------------	--------------------------------

# 표 스타일

## text-align

셀 안에서의 수평 정렬 방법

기본형 text-align : left | right | center

```
<style>
.table1 td {
  text-align:center;
}
</style>
```

셀 가운데 정렬			
내용1	내용2	내용3	내용4

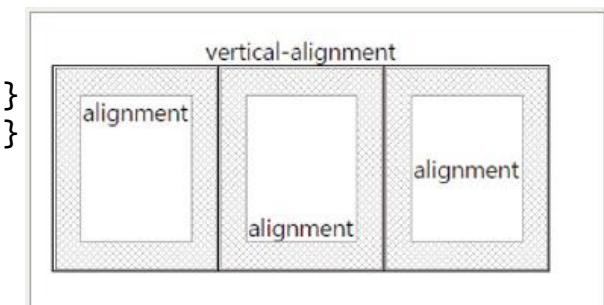
## vertical-align

셀 안에서의 수직 정렬 방법

기본형 vertical-align: top | bottom | middle

속성 값	설명
top	위쪽 패딩 가장자리에 내용의 윗부분을 맞춥니다.*
bottom	아래쪽 패딩 가장자리에 내용의 아랫부분을 맞춥니다.
middle	패딩의 중앙에 내용의 중앙을 맞춥니다.

```
<style>
.va1 { vertical-align:top; }
.va2 { vertical-align:bottom; }
.va3 { vertical-align:middle; }
</style>
```



# 12. 다재다능한 CSS3 선택자

# 연결 선택자

# 속성 선택자

## 가상 클래스와 가상 요소



```
index.html *
1 <html>
2   <head>
3     <title>index result.html</title>
4     <style>
5       .photo {
6         float: right;
7         margin-right: 5%;
8       }
9       .choice_text {
10        float: left;
11      }
12     </style>
13   </head>
14   <body>
15     <div id="container">
16       <header>
17         <nav>
18           <a href="#">카페 소개</a>
19           <a href="#">오시는 길</a>
20           <a href="#">choice</a>
21           <a href="#">이 달의 추천</a>
22         </nav>
23       </header>
24
25       <section id="intro">
26         <div class="page-title">
27           <h1>카페 소개</h1>
28         </div>
29         <div class="content">
30           <img alt="" data-bbox="100px 100px" />
31         </div>
32       </section>
33     </div>
34   </body>
35 </html>
```

# 연결 선택자

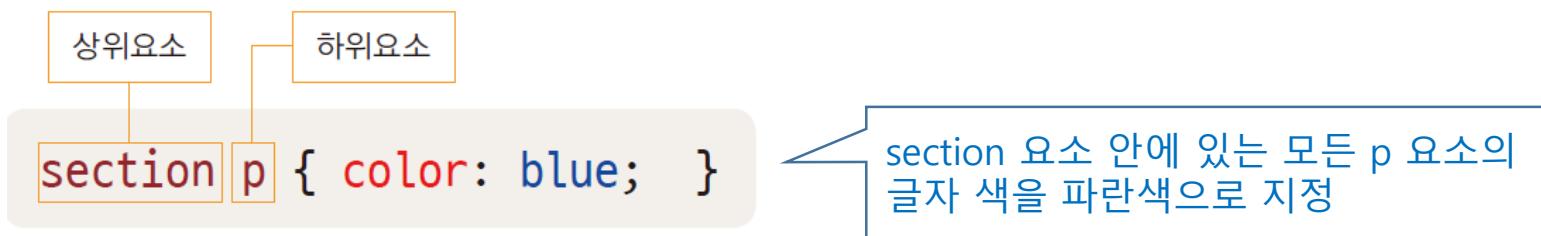
- 연결 선택자 : 선택자와 선택자를 연결해 적용 대상을 한정하는 선택자.
- 컴비네이션 선택자(combination selector) 또는 '조합 선택자'라고도 함

## 하위 선택자

기본형

상위요소 하위요소

- 부모 요소에 포함된 모든 하위 요소에 스타일이 적용된다
- 자식 요소뿐만 아니라 손자 요소, 손자의 손자 요소 등 모든 하위 요소까지 적용
- 하위 선택자를 정의할 때는 상위 요소와 하위 요소를 나란히 쓴다.



# 연결 선택자

## 하위 선택자

```

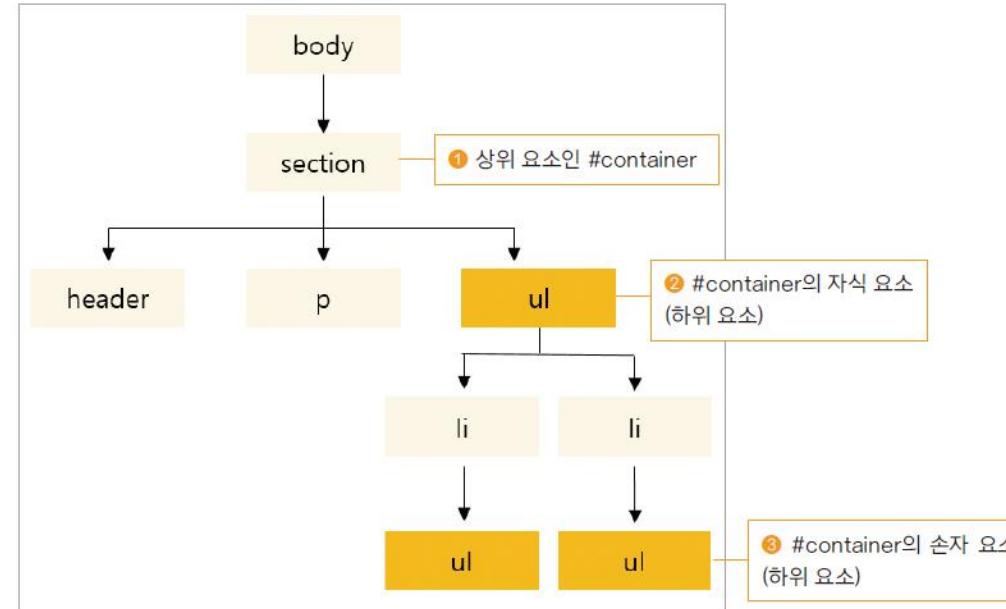
<style>
  #container ul {
    border: 1px dotted blue;
  }
</style>
<section id="container">
  <header><h1>예약 방법 및 요금</h1></header>
  <p>요안도라에 예약하려면?
  <ul>
    <li>예약 방법
      <ul>
        <li>직접 통화</li>
        <li>문자 남기기</li>
      </ul>
    </li>
    <li>요금
      <ul>
        <li>1인 : 40,000원</li>
        <li>2인 : 60,000원</li>
        <li>3인 : 80,000원</li>
        <li>4인 : 100,000원</li>
      </ul>
    </li>
  </ul>
</section>

```

**예약 방법 및 요금**

요안도라에 예약하려면?

- 예약 방법
  - 직접 통화
  - 문자 남기기
- 요금
  - 1인 : 40,000원
  - 2인 : 60,000원
  - 3인 : 80,000원
  - 4인 : 100,000원

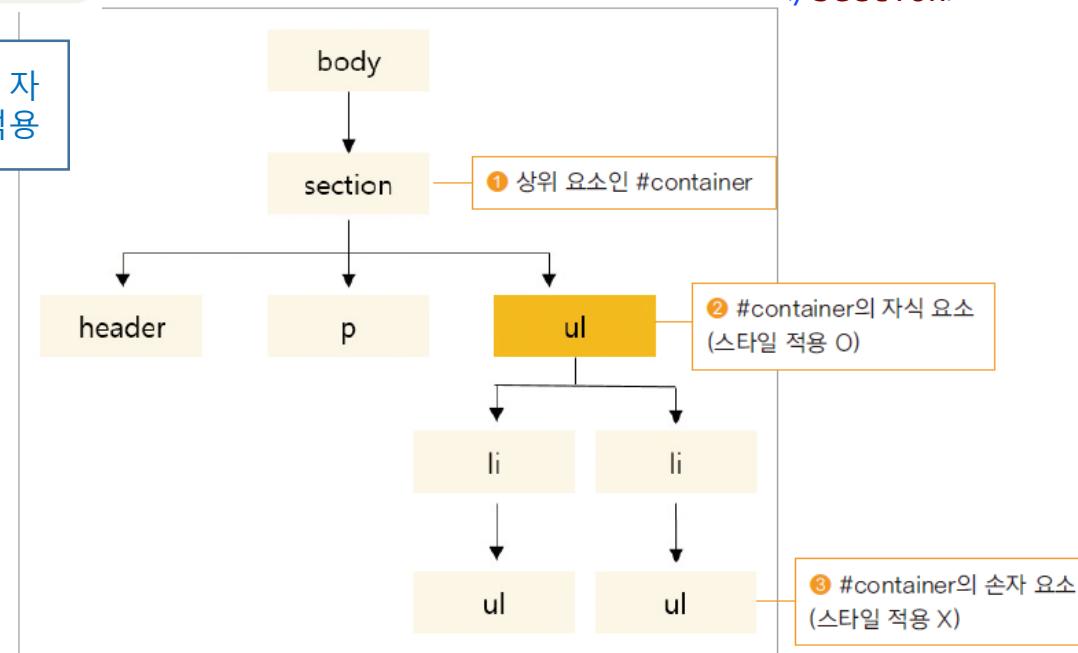


# 연결 선택자

## 자식 선택자

기본형    부모요소 > 자식요소

- 자식 요소에 스타일을 적용하는 선택자
- 두 요소 사이에 '>(부등호)'를 표시해 부모 요소와 자식 요소를 구분



```

<style>
  #container > ul {
    border: 1px dotted blue;
  }
</style>
<section id="container">
  <header><h1>예약 방법 및 요금</h1></header>
  <p>요안도라에 예약하려면?
  <ul>
    <li>예약 방법
      <ul>... ... </ul>
    </li>
    <li>요금
      <ul>... ... </ul>
    </li>
  </ul>
</section>
  
```

### 예약 방법 및 요금

요안도라에 예약하려면?

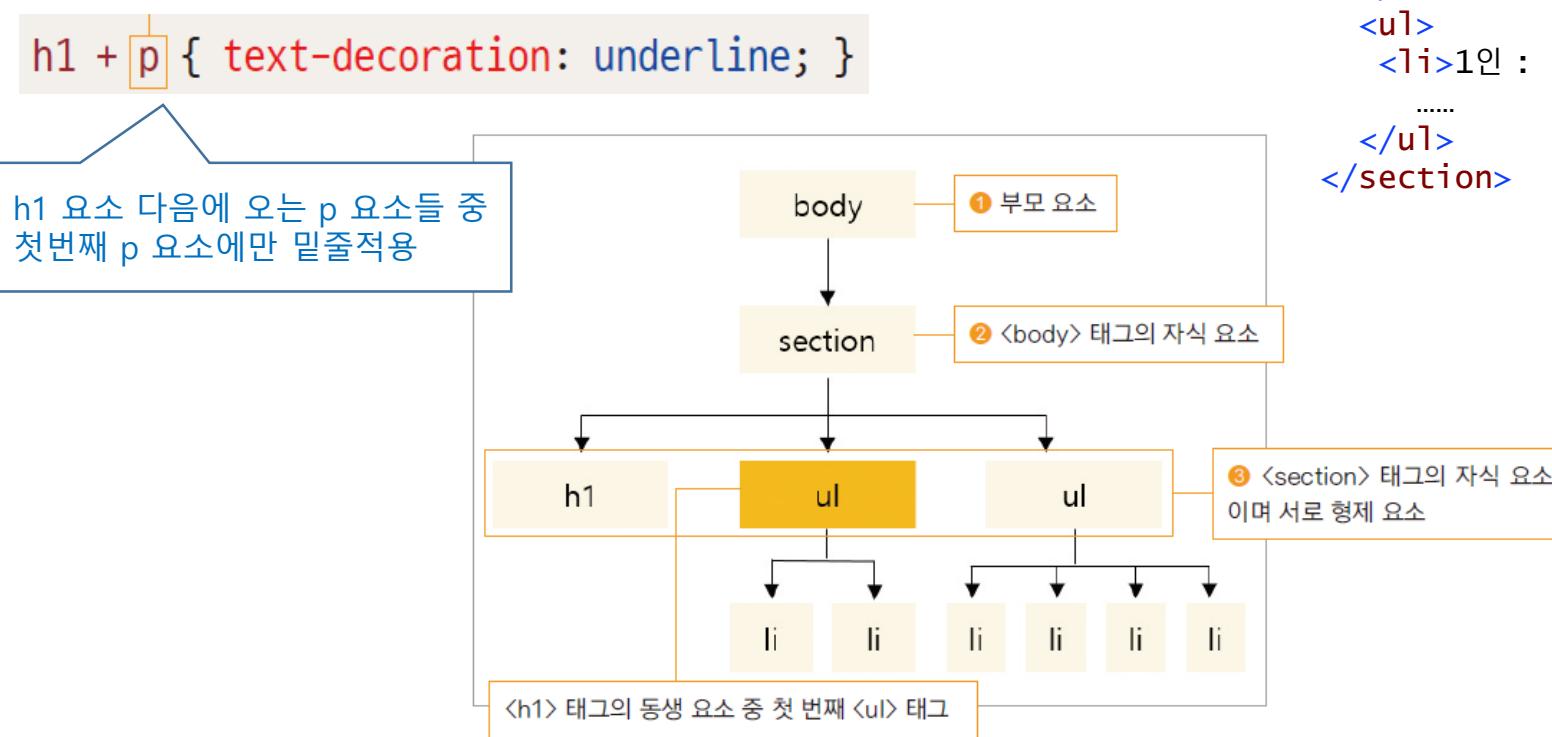
- 예약 방법
  - 직접 통화
  - 문자 남기기
- 요금
  - 1인 : 40,000원
  - 2인 : 60,000원
  - 3인 : 80,000원
  - 4인 : 100,000원

# 연결 선택자

## 인접 형제 선택자

기본형      요소 1 + 요소 2

- 같은 부모를 가진 형제 요소 중 첫 번째 동생 요소에만 스타일 적용
- 요소 1과 요소 2는 같은 레벨이면서 요소 1 이후 맨 먼저 오는 요소 2에 스타일을 적용



```

<style>
h1 + ul {
  color: blue;
  font-weight: bold;
}
</style>

```

```

<section id="container">
  <h1>예약 방법 및 요금</h1>
  <ul>
    <li>직접 통화</li>
    <li>문자 남기기</li>
  </ul>
  <ul>
    <li>1인 : 40,000원</li>
    ....
  </ul>
</section>

```

## 예약 방법 및 요금

- 직접 통화
- 문자 남기기
- 1인 : 40,000원
- 2인 : 60,000원
- 3인 : 80,000원
- 4인 : 100,000원

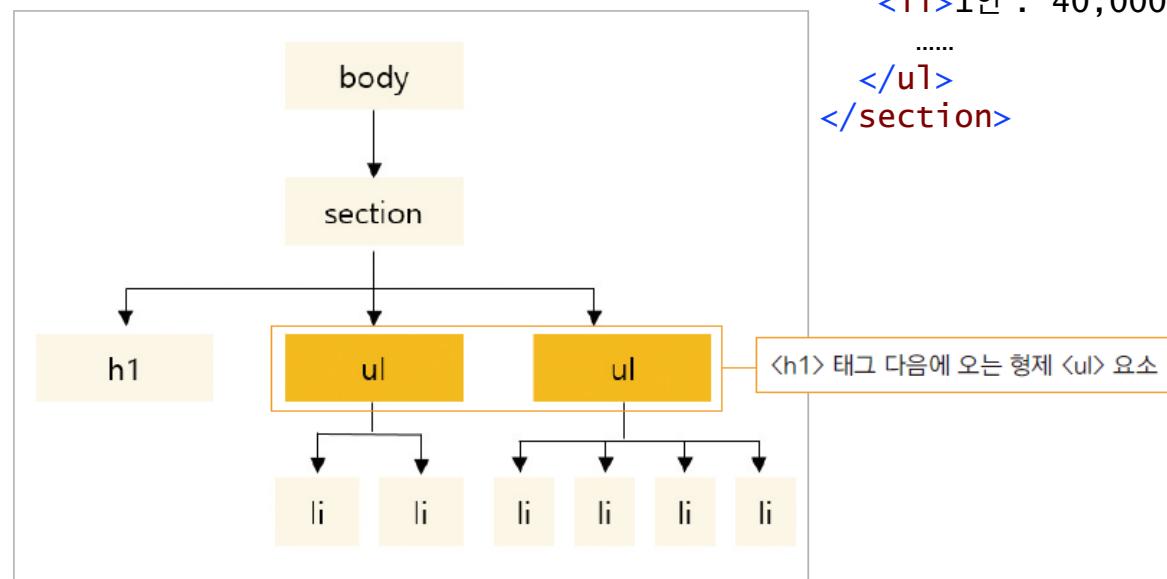
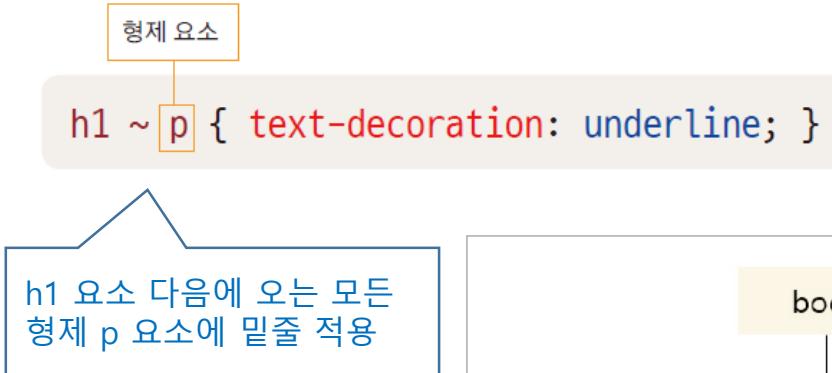
# 연결 선택자

## 형제 선택자

기본형

요소 1 ~ 요소 2

- 형제 요소들에 스타일 적용
- 인접 형제 선택자와 다른 점은 모든 형제 요소에 다 적용된다는 것



```

<style>
  h1 ~ ul {
    color: blue;
    font-weight: bold;
  }
</style>

<section id="container">
  <h1>예약 방법 및 요금</h1>
  <ul>
    <li>직접 통화</li>
    <li>문자 남기기</li>
  </ul>
  <ul>
    <li>1인 : 40,000원</li>
    ....
  </ul>
</section>
  
```

## 예약 방법 및 요금

- 직접 통화
- 문자 남기기
- 1인 : 40,000원
- 2인 : 60,000원
- 3인 : 80,000원
- 4인 : 100,000원

# 속성 선택자

## [속성] 선택자

지정한 속성을 가진 요소를 찾아 스타일 적용

```
<style>
  a[href] {
    background: yellow;
  }
</style>

<body>
  <ul>
    <li><a>메인 메뉴 : </a></li>
    <li><a href="#">메뉴 1</a></li>
    <li><a href="#">메뉴 2</a></li>
    <li><a href="#">메뉴 3</a></li>
    <li><a href="#">메뉴 4</a></li>
  </ul>
</body>
```



## [속성 = 값] 선택자

주어진 속성과 속성 값이 일치하는 요소를 찾아 스타일 적용

```
<style>
  a[target="_blank"] {
    padding-right: 30px;
    background: url(images/newwindow.png) no-repeat center right;
  }
</style>

<h1>HTML5 표준안 사이트 </h1>
<ul>
  <li><a href="http://www.w3c.org/TR/html" target="_blank">HTML</a></li>
  <li><a href="https://www.w3.org/TR/selectors">
    CSS Selector Level 3</a></li>
  <li><a href="https://www.w3.org/TR/css3-mediaqueries">미디어쿼리</a>
  </li>
</ul>
```

### HTML5 표준안 사이트



CSS Selector Level 3

미디어쿼리

# 속성 선택자

## [속성~=값] 선택자

여러 속성 값 중에 해당 값이 포함되어 있는 요소를 찾아 스타일 적용

```
<style>
  [class ~="button"] {
    border: 2px solid black;
    box-shadow: rgba(0,0,0,0.4) 5px 5px;
  }
</style>
<body>
  <ul>
    <li><a href="#">메뉴 1</a></li>
    <li><a href="#">메뉴 2</a></li>
    <li><a href="#" class="button">메뉴 3</a></li>
    <li><a href="#" class="flat button">메뉴 4</a></li>
  </ul>
</body>
```

class 속성 값에 button이 포함된 요소를 찾아내 스타일 적용



## [속성 |= 값] 선택자

특정 값이 포함된 속성을 가진 요소를 찾아 스타일 적용  
하이픈으로 연결해 한 단어 값을 이루는 요소에도 적용

```
<style>
  a[title |= "us"] {
    background: url(images/us.png) no-repeat left center;
    padding: 5px 25px;
  }
  a[title |= "jap"] {
    background: url(images/jp.png) no-repeat left center;
    padding: 5px 25px;
  }
</style>

<ul>
  <li>외국어 서비스 : </li>
  <li><a href="#" title="us">영어</a></li>
  <li><a href="#" title="us-english">영어</a></li>
  <li><a href="#" title="japanese">일본어</a></li>
</ul>
```

외국어 서비스 :	영어	영어	일본어
-----------	----	----	-----

# 속성 선택자

## [속성 ^= 값] 선택자

특정 값으로 시작하는 속성을 가진 요소를 찾아 스타일 적용

```
<style>
  a[title ^="eng"] {
    background: url(images/us.png) no-repeat left center;
    padding: 5px 25px;
  }
  a[title ^="jap"] {
    background: url(images/jp.png) no-repeat left center;
    padding: 5px 25px;
  }
  a[title ^="chin"] {
    background: url(images/ch.png) no-repeat left center;
    padding: 5px 25px;
  }
</style>

<ul>
  <li>외국어 서비스 : </li>
  <li><a href="#" title="english">영어</a></li>
  <li><a href="#" title="japanese ">일본어</a></li>
  <li><a href="#" title="chinese">중국어</a></li>
</ul>
```

외국어 서비스 : 영어 일본어 중국어

## [속성 \$= 값] 선택자

특정 값으로 끝나는 속성을 가진 요소를 찾아 스타일 적용

```
<style>
  a[href $= "hwp"] {
    background: url(images/hwp_icon.gif) center right no-repeat;
    padding-right: 25px;
  }
  a[href $= "xls"] {
    background: url(images/excel_icon.gif) center right no-
repeat;
    padding-right: 25px;
  }
</style>

<ul>
  <li><a href="intro.hwp">hwp 파일</a></li>
  <li><a href="intro.xls ">엑셀 파일</a></li>
</ul>
```

### 회사 소개 파일 다운 받기

- hwp 파일
- 엑셀 파일

# 속성 선택자

## [속성\*=값] 선택자

값의 일부가 일치하는 속성을 가진 요소를 찾아 스타일 적용

```
<style>
a[href *= "w3"] {
  background: blue;
  color: white;
}
</style>

<ul>
<li><a href="http://www.w3c.org/TR/html">HTML 표준안 사이</a>
</li>
<li><a href="http://www.webplatform.org">튜토리얼과 아티</a>
</li>
<li><a href="http://caniuse.com">HTML 지원 여부 체크</a></li>
<li><a href="http://www.w3c.org/TR/css3-mediaqueries">
  미디어쿼리</a>
</li>
</ul>
```

기본형	스타일 적용 경우	예시 - 선택자	예시 - 적용되는 요소
[속성]	지정한 '속성'일 때	[href]	<a href="#">메뉴 1</a>
[속성 = 값]	'값'에 일치할 때	[target="_blank"]	<a href="....." target=_blank>HTML</a>
[속성 ~= 값]	여러 값들 중 특정 '값'이 포함되어 있을 때(단어별)	[class~="button"]	<a href="#" class="flat button">메뉴 4</a>
[속성 != 값]	'값'이 포함되어 있을 때 (하이픈 포함 단어별)	[class != "us"]	<a href="#" title="us">영어</a> <a href="#" title="us-english">영어</a>
[속성 ^= 값]	'값'으로 시작할 때	[title^="eng"]	<a href="#" title="english">영어</a>
[속성 \$= 값]	'값'으로 끝날 때	[href \$="xls"]	<a href="intro.xls">엑셀 파일</a>
[속성 *= 값]	'값'이 속성 값의 일부일 때	[href *= "w3"]	<a href="http://www.w3c.org/TR/html">HTML 사이트</a>

외국어 서비스 : 영어

일본어

중국어

# 가상 클래스와 가상 요소

## 사용자 동작에 반응하는 가상 클래스

표기	설명
:link	방문하지 않은 링크에 스타일 적용
:visited	방문한 링크에 스타일 적용
:active	웹 요소를 활성화 했을 때의 스타일 적용
:hover	웹 요소에 마우스 커서를 올려놓을 때의 스타일 적용
:focus	웹 요소에 초점이 맞추어 졌을 때의 스타일 적용



```

<style>
  .navi a:link, .navi a:visited {
    padding: 10px 5px 5px 35px;
    display: block;
    color: #fff;
    width: 150px;
    text-decoration: none;
  }
  .navi a:hover, .navi a:focus {
    text-shadow: 0px 2px 2px #000;
    color: #FC0;
  }
  .navi a:active {
    color: red;
  }
</style>

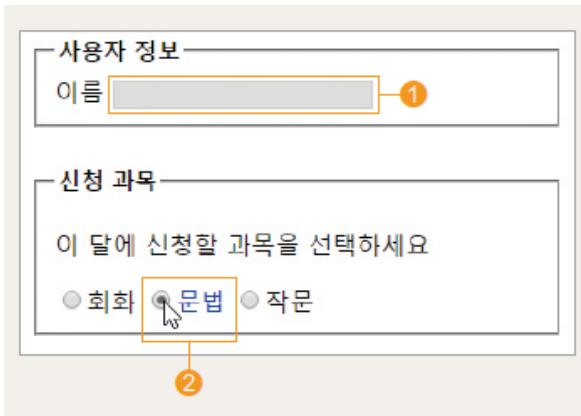
<nav class="navi">
  <ul>
    <li><a href="#">이용 안내</a></li>
    <li><a href="#">객실 소개</a></li>
    <li><a href="#">예약 방법 및 요금</a></li>
    <li><a href="#">예약하기</a></li>
  </ul>
</nav>

```

# 가상 클래스와 가상 요소

## UI 요소 상태에 따른 가상 클래스

표기	설명
:enabled, :disabled	요소를 사용할 수 있을 때와 없을 때의 스타일 지정
:checked	라디오 박스나 체크 박스에서 항목을 선택했을 때의 스타일 지정



```

<style>
  input:disabled {
    background:#ddd;
    border:1px #ccc solid;
  }
  input:checked + span {
    color:blue;
  }
</style>

<p>이 달에 신청할 과목을 선택하세요</p>
<label>
  <input type="radio" name="subject" value="speaking">
  <span>회화</span>
</label>
<label>
  <input type="radio" name="subject" value="grammar">
  <span>문법</span>
</label>
<label>
  <input type="radio" name="subject" value="writing">
  <span>작문</span>
</label>

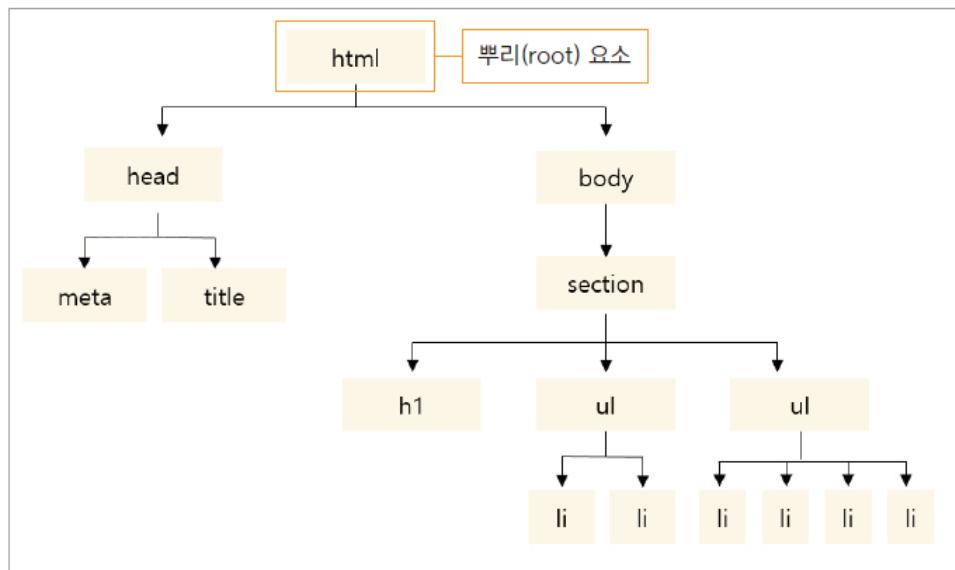
```

# 가상 클래스와 가상 요소

## 구조 가상 클래스

### :root

- 문서 안의 루트(root) 요소에 스타일을 적용
- 웹 문서의 최상위 요소는 <html> 이므로 이 스타일을 이용하면  
웹 문서 전체에 스타일 적용



# 가상 클래스와 가상 요소

## 구조 가상 클래스

### :nth-child(n)와 :nth-last-child(n)

- :nth-child(n) : 앞에서부터 n번째 자식 요소에 스타일 적용
- :nth-last-child(n) : 뒤에서부터 n번째 자식 요소에 스타일 적용
- 위치를 나타낼 때  $an+b$  처럼 수식을 사용할 수도 있음.  
이 때 n 값은 0부터.

```
/* div 요소 안에서 세 번째 자식 요소인 p 요소에 스타일 적용 */
div p:nth-child(3)

/* div 요소 안에서 홀수 번째로 나타나는 자식 요소인 p 요소에 스타일 적용 */
div p:nth-child(odd), div p:nth-child(2n+1)

/* div 요소 안에서 짝수 번째로 나타나는 자식 요소인 p 요소에 스타일 적용 */
div p:nth-child(even), div p:nth-child(2n+0), div p:nth-child(2n)
```

```
<style>
table tr:nth-child(2n+1) {
background: lightgray;
color: black;
}
</style>
```

```
<table border="1">
<tr> <td>블루베리</td></tr>
<tr> <td>귀리</td></tr>
<tr> <td>토마토</td></tr>
.....
</table>>
```

## 건강에 좋은 건강 식품

블루베리
귀리
토마토
시금치
적포도주
건과류
브로콜리
연어
마늘
녹차

# 가상 클래스와 가상 요소

## 구조 가상 클래스

### `:nth-of-type(n), :nth-last-of-type(n)`

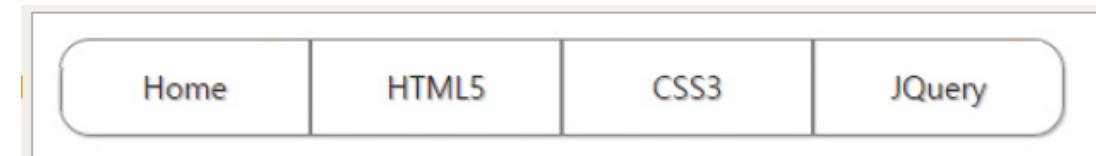
- `:nth-of-type(n)` : 앞에서부터 n번째 요소에 스타일 적용
- `:nth-last-of-type(n)` : 뒤에서부터 n번째 요소에 스타일 적용

### `:first-child, :last-child`

- `:first-child`: 첫번째 자식 요소 스타일 적용
- `:last-child`: 마지막 자식 요소에 스타일 적용

```
<style>
ul.navi li:first-child {
    border-top-left-radius: 1em;
    border-bottom-left-radius: 1em;
}
ul.navi li:last-child {
    border-top-right-radius: 1em;
    border-bottom-right-radius: 1em;
}
</style>

<ul class="navi">
    <li class="home"><a href="#">Home</a></li>
    <li id="html5"><a href="#">HTML5</a></li>
    <li id="css3"><a href="#">CSS3</a></li>
    <li id="jquery"><a href="#">JQuery</a></li>
</ul>
```



# 가상 클래스와 가상 요소

## 구조 가상 클래스

### :first-of-type(n), :last-of-type(n)

- :first-of-type : 형제 요소들 중 첫번째 요소에 스타일 적용
- :last-of-type : 형제 요소들 중 마지막 요소에 스타일 적용

```
p:first-of-type { color: blue; }
p:last-of-type { color: red; }
```

### :only-child, :only-of-type

- :only-child : 부모 요소 안에 자식 요소가 유일하게 하나일 때 스타일 적용
- :only-of-type : 자신이 유일한 요소일 때 스타일 적용

```
p:only-child { color: green; }
```

자식 요소가 오직 p 요소 뿐일 때 p의 글자색을 green으로  
(다른 자식 요소가 있으면 안됨)

```
p:only-of-type { font-weight: bold; }
```

p 요소가 오직 하나 뿐일 때 (다른 자식 요소 있어도 됨)

# 가상 클래스와 가상 요소

## 그 외 가상 클래스

### :target

앵커로 연결된 부분(목적지)에 스타일 지정

```
#intro:target { background-color: yellow; }
```

#intro 앵커가 연결하는 부분의 배경 색을 노랑으로

### :not

괄호 안에 있는 요소를 제외한 부분에 스타일 지정

```
p:not(#ex) { color: blue; }
```

#ex가 아닌 모든 p 요소에서 글자색을 파랑으로

# 가상 클래스와 가상 요소

표기	설명
<code>:target</code>	앵커 목적지에 스타일 적용
<code>:not</code>	특정 요소가 아닐 때 스타일 적용하기
<code>:nth-last-child(n)</code>	부모 요소의 뒤로부터 n번째의 자식 요소에 스타일을 적용한다.
<code>:nth-of-type(n)</code>	같은 유형의 요소 중에서 n 번째 나타나는 요소에 스타일을 적용한다.
<code>:nth-last-of-type(n)</code>	같은 유형의 요소 중에서 끝에서부터 세어 n번째 나타나는 요소에 스타일을 적용한다.
<code>:first-child</code>	첫번째 자식 요소에 스타일을 적용한다.
<code>:last-child</code>	마지막 자식 요소에 스타일을 적용한다.
<code>:first-of-type</code>	형제 중에서 첫번째 요소에 스타일을 적용한다.
<code>:last-of-type</code>	형제 중에서 마지막 요소에 스타일을 적용한다.
<code>:only-child</code>	해당 요소가 유일한 자식 요소일 때 스타일을 적용한다.
<code>:only-of-type</code>	해당 요소가 하나 뿐인 요소에 스타일을 적용한다.

# 가상 클래스와 가상 요소

## 가상 요소

- ::first-line : 특정 요소의 첫번째 줄에 스타일 적용
- ::first-letter : 특정 요소의 첫번째 글자에 스타일 적용
- ::before : 특정 요소의 앞에 지정한 내용을 추가
- ::after : 특정 요소의 뒤에 지정한 내용을 추가

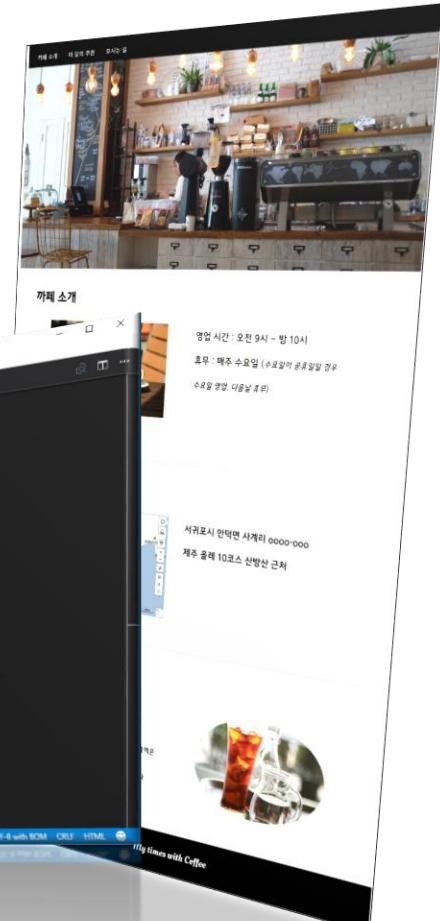
# 13. CSS3와 애니메이션

면형

## 변형과 관련된 속성들

# 트랜지션

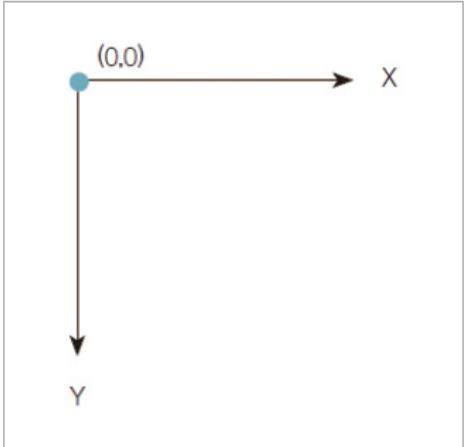
# 애니메이션



**변형(transform, 트랜스폼)** : 특정 요소의 크기나 형태 등 스타일이 바뀌는 것

### 2차원 변형

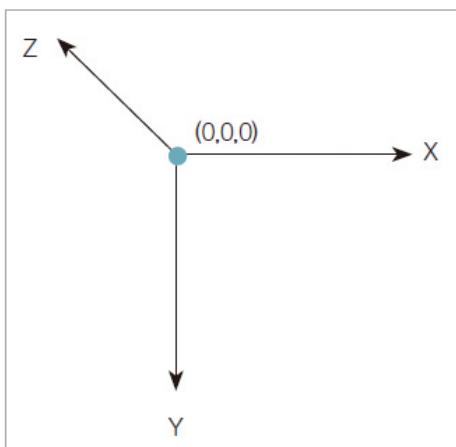
- 수평이나 수직으로 웹 요소 변형
- 크기나 각도만 지정하면 됨
- 2차원 좌표 사용



2차원 좌표계

### 3차원 변형

- x축과 y축에 원근감 추가
- z축은 앞뒤로 이동. 보는 사람 쪽으로  
    다가올 수록 값이 더 커짐



3차원 좌표계

## transform과 변형 함수

- 웹 요소를 변형하려면 transform: 다음에 변형 함수를 함께 입력함
- 구식 모던 브라우저까지 고려한다면 브라우저 접두사를 붙여야 함

### 2차원 변형 함수

변형 함수	설명
translate(tx, ty)	지정한 크기만큼 x축과 y축으로 이동합니다.
translateX(tx)	지정한 크기만큼 x축으로 이동합니다.
translateY(ty)	지정한 크기만큼 y축으로 이동합니다.
scale(sx, sy)	지정한 크기만큼 x축과 y축으로 확대/축소합니다.
scaleX(sx)	지정한 크기만큼 x축으로 확대/축소합니다.
scaleY(sy)	지정한 크기만큼 y축으로 확대/축소합니다.
rotate(각도)	지정한 각도만큼 회전합니다.
skew(ax, ay)	지정한 각도만큼 x축과 y축으로 왜곡합니다.
skewX(ax)	지정한 각도만큼 x축으로 왜곡합니다.
skewY(ay)	지정한 각도만큼 y축으로 왜곡합니다.

### 3차원 변형 함수

변형 함수	설명
matrix3d(n [, n])	4*4 행렬을 이용해 이동과 확대/축소, 회전 등의 변환을 지정합니다.
translate3d(tx, ty, tz)	지정한 크기만큼 x축과 y축, z축으로 이동합니다.
translateZ(tz)	지정한 크기만큼 z축으로 이동합니다.
scale3d(sx, sy, sz)	지정한 크기만큼 x축과 y축, z축으로 확대/축소합니다.
scaleZ(sz)	지정한 크기만큼 z축으로 확대/축소합니다.
rotate3d(rx, ry, rz, 각도)	지정한 각도만큼 회전합니다.
rotateX(각도)	지정한 각도만큼 x축으로 회전합니다.
rotateY(각도)	지정한 각도만큼 y축으로 회전합니다.
rotateZ(각도)	지정한 각도만큼 z축으로 회전합니다.
perspective(깊이)	입체적으로 보일 수 있는 깊이 값을 지정합니다.

## translate 함수

지정한 방향으로 이동할 거리를 지정하면 해당 요소를 이동시킴

기본형 `transform:translate(tx, ty)`

`transform:translate3d(tx, ty, tz)`

`transform:translateX(tx)`

`transform:translateY(ty)`

`transform:translateZ(tz)`

- **transform:translate(tx, ty)** - x축 방향으로 tx만큼, y축 방향으로 ty만큼 이동합니다. tx와 ty 두 가지 값을 사용하지만 ty 값이 주어지지 않으면 0으로 간주합니다.
- **transform:translate3d(tx, ty, tz)** - x축 방향으로 tx만큼, y축 방향으로 ty만큼, 그리고 z축 방향(앞뒤)으로 tz만큼 이동합니다.
- **transform:translateX(tx)** - x축 방향으로 tx만큼 이동합니다.
- **transform:translateY(ty)** - y축 방향으로 ty만큼 이동합니다.
- **transform:translateZ(tz)** - z축 방향으로 tz만큼 이동합니다.

```
<style>
  .movex:hover { transform: translateX(50px); }
  .movey:hover { transform: translateY(20px); }
  .movexy:hover { transform: translate(10px, 20px); }
</style>
```

```
<div class="origin">
  <div class="movex"></div>
</div>
<div class="origin">
  <div class="movey"></div>
</div>
<div class="origin">
  <div class="movexy"></div>
</div>
```



## scale 함수

지정한 크기만큼 요소를 확대/축소

기본형 `transform:scale(sx, sy)` ①

`transform:scale3d(sx, sy, sz)` ②

`transform:scaleX(sx)` ③

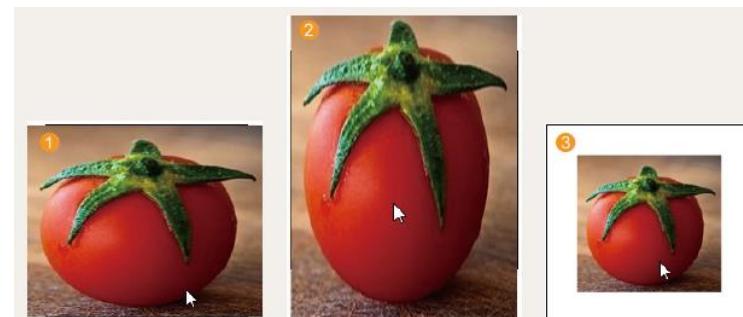
`transform:scaleY(sy)` ④

`transform:scaleZ(sz)` ⑤

- **`transform:scale(sx, sy)`** - x축 방향으로 sx만큼, y축 방향으로 sy만큼 확대합니다. sy 값이 주어지지 않는다면 sx 값과 같다고 간주합니다. 예를 들어 `scale(2.0)`는 `scale(2,2)`와 같은 함수이며 요소를 두 배로 확대합니다.
- **`transform:scale3d(sx, sy, sz)`** - x축 방향으로 sx만큼, y축 방향으로 sy만큼, 그리고 z축 방향으로 sz만큼 확대합니다.
- **`transform:scaleX(sx)`** - x축 방향으로 sx만큼 확대합니다.
- **`transform:scaleY(sy)`** - y축 방향으로 sy만큼 확대합니다.
- **`transform:scaleZ(sz)`** - z축 방향으로 sz만큼 확대합니다.

```
<style>
  .scalex:hover { transform: scalex(1.2); }
  .scaley:hover { transform: scaleY(1.5); }
  .scale:hover { transform: scale(0.7); }
</style>
```

```
<div class="origin">
  <div class="scalex">
    
  </div>
</div>
<div class="origin">
  <div class="scaley">
    
  </div>
</div>
<div class="origin">
  <div class="scale">
    
  </div>
</div>
```



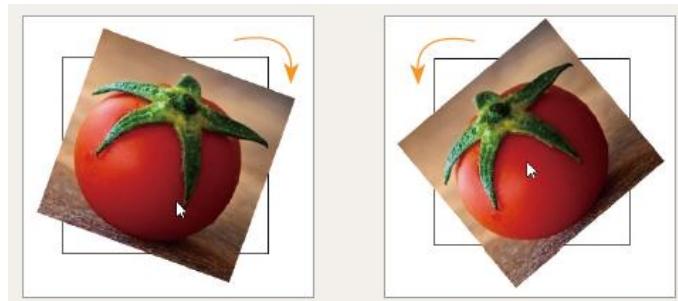
## rotate 함수

- 각도만큼 웹 요소를 시계 방향이나 시계 반대 방향으로 회전
- 일반 각도(degree)나 래디안(radian) 값 사용(1래디안=1/180°)

2차원 함수 기본형 `transform:rotate(각도)`

```
<style>
  .rotate1:hover { transform: rotate(20deg); }
  .rotate2:hover { transform: rotate(-40deg); }
</style>
```

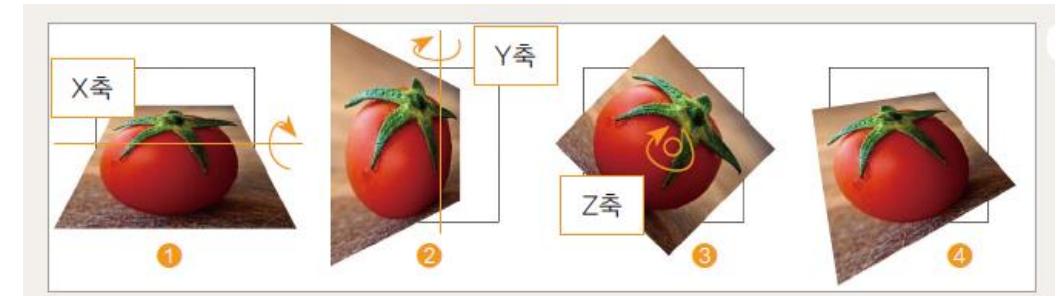
```
<div class="origin">
  <div class="rotate1"> </div>
</div>
<div class="origin">
  <div class="rotate2"> </div>
</div>
```



3차원 함수 기본형 `transform:rotate(rx, ry, 각도)`  
`transform:rotate3d(rx, ry, rz, 각도)`  
`transform:rotateX(각도)`  
`transform:rotateY(각도)`  
`transform:rotateZ(각도)`

```
<style>
  .origin { perspective: 200px; }
  .rotatex { transform: rotateX(45deg); }
  .rotatey { transform: rotateY (45deg); }
  .rotatez { transform: rotateZ(45deg); }
  .rotatexyz { transform: rotate3d(2.5, 1.2, -1.5, 45deg); }
</style>

<div class="origin">
  <div class="rotatex"></div>
</div>
<div class="origin">
  <div class="rotatey"></div>
</div>
<div class="origin">
  <div class="rotatez"></div>
</div>
<div class="origin">
  <div class="rotatexyz"></div>
</div>
```



## skew 함수

요소를 지정한 각도만큼 비틀어 왜곡

기본형    `transform:skew(ax, ay)` ①  
               `transform:skewX(ax)` ②  
               `transform:skewY(ax)` ③

- **`transform:skew(ax, ay)`** - 첫 번째 각도는 x축에서의 왜곡 각도이고 두 번째 각도는 y축에서의 왜곡 각도입니다. 두 번째 값이 주어지지 않으면 y축에 대한 왜곡 각도를 0으로 간주해 y축으로는 왜곡이 생기지 않습니다.
- **`transform:skewX(ax)`** - x축에서만 주어진 각도만큼 왜곡합니다.
- **`transform:skewY(ay)`** - y축에서만 주어진 각도만큼 왜곡합니다.

3차원 함수 기본형    `transform:rotate(rx, ry, 각도)`  
                           `transform:rotate3d(rx, ry, rz, 각도)`  
                           `transform:rotateX(각도)`  
                           `transform:rotateY(각도)`  
                           `transform:rotateZ(각도)`

```
<style>
  .skewx:hover { transform: skewX(30deg); }
  .skewy:hover { transform: skewY(15deg); }
  .skewxy:hover { transform: skew(-25deg, -15deg); }
</style>

<div class="origin">
  <div class="skewx"></div>
</div>
<div class="origin">
  <div class="skewy"></div>
</div>
<div class="origin">
  <div class="skewxy"></div>
</div>
```



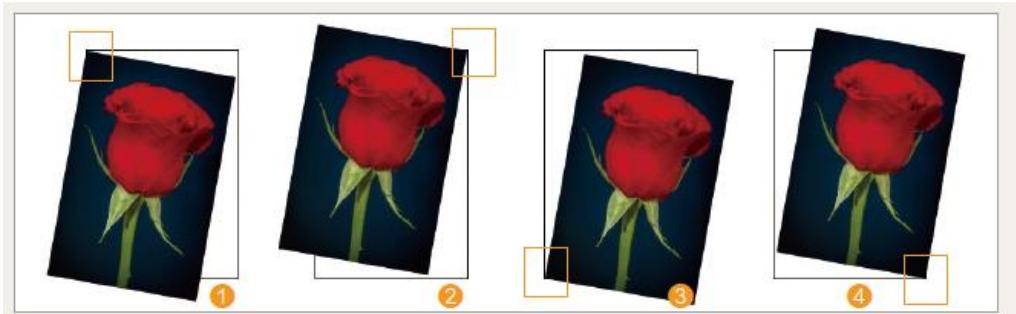
# 변형과 관련된 속성들

## transform-origin 속성

특정 지점을 변형의 기준으로 설정

기본형    transform-origin: <x축> <y축> <z축> | initial | inherit ;

속성 값	설명
<x축>	원점 기준의 x 좌푯값으로 길이 값이나 <백분율>, left, center, right 중에서 사용할 수 있습니다.
<y축>	원점 기준의 y 좌푯값으로 길이 값이나 <백분율>, top, center, bottom 중에서 선택할 수 있습니다.
<z축>	원점 기준의 z 좌푯값으로 길이 값만 사용할 수 있습니다.



```

<style>
  .rose { transform: rotateZ(10deg); }
  .ltop .rose{ transform-origin: left top; }
  .rtop .rose { transform-origin: right top; }
  .lbottom .rose { transform-origin: left bottom; }
  .rbottom .rose { transform-origin : right bottom; }
</style>

<div class="origin">
  <div class="ltop"></div>
</div>
<div class="origin">
  <div class="rtop"></div>
</div>
<div class="origin">
  <div class="lbottom"></div>
</div>
<div class="origin">
  <div class="rbottom"></div>
</div>

```

# 변형과 관련된 속성들

## **perspective 속성**

- 원근감을 갖게 함.
- 속성 값은 0보다 커야 하며 값이 클수록 사용자로부터 멀어짐.

기본형    **perspective: <크기> | none;**

속성 값	설명
<크기>	원래 위치에서 사용자가 있는 방향으로 얼마나 이동하는지를 픽셀 크기로 지정합니다.
none	perspective를 지정하지 않습니다. 기본 값입니다.★

## **perspective-origin 속성**

- 입체적으로 표현할 요소의 아랫부분(bottom) 위치 지정
- 좀 더 높은 곳에서 원근을 조절하는 듯한 느낌을 갖게 함.

기본형    **perspective-origin: <x축 값> | <y축 값>;**

속성 값	설명
<x축 값>	웹 요소가 x축에서 어디에 위치하는지를 지정합니다. 사용할 수 있는 값은 길이 값이나 백분율, left, right, center입니다. 기본 값은 50%입니다.
<y축 값>	웹 요소가 y축에서 어디에 위치하는지를 지정합니다. 사용할 수 있는 값은 길이 값이나 백분율, top, center, bottom입니다. 기본 값은 50%입니다.

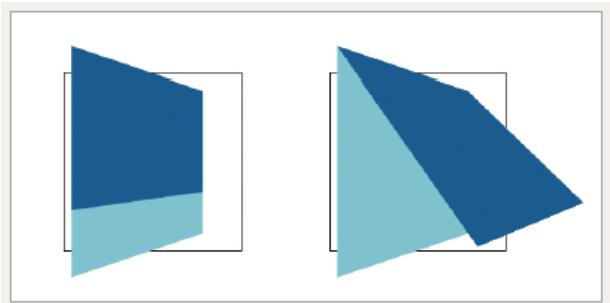
# 변형과 관련된 속성들

## transform-style 속성

부모 요소에 적용한 3D 변형을 하위 요소에도 적용

기본형    transform-style: flat | preserve-3d

속성 값	설명
flat	하위 요소를 평면으로 처리합니다.
preserve-3d	하위 요소들에 3D효과를 적용합니다.



```

<style>
  .box1 {
    background:#82cbd8;
    transform: rotateY(45deg);
  }
  .box2 {
    background: #0d6097;
    transform-origin: left top;
    transform: rotateX(45deg);
  }
  #tr-style1 { transform-style:flat ; }
  #tr-style2 { transform-style:preserve-3d; }
</style>

<div class="container">
  <div class="box1" id="tr-style1">
    <div class="box2"></div>
  </div>
</div>
<div class="container">
  <div class="box1" id="tr-style2">
    <div class="box2"></div>
  </div>
</div>

```

# 변형과 관련된 속성들

## backface-visibility 속성

요소의 뒷면, 즉 반대쪽 면을 표시할 것인지 결정

기본형 backface-visibility : visible | hidden

속성 값	설명
visible	뒷면을 표시합니다. 기본 값입니다.★
hidden	뒷면을 표시하지 않습니다.

### 두 개 이상의 변형 동시에 사용하려면

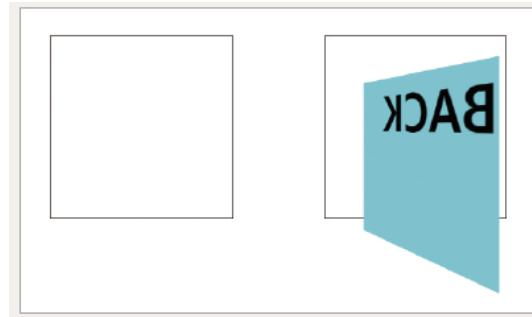
transform 속성에 변형 함수를 나열함.

예) 크기를 2배 확대하면서 x 축 기준으로 180도 회전.

`transform: scale(2) perspective(120px) rotateX(180deg);`

```
<style>
  .box {
    background:#82cbd8;
    transform: rotateY(135deg);
  }
  #back1 { backface-visibility: hidden; }
  #back2 { backface-vis : visible; }
</style>
```

```
<div class="container">
  <div class="box" id="back1"><h1>BACK</h1></div>
</div>
<div class="container">
  <div class="box" id="back2"><h1>BACK</h1></div>
</div>
```

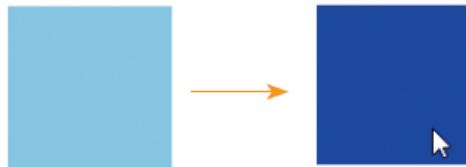


# 트랜지션

## 트랜지션이란

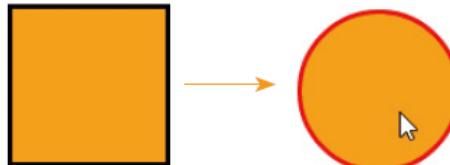
웹 요소의 스타일 속성이 조금씩 자연스럽게 바뀌는 것

예) 하늘색 도형 위로 마우스를 올려놓으면 도형이 하늘색에서 파란색으로 바뀌고 마우스를 치우면 원래 배경 색으로 되돌아감.



사각형의 배경 색이 바뀌는 트랜지션

예) 도형 위로 마우스를 올려놓으면 사각형의 테두리와 테두리색이 바뀌고 마우스를 치우면 원래 스타일로 되돌아감.



사각형의 모양과 테두리 색이 바뀌는 트랜지션

## 트랜지션의 속성

속성	설명
transition-property	트랜지션 대상을 설정합니다.
transition-duration	트랜지션 진행 시간을 설정합니다.
transition-timing-function	트랜지션 속도 곡선을 설정합니다.
transition-delay	트랜지션 지연 시간을 설정합니다.
transition	transition-property와 transition-duration, transition-timing-function, transition-delay 속성을 한꺼번에 설정합니다.

# 트랜지션

## transition-property 속성

- 트랜지션을 적용할 속성 선택
- 이 속성을 지정하지 않으면 모든 속성이 트랜지션 대상이 됨.

기본형 transition-property: all | none | <속성 이름>

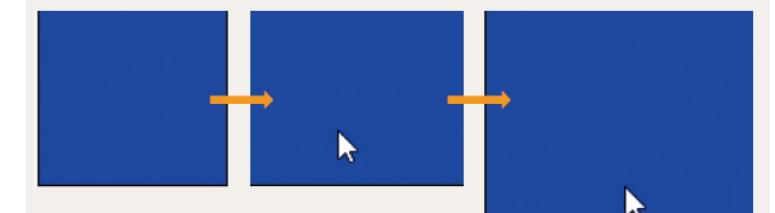
속성 값	설명
all	all 값을 사용하거나 transition-property를 생략할 경우, 요소의 모든 속성이 트랜지션 대상이 됩니다. 기본 값입니다.★
none	트랜지션 동안 아무 속성도 바뀌지 않습니다.
<속성 이름>	트랜지션 효과를 적용할 속성 이름을 지정합니다. 예를 들어 배경 색만 바꿀 것인지, width 값을 바꿀 것인지 원하는 대상만 골라 지정할 수 있습니다. 속성이 여러 개일 경우, 쉼표(,)로 구분해 나열합니다.

```
transition-property:all; /* 해당 요소의 모든 속성에 트랜지션 적용 */
transition-property:background-color; /* 해당 요소의 배경 색에 트랜지션 적용 */
transition-property:width, height; /* 해당 요소의 너비와 높이에 트랜지션 적용 */
```

## transition-duration 속성

- 트랜지션 진행 시간 지정
- 시간 단위는 초(seconds) 또는 밀리초(milliseconds)
- 트랜지션이 여러 개라면 쉼표(,)로 구분해 진행 시간 지정

```
<style>
.tr1 {
  width: 100px;
  height: 100px;
  background-color: blue;
  border: 1px solid black;
  transition-property: width, height;
  transition-duration: 2s, 1s;
}
.tr1:hover {
  width:200px;
  height:120px;
}
</style>
```



# 트랜지션

## transition-timing-function 속성

트랜지션의 시작과 중간, 끝에서의 속도 지정

```
기본형 transition-timing-function: linear | ease | ease-in | ease-out |
ease-in-out | cubic-bezier(n,n,n,n)
```

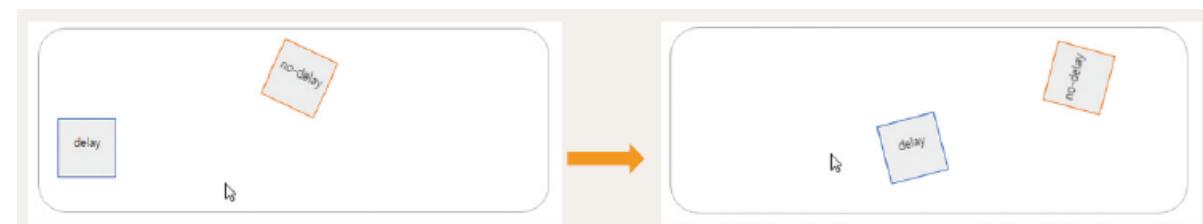
속성 값	설명
linear	시작부터 끝까지 똑같은 속도로 트랜지션을 진행합니다.
ease	처음에는 천천히 시작하고 점점 빨라지다가 마지막에는 천천히 끝냅니다. 기본 값입니다.★
ease-in	시작을 느리게 합니다.
ease-out	느리게 끝냅니다.
ease-in-out	느리게 시작하고 느리게 끝냅니다.
cubic-bezier(n,n,n,n)	베지에 함수를 직접 정의해 사용합니다. n에서 사용할 수 있는 값은 0~1입니다.

```
transition-timing-function: linear;
```

## transition-delay 속성

- 트랜지션이 언제부터 시작될지 지연 시간 지정
- 시간 단위는 초(seconds) 또는 밀리초(milliseconds). 기본값 0

```
<style>
#no-delay {
    transition-duration: 3s;
}
#delay {
    transition-duration: 3s;
    transition-delay : 1s;
}
</style>
```



# 트랜지션

## transition-timing-function 속성

트랜지션의 시작과 중간, 끝에서의 속도 지정

기본형    transition-timing-function: linear | ease | ease-in | ease-out |  
ease-in-out | cubic-bezier(n,n,n,n)

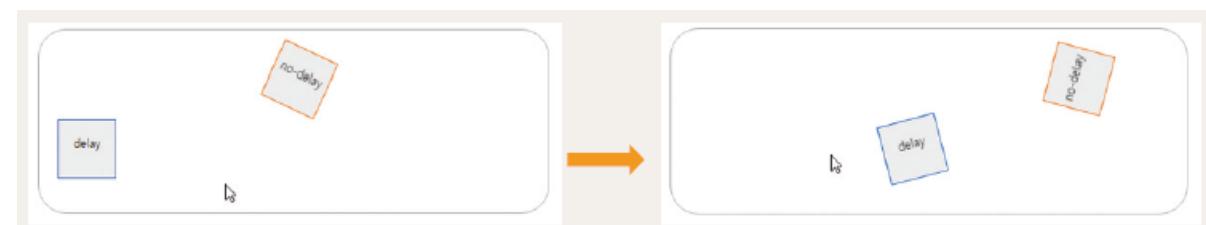
속성 값	설명
linear	시작부터 끝까지 똑같은 속도로 트랜지션을 진행합니다.
ease	처음에는 천천히 시작하고 점점 빨라지다가 마지막에는 천천히 끝냅니다. 기본 값입니다.★
ease-in	시작을 느리게 합니다.
ease-out	느리게 끝냅니다.
ease-in-out	느리게 시작하고 느리게 끝냅니다.
cubic-bezier(n,n,n,n)	베지에 함수를 직접 정의해 사용합니다. n에서 사용할 수 있는 값은 0~1입니다.

transition-timing-function: linear;

## transition-delay 속성

- 트랜지션이 언제부터 시작될지 지연 시간 지정
- 시간 단위는 초(seconds) 또는 밀리초(milliseconds). 기본값 0

```
<style>
#no-delay {
  transition-duration: 3s;
}
#delay {
  transition-duration: 3s;
  transition-delay : 1s;
}
</style>
```



# 애니메이션

## CSS와 애니메이션

- 웹 요소에 애니메이션 추가
- 애니메이션을 시작해 끝내는 동안 원하는 곳 어디서든 스타일을 바꾸며 애니메이션을 정의할 수 있다.
- 키프레임(keyframe) : 애니메이션 중간에 스타일이 바뀌는 지점

인터넷 익스플로러 10 이상과 최신 모던 브라우저에서 지원하며 이전 모던 브라우저를 고려하려면 -webkit-, -moz- 접두사를 붙여야 한다.

## CSS 애니메이션에서 사용하는 주요 속성

속성	설명
@keyframes	애니메이션이 바뀌는 지점을 설정합니다.
animation-delay	애니메이션 지연 시간을 지정합니다.
animation-direction	애니메이션 종료 후 처음부터 시작할지, 역방향으로 진행할지를 지정합니다.
animation-duration	애니메이션 실행 시간을 설정합니다.
animation-fill-mode	애니메이션이 종료되었거나 지연되어 애니메이션이 실행되지 않는 상태일 때 요소의 스타일을 지정합니다.
animation-iteration-count	애니메이션 반복 횟수를 지정합니다.
animation-name	@keyframes로 설정해 놓은 중간 상태의 이름을 지정합니다.
animation-play-state	애니메이션을 멈추거나 다시 시작합니다.
animation-timing-function	애니메이션의 속도 곡선을 지정합니다.
animation	animation 하위 속성들을 한꺼번에 묶어 지정합니다.

# 트랜지션

## @keyframes 속성

- 애니메이션의 시작과 끝을 비롯해 상태가 바뀌는 지점을 설정
- '이름'으로 애니메이션 구별

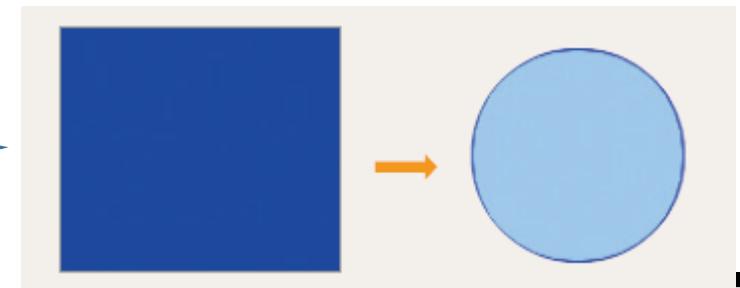
```
기본형 @keyframes <이름> {
    <선택자> { <스타일> }
}
```

- 시작 위치는 0%, 끝 위치 100%로 놓고 위치 지정
- 시작과 끝 위치만 사용한다면 from, to 키워드 사용 가능
- @-webkit-keyframes나 @-moz-keyframes처럼 브라우저 접두사를 붙여함

```
<style>
div {
    width: 100px;
    height: 100px;
    background-color: blue;
    animation-name: change-bg;
    animation-duration: 3s;
}

@keychange change-bg {
    from {
        background-color: blue;
        border: 1px solid black;
    }
    to {
        background-color: #a5d6ff;
        border: 1px solid blue;
        border-radius: 50%;
    }
}
</style>
```

시작할 때 파란색 사각형이었다가 끝날 때  
옅은 파란색 원으로 바뀌는 애니메이션



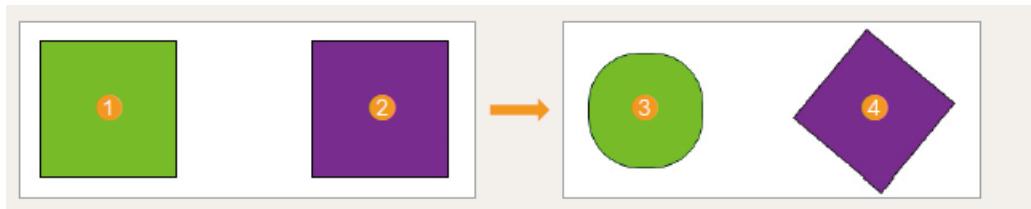
# 트랜지션

## animation-name 속성

@keyframes 속성에서 만든 애니메이션 이름을 사용

@keyframes를 이용해 shape와 rotate라는 애니메이션 정의

#box1에는 animation-name:shape를,  
#box2에는 animation-name:rotate를 사용해 애니메이션 실행



```

<style>
#box1 {
    background-color: #4cff00;
    border: 1px solid black;
    animation-name: shape;
    animation-duration: 3s;
}
#box2 {
    background-color: #8f06b0;
    border: 1px solid black;
    animation-name: rotate;
    animation-duration: 3s;
}
@keyframes shape {
    from {
        border: 1px solid black;
    }
    to {
        border: 1px solid black;
        border-radius: 50%;
    }
}
@keyframes rotate {
    from {
        transform: rotate(0deg);
    }
    to {
        transform: rotate(45deg);
    }
}
</style>

```

```

<div id="box1"></div>
<div id="box2"></div>

```

# 트랜지션

## animation-duration 속성

- 애니메이션 실행 시간 설정. 기본값 0
- 사용 가능한 값은 초(s)나 밀리초(ms)

기본형 `animation-duration: <시간>`

## animation-direction 속성

애니메이션이 끝난 후 원래 위치로 돌아가거나 반대 방향으로 애니메이션 실행하도록 지정

기본형 `animation-direction: normal | alternate`

속성 값	설명
normal	애니메이션을 끝까지 실행하면 원래 있던 위치로 돌아갑니다. 기본 값입니다.★
alternate	애니메이션을 끝까지 실행하면 왔던 방향으로 되돌아가면서 애니메이션을 실행합니다.

## animation-iteration-count 속성

애니메이션 반복 횟수 지정하기

기본형 `animation-iteration-count: <숫자> | infinite`

속성 값	설명
<숫자>	입력한 숫자만큼 반복합니다. 기본 값은 1입니다.
infinite	무한 반복합니다.

## animation-timing-function 속성

애니메이션 속도 곡선 지정

기본형 `animation-timing-function: linear | ease | ease-in | ease-out | ease-in-out | cubic-bezier(<n>,<n>,<n>,<n>)`

## animation 속성

- 여러 개의 애니메이션 속성을 하나의 속성으로 줄여서 사용
- 지정하지 않은 속성은 기본 값 사용. 하지만 animation-duration 속성 값은 반드시 지정해야 함.

```
기본형   animation: <animation-name> | <animation-duration> |
           <animation-timing-function> | <animation-delay> |
           <animation-iteration-count> | <animation-direction>
```

```
.box {
  animation-name: moving;
  animation-duration: 3s;
  animation-timing-function: ease-in;
  animation-direction: alternate;
  animation-iteration-count: infinite;
}
```

→ .box {  
 animation: moving 3s alternate  
 infinite ease-in;  
}

```
<style>
.box {
  width: 60px;
  height: 60px;
  margin: 60px;
  animation: rotate 1.5s infinite, background 1.5s infinite
  alternate;
}
@keyframes rotate {
  from { transform: perspective(120px) rotateX(0deg)
  rotateY(0deg); }
  50% { transform: perspective(120px) rotateX(-180deg)
  rotateY(0deg); }
  to { transform: perspective(120px) rotateX(-180deg)
  rotateY(-180deg); }
}
@keyframes background {
  from { background: red; }
  50% { background-color: green; }
  to { background-color: blue; }
}
</style>

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



# 14. 반응형 웹이란

모바일 기기와 웹 디자인

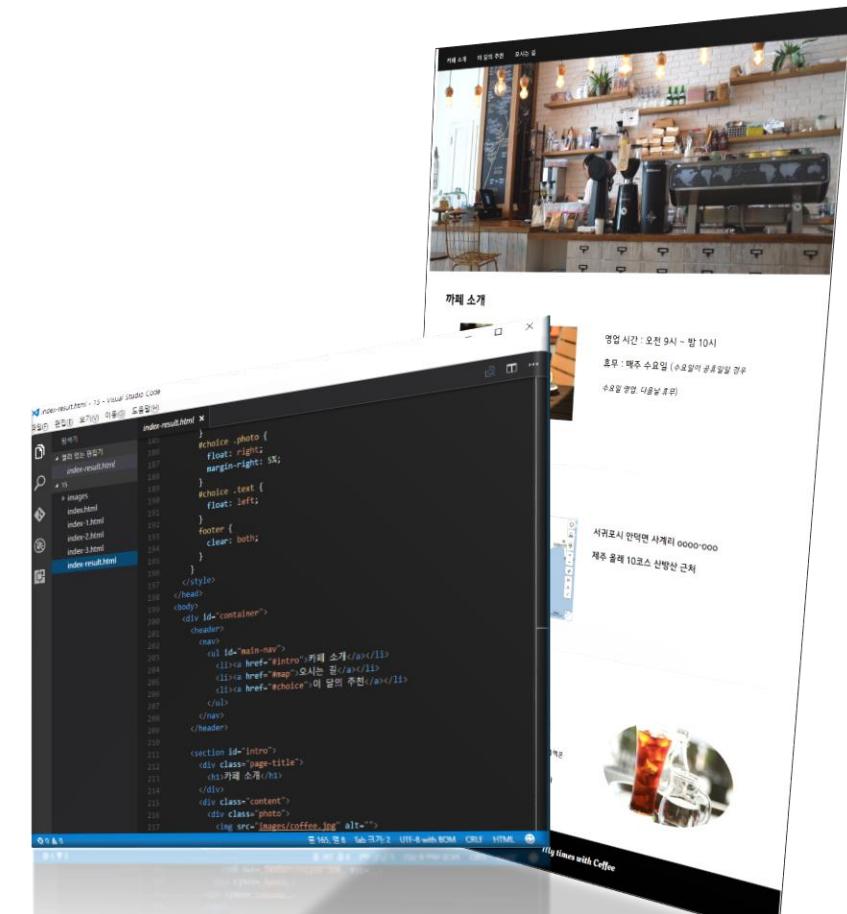
가변 그리드 레이아웃

가변 요소

미디어 쿼리

미디어 쿼리를 이용한 사이트 구성

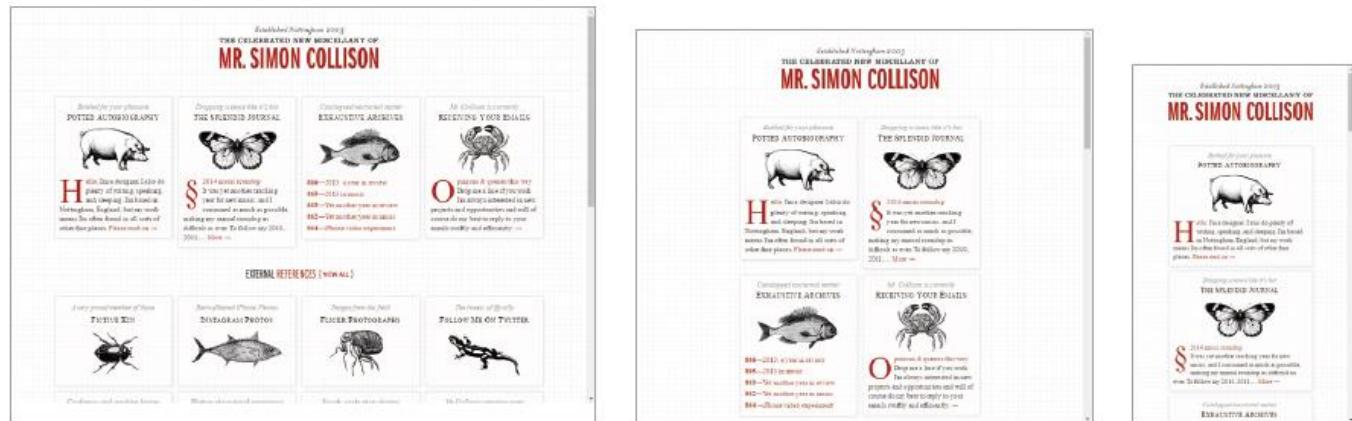
플렉서블 박스 레이아웃



# 모바일 기기와 웹 디자인

## 반응형 웹 디자인

- 웹 사이트의 내용을 그대로 유지하면서 다양한 화면 크기에 맞게 웹 사이트를 표시하는 방법
- 다양한 화면 크기의 모바일 기기들이 계속 쏟아져 나오는데 그 때마다 그 크기에 맞춘 사이트를 별도로 제작하는 것은 비효율적  
→ 화면 크기에 '반응'해 화면 요소들을 자동으로 바꾸어 사이트를 구현하는 것이 바로 반응형 웹 디자인



반응형 웹 사이트(<http://colly.com>)

# 모바일 기기와 웹 디자인

## 반응형 웹 디자인의 장단점

### 장점

- 모든 스마트 기기에서 접속 가능
- 가로 모드에 맞춘 레이아웃 변경 가능
- 사이트 유지, 관리 용이

### 단점

반응형 웹 기술이 최신 웹 표준인 CSS3의 일부

→ 최신 모던 웹 브라우저에서만 지원됨

# 모바일 기기와 웹 디자인

## 뷰포트(viewport)

- 뷰포트 : 실제 내용이 표시되는 영역
- PC 화면과 모바일 화면의 픽셀 표시 방법이 다르기 때문에 모바일 화면에서 의도한대로 표시되지 않음  
→ 뷰포트를 지정하면 기기 화면에 맞춰 확대/축소해서 내용 표시

## 뷰포트 지정하기

- <head> 태그 안에서 <meta> 태그를 이용해 뷰포트 지정

```
기본형 <meta name="viewport" content="속성1=값, 속성2=값2, ... ">
```

- 일반적인 사용법 : 뷰포트의 너비를 스마트폰 화면 너비에 맞추고 초기 화면 배율을 1로 지정

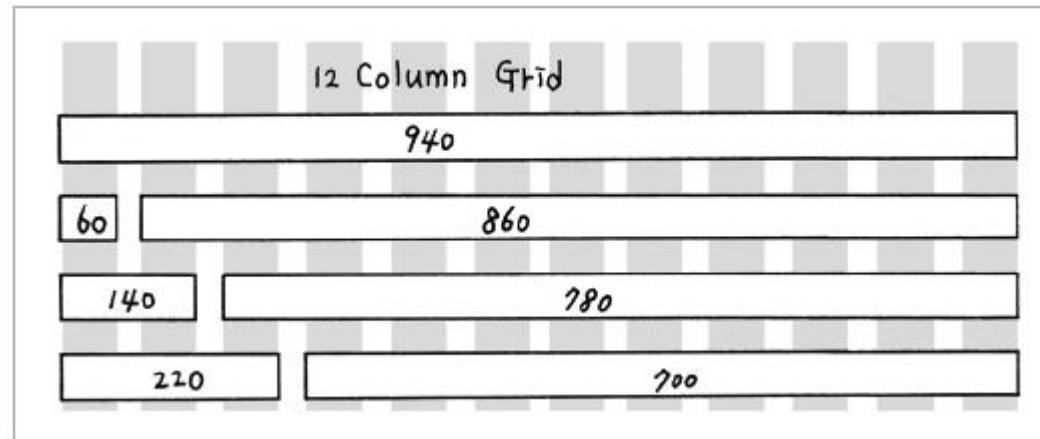
```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

속성	설명	사용 가능한 값	기본 값
width	뷰포트 너비	device-width 또는 크기	브라우저 기본 값
height	뷰포트 높이	device-height 또는 크기	브라우저 기본 값
user-scalable	확대/축소 가능 여부	yes 또는 no	yes
initial-scale	초기 확대/축소 값	1~10	1
minimum-scale	최소 확대/축소 값	0~10	0.25
maximum-scale	최대 확대/축소 값	0~10	1.6

# 가변 그리드 레이아웃

## 그리드 시스템(grid system)

- 화면을 여러 개의 칼럼(column)으로 나누어, 필요할 때마다 칼럼들을 묶어 배치하는 방법
- 화면 너비 값에 따라 '960 그리드 시스템', '1200 그리드 시스템' 등으로 나뉨
- 칼럼 개수에 따라 '12 칼럼 그리드 시스템', '16 칼럼 그리드 시스템', '24 칼럼 그리드 시스템' 등으로 나뉨
- 주로 960 픽셀 12 칼럼의 그리드 시스템 사용
- 고정 그리드 : 화면 너비를 일정하게 고정하고 레이아웃 만듦.
- 가변 그리드 : 화면 너비를 % 같은 가변 값으로 지정.
- 가변 그리드 레이아웃을 사용할 경우, 너비 값이 줄어들면 실제 콘텐츠를 확인하기 불편하므로 가능하면 간결한 디자인을 사용하는 것이 좋음.

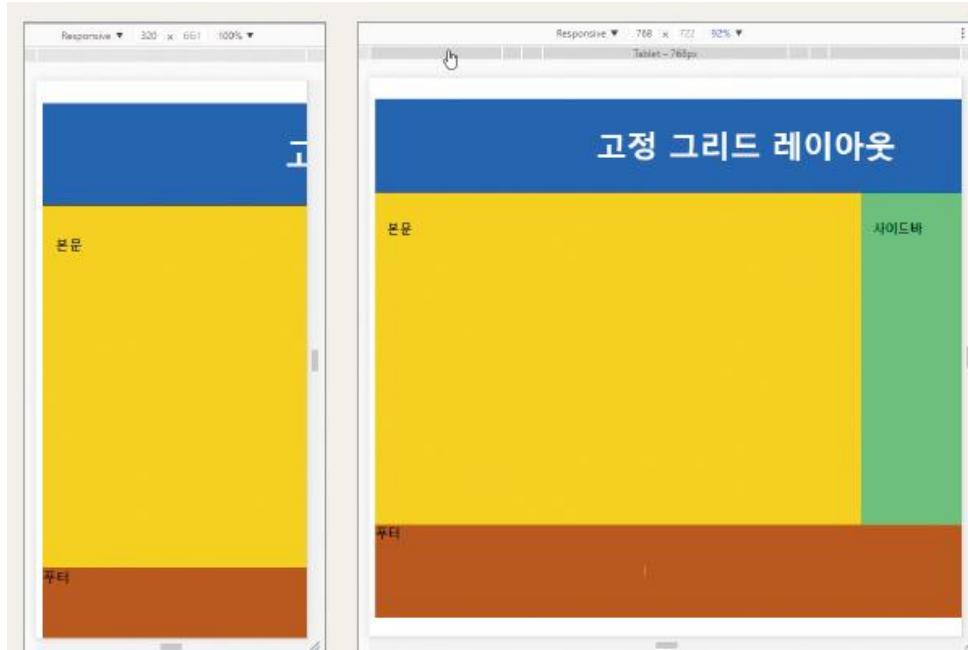


문서 좌우에 10px씩의 패딩이 있다고 가정한 레이아웃 →

# 가변 그리드 레이아웃

## 고정 그리드 레이아웃일 경우

- 문서의 맨 바깥 부분을 #wrapper 요소로 묶고 너비를 960px로 지정
- 헤더와 본문, 사이드 바, 푸터를 배치. 이 때 너비는 px 값.
- 화면 너비가 좁아질 경우 내용의 일부가 가려질 수 있음.



```

<style>
#wrapper {
  width:960px;
  margin:0 auto;
}
header {
  width:960px;
  height:120px;
}
.content {
  float:left;
  width:600px;
}
.right-side {
  float:right;
  width:300px;
}
footer {
  clear:both;
  width:960px;
}
</style>

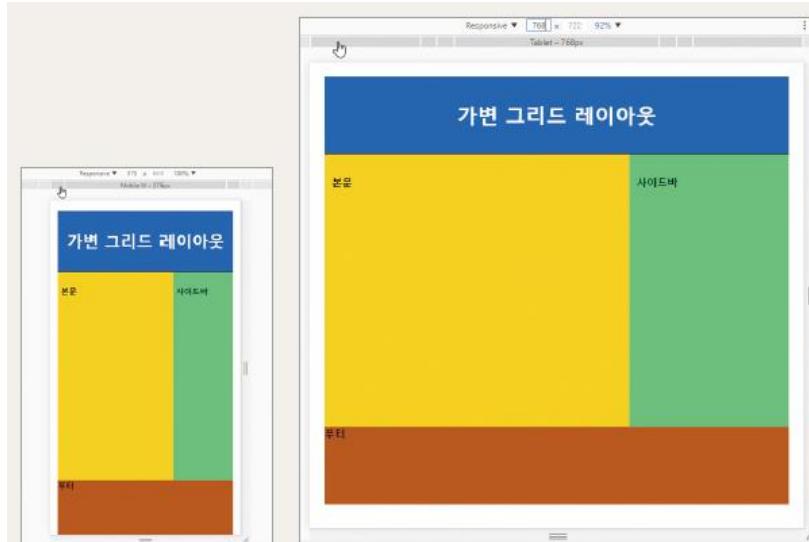
```

# 가변 그리드 레이아웃

## 가변 그리드 레이아웃 만들기

- ① 전체를 감싸는 요소의 너비를 %로 변환 (화면에 꽉 차게 하고 싶다면 100%, 여유를 두려면 적당히)
- ② 전체를 감싸는 요소의 너비를 기준으로 각 요소의 너비를 계산

(요소의 너비 / 콘텐츠 전체를 감싸는 요소의 너비) \* 100



```

<style>
#wrapper {
  width:96%;
  margin:0 auto;
}
header {
  width:100%;
  height:120px;
}
.content {
  float:left;
  width:62.5%;
  height:400px;
  padding:1.5625%;
  background-color:#ffd800;
}
.right-side {
  float:right;
  width:31.25%;
  height:400px;
  padding:1.5625%;
  background-color:#00ff90;
}
footer {
  clear:both;
  width:100%;
  height:120px;
  background-color:#c3590a;
}
</style>

```

## 가변 글꼴

### em 단위

부모 요소 폰트의 대문자 M 너비를 1em으로 지정. 1em=16px

$$\text{글자 크기(em)} = \frac{\text{글자 크기(px)}}{16\text{px}}$$

```
<style>
  .header-text{ font-size:2em; }
  .content { font-size: 1.5em; }
  .right-side { font-size: 1.5em; }
  footer { font-size: 1.5em; }
</style>
```

### rem 단위

- em 단위는 부모 요소가 중첩될 경우 글자 크기가 계속 달라짐
- rem은 처음부터 기본 크기를 지정하고 그것을 기준으로 글자 크기 지정

```
<style>
  body { font-size:16px; }
  .header-text{ font-size:2rem; }
  .fluid-text { font-size:1.5rem; }
</style>
```

## 가변 이미지

- 브라우저 창의 너비가 변하더라도 이미지 너비 값은 변하지 않음 → 브라우저 화면 너비를 줄일 경우 이미지 일부가 가려짐
- 가변 이미지(fluid image)로 만들면 창의 너비에 따라 이미지 너비도 조절됨

### 1) CSS를 이용한 방법

이미지를 감싸고 있는 부모 요소만큼만 커지거나 작아지도록 max-width 속성 값을 100%로 지정

```
<style>
  .content img {
    max-width:100%;
    height:auto;
  }
</style>
```



## 가변 이미지

### 2) <img> 태그와 srcset 속성

화면 너비 값이나 픽셀 밀도에 따라 고해상도의 이미지  
파일 지정 가능

기본형    `<img src=<이미지> srcset=<이미지1>, <이미지2>, <이미지3>, ...>`

예)

```

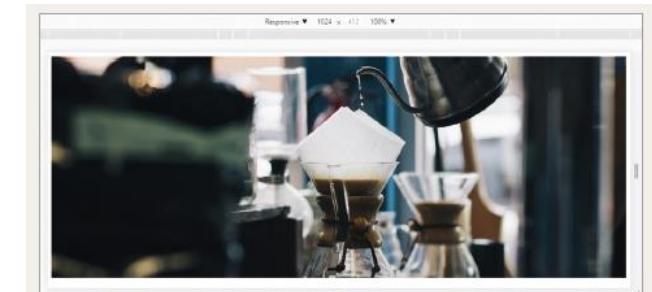
```

```
<picture>
  <source srcset="images/shop-large.jpg" media="(min-width:1024px)">
  <source srcset="images/shop-medium.jpg" media="(min-width:768px)">
  <source srcset="images/shop-small.jpg" media="(min-width:320px)">
  
</picture>
```

### 3) <picture> 태그와 <source> 태그

화면 해상도뿐만 아니라 화면 너비에 따라 다른 이미지 파일 표시

속성	설명
srcset	이미지 파일의 경로
media	srcset에 지정한 이미지를 표시하기 위한 조건(속성 값은 14-4 미디어 쿼리 참고).
type	파일 유형
sizes	파일의 크기



## 가변 비디오

CSS를 사용해 max-width 속성을 100%로 지정

```
<style>
  video { max-width: 100%; }
</style>

<video autoplay loop src="assets/cars.mp4"> </video>
```



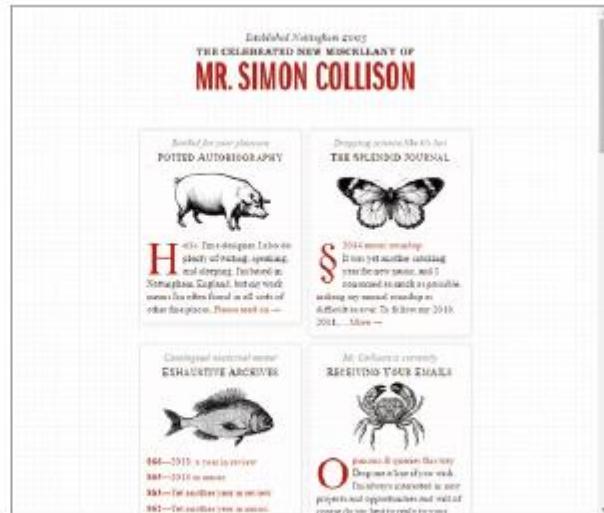
# 미디어 쿼리

접속하는 장치(미디어)에 따라 특정한 CSS 스타일을 사용하도록 함.

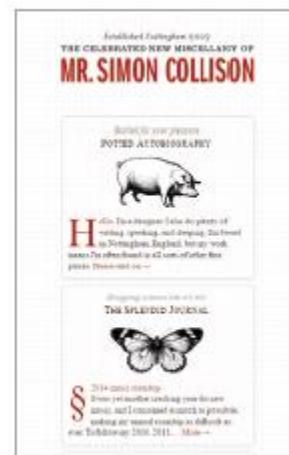
예) colly.com

브라우저 창의 너비를 조절할 때마다 화면에 표시되는 칼럼 개수가 달라짐

PC나 태블릿, 스마트폰의 웹 브라우저 화면 크기에 따라 사이트 레이아웃이 바뀜.



미디어 쿼리를 이용해 제작된 사이트들을 모아놓은  
<http://mediaqueri.es> 참고



colly.com

# 미디어 쿼리

## 미디어 쿼리 구문

기본형 @media [only | not] 미디어 유형 [and 조건] \* [and 조건]

연산자	설명
and	조건을 계속 추가할 수 있습니다.
,(쉼표)	동일한 스타일 유형을 사용할 미디어의 유형과 조건이 있다면 쉼표를 이용해 추가합니다.
only	미디어 쿼리를 지원하는 웹 브라우저에서만 조건을 인식하게 합니다. 이 키워드를 사용하면 미디어 쿼리를 지원하지 않는 웹 브라우저에서는 미디어 쿼리를 무시하고 실행하지 않습니다. IE에서는 미디어 쿼리를 제대로 인식하지 못하기 때문에 only 키워드를 사용하더라도 큰 의미가 없습니다.
not	not 다음에 지정하는 미디어 유형을 제외합니다. 예를 들어 'not tv' 라고 지정한다면 TV를 제외한 미디어 유형에만 적용합니다.

## 미디어 유형의 종류

미디어 유형	사용 가능한 미디어
all	모든 미디어 유형
print	인쇄 장치
screen	컴퓨터 스크린(스마트폰 스크린 포함)
tv	음성과 영상이 동시 출력되는 TV
aural	음성 합성 장치(주로 화면을 읽어 소리로 출력해 주는 장치)
braille	점자 표시 장치
handheld	패드(pad)처럼 손에 들고 다니는 장치
projection	프로젝터
tty	디스플레이 기능이 제한된 장치(픽셀(px) 단위를 사용할 수 없음)
embossed	점자 프린터

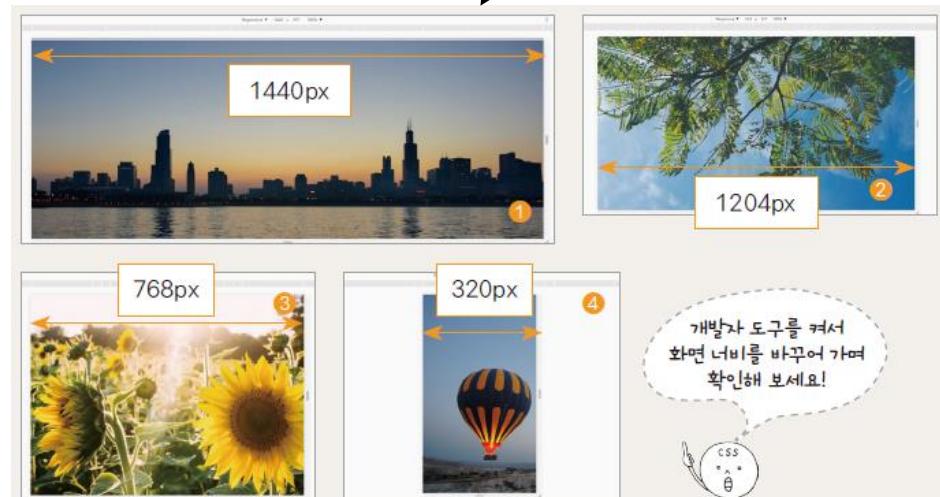
예) 미디어 유형이 'screen'이면서 최소 너비가 '200px'이고 최대 너비가 '360px'일 경우에 적용할 CSS를 정의하는 구문

```
@media screen and (min-width:200px) and (max-width:360px) {
    ...
}
```

## 미디어 쿼리 조건

### 웹 문서의 가로 너비와 세로 높이(뷰포트)

가로, 세로 값 설정하는 속성	설명
width, height	웹 페이지의 가로 너비, 세로 높이
min-width, min-height	최소 너비, 최소 높이
max-width, max-height	최대 너비, 최대 높이



```

<style>
body {
    background: url(images/bg0.jpg) no-repeat
    fixed;
    background-size: cover;
}
@media screen and (max-width:1024px) {
    body {
        background: url(images/bg1.jpg) no-
repeat fixed;
        background-size: cover;
    }
}
@media screen and (max-width:768px) {
    body {
        background: url(images/bg2.jpg) no-
repeat fixed;
        background-size:cover;
    }
}
@media screen and ( m -w :320px) {
    body {
        background: url(images/bg3.jpg) no-
repeat fixed;
        background-size: cover;
    }
}
</style>

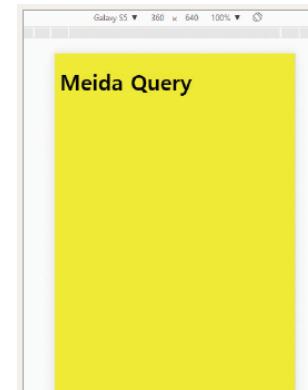
```

## 미디어 쿼리 조건

### 단말기의 가로 너비와 세로 높이

단말기의 가로, 세로 값을 설정하는 속성	설명
device-width, device-height	단말기의 가로 너비, 세로 높이
min-device-width, min-device-height	단말기의 최소 너비, 최소 높이
max-device-width, max-device-height	단말기의 최대 너비, 최대 높이

단말기 크기와 뷰포트 크기를 하나로 통일해 사용하기 위해 뷰포트를 지정할 때 width="device-width"로 놓고 사용



### 화면 회전

orientation 속성을 사용해서 화면 방향 체크

속성	설명
orientation: portrait	단말기 세로 방향
orientation: landscape	단말기 가로 방향

```
<style>
@media screen and
(orientation:landscape) {
    body {background-color:orange;}
}
@media screen and
(orientation :portrait) {
    body {background-color:yellow;}
}
</style>
```

## 미디어 쿼리 조건

### 화면 비율

단말기 브라우저 화면의 너비 값을 높이 값으로 나눈 것

속성	설명
aspect-ratio	화면 비율(width 값/ height 값)
min-aspect-ratio	최소 화면 비율
max-aspect-ratio	최대 화면 비율

### 단말기 화면 비율

단말기의 너비 값을 높이 값으로 나눈 것

속성	설명
device-aspect-ratio	단말기 화면 비율(width 값/ height 값)
min-device-aspect-ratio	단말기 최소 화면 비율
max-device-aspect-ratio	단말기 최대 화면 비율

### 색상당 비트 수

단말기의 최대 색상 비트 수를 조건으로 사용

color:1 →  $2^1 = 2$

color:2 →  $2^2 = 4$

color:3 →  $2^3 = 8$

속성 값	설명
color	비트 수
min-color	최소 비트 수
max-color	최대 비트 수

## 미디어 쿼리 중단점 만들기

- 중단점(breakpoint) : 서로 다른 CSS를 적용할 화면 크기
- 대부분 기기의 화면 크기 기준.
- 모든 기기를 반영할 수 없기 때문에 스마트폰과 태블릿, 데스크톱 정도로 구분
- 모바일 퍼스트(mobile first) : 모바일 기기 레이아웃을 기본으로 작성 → 태블릿 & PC 레이아웃 작성
- 미디어 쿼리 중단점은 개발자나 작업 조건에 따라 달라질 수 있다.

```
/* 스마트폰 세로 */  
@media only screen and (min-width: 320px) { ..... }  
  
/* 스마트폰 가로 */  
@media only screen and (min-width: 480px) { ..... }  
  
/* 태블릿 세로 */  
@media only screen and (min-width: 768px) { ..... }  
  
/* 태블릿 가로 / 데스크톱 */  
@media only screen and (min-width: 1024px) { ..... }
```

# 미디어 쿼리 사용한 사이트 구성

## 외부 CSS 파일 연결

특정 조건에 맞을 경우, 지정한 css 파일을 가져와 적용

### <link> 태그 사용하기

<head>와 </head> 사이에 삽입

기본형

```
<link rel="stylesheet" media="미디어 쿼리 조건" href="css 파일 경로">
```

예) 화면 너비가 768px 이하일 때 미리 만들어 놓은  
태블릿용 스타일 시트 파일을 적용하려면

```
<link rel="stylesheet" media="screen and (max-width:768px)"  
      href="css/tablet.css">
```

### @import 구문 사용하기

<style>와 </style> 사이에 삽입

기본형

@import url(css 파일 경로) 미디어 쿼리 조건

예) 화면 너비가 321px 이상 768px 이하일 때 미리 만들어  
놓은 태블릿용 스타일 시트 파일을 적용하려면

```
@import url("css/tablet.css") only screen and (min-  
width:321px) and (maxwidth:768px);
```

# 미디어 쿼리 사용한 사이트 구성

## 웹 문서에서 직접 정의하기

### <style> 태그 안에 media 속성 사용

```
기본형 <style media="미디어 쿼리 조건">
          스타일 규칙들
      </style>
```

예) 최대 너비가 320px일 때 즉 너비가 320px 이하인 경우

```
<style media="screen and (max-width:320px)">
  body {
    background-color: orange;
  }
</style>
```

### <style> 태그 안에 @media 구문 사용하기

```
기본형 <style>
          @media 미디어 쿼리 조건{
            스타일 규칙들
          }
      </style>
```

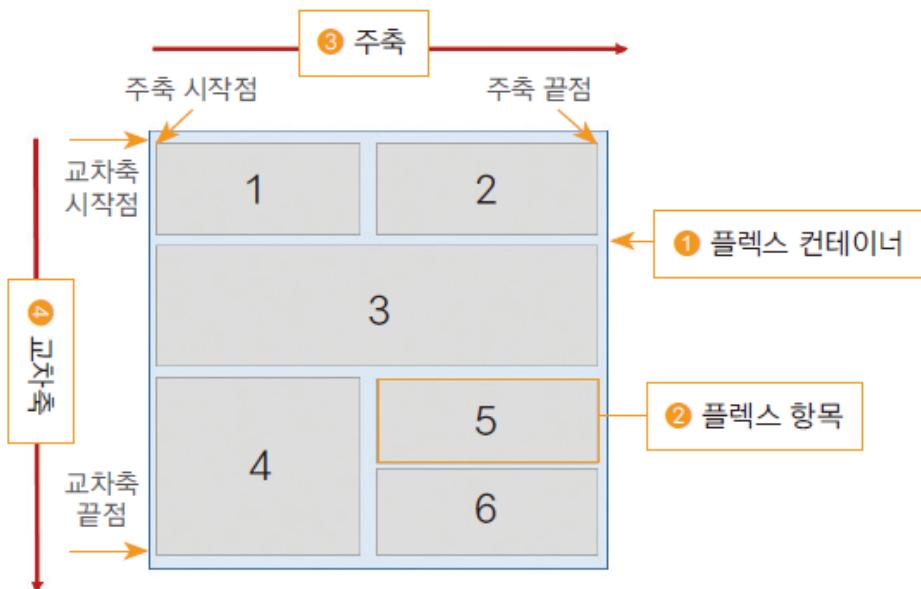
예) 화면 너비가 320px 이하일 때 배경 색을 주황색으로 바꾸는 미디어 쿼리

```
<style>
  @media screen and (max-width:320px) {
    body {
      background-color: orange;
    }
  }
</style>
```

# 플렉서블 박스 레이아웃

## 플렉서블 박스 레이아웃(flexible box layout)

- 그리드 레이아웃을 기본으로, 플렉스 박스를 원하는 위치에 배치하는 것.
- 여유 공간에 따라 너비나 높이, 위치를 자유롭게 변형할 수 있음



① **플렉스 컨테이너(flex container)** - 웹 문서에 텍스트나 이미지, 표 등 웹 요소들을 플렉서블하게 사용하려면 먼저 플렉스 컨테이너로 묶어 주어야 합니다.

② **플렉스 항목(flex item)** - 플렉스 컨테이너에 담기는 웹 요소입니다. 위의 그림에서 1~6번 요소들은 모두 플렉스 항목입니다.

③ **주축(main axis)** - 플렉스 컨테이너 안에서 플렉스 항목을 배치하는 기본 방향입니다. 기본적으로 주축은 왼쪽에서 오른쪽으로 수평 방향으로 배치합니다. 주축에서 플렉스 항목이 배치되기 시작하는 지점을 '주축 시작점', 배치가 끝나는 지점을 '주축 끝점'이라고 합니다.

④ **교차축(cross axis)** - 교차축은 주축과 교차되는 방향입니다. 기본적으로 위에서 아래로 배치합니다. 교차축에서 배치가 시작되는 지점을 '교차축 시작점', 배치가 끝나는 지점을 '교차축 끝점'이라고 합니다.

# 플렉서블 박스 레이아웃

## 플렉서블 박스 레이아웃 기본 속성

### display 속성

배치 요소들을 감싸는 부모 요소를 플렉스 컨테이너로 지정

기본형    `display: flex | inline-flex`

속성 값	설명
<code>flex</code>	플렉스 박스를 박스 레벨 요소로 정의합니다.
<code>inline-flex</code>	플렉스 박스를 인라인 레벨 요소로 정의합니다.

```
<style>
#container {
  display:flex;
}
</style>
```

```
<div id="container">
  <div></div>
  <div></div>
</div>
```

### display 속성과 브라우저 접두사

- 최신 모던 브라우저에서는 모두 지원되고 대부분의 구식 버전에서도 지원됨
- 하지만**, 브라우저마다 플렉스 박스를 지원하는 방법이 달라 브라우저 접두사를 붙여야 함

```
.wrapper {
  display:-webkit-box; /* iOS 6 이하, 사파리 3.1 */
  display:-moz-box; /* 파이어폭스 19 이하 */
  display:-ms-flexbox; /* IE 10 */
  display:-webkit-flex; /* 웹킷 구 버전 */
  display: flex; /* 표준 스펙 */
}
```

# 플렉서블 박스 레이아웃

## 플렉서블 박스 레이아웃 기본 속성

### flex-direction 속성

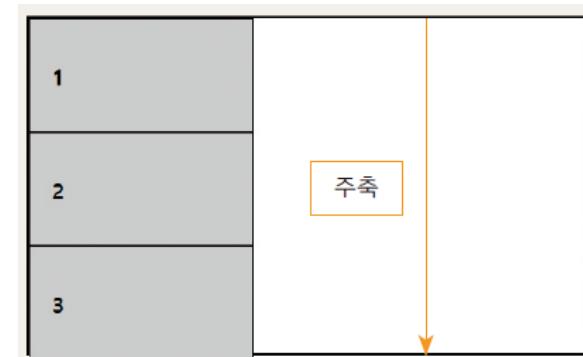
플렉스 항목 배치 방향 지정

기본형    `flex-direction: row | row-inverse | column | column-inverse`

속성 값	설명
row	주축을 가로로 교차축을 세로로 지정합니다. 플렉스 항목은 주축 시작점에서 끝점으로(왼쪽에서 오른쪽으로) 배치됩니다.★
row-inverse	주축을 가로로 교차축을 세로로 지정합니다. 플렉스 항목은 주축 끝점에서 시작점으로(오른쪽에서 왼쪽으로) 배치됩니다.
column	주축을 세로로 교차축을 가로로 지정합니다. 플렉스 항목은 주축 시작점에서 끝점으로(위쪽에서 아래쪽으로) 배치됩니다.
column-inverse	주축을 세로로 교차축을 가로로 지정합니다. 플렉스 항목은 주축 끝점에서 시작점으로(아래쪽에서 위쪽으로) 배치됩니다.

```
<style>
#container {
  display:flex;
  flex-direction: column;
}
#container div {
  width: 200px;
  border: 1px solid black;
  background: #ccc;
}
</style>
```

```
<div id="container">
  <div id="box1"><h2>1</h2></div>
  <div id="box2"><h2>2</h2></div>
  <div id="box3"><h2>3</h2></div>
</div>
```



# 플렉서블 박스 레이아웃

## 플렉서블 박스 레이아웃 기본 속성

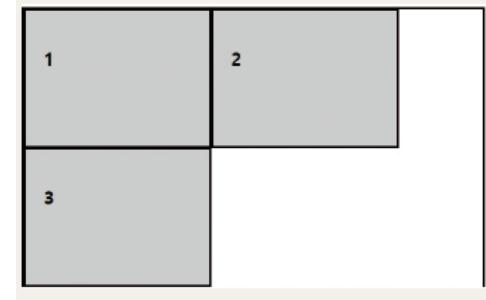
### flex-wrap 속성

플렉스 항목을 한 줄 또는 여러 줄로 배치

기본형    flex-wrap: no-wrap | wrap | wrap-reverse

속성 값	설명
no-wrap	플렉스 항목들을 한 줄에 표시합니다. 기본 값입니다.★
wrap	플렉스 항목을 여러 줄에 표시합니다.
wrap-reverse	플렉스 항목을 여러 줄에 표시하되 기준 방향과 반대로 배치합니다.

```
<style>
#container {
  display:flex;
  flex-wrap:wrap;
}
</style>
```



```
<style>
#container {
  display:flex;
  flex-wrap:wrap-reverse;
}
</style>
```



# 플렉서블 박스 레이아웃

## 플렉서블 박스 레이아웃 기본 속성

### flex-flow 속성

- 플렉스 배치 방향과 여러 줄 배치를 한꺼번에 지정
- 기본 값은 flex-flow:row no-wrap

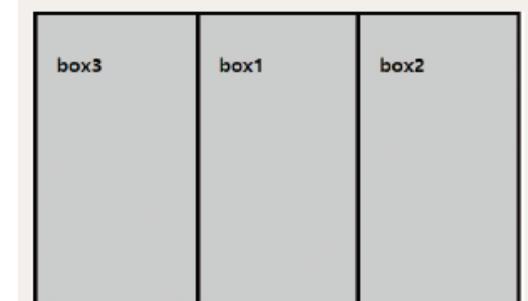
기본형    flex-flow : <플렉스 방향> <플렉스 줄 배치>

### order 속성

- 플렉스 항목의 배치 순서 바꾸기
- order 값에 지정된 순서에 따라 배치됨.

```
<style>
#container {
  display:flex;
}
#box1 { order:2; }
#box2 { order:3; }
#box3 { order:1; }
</style>
```

```
<div id="container">
  <div id="box1">
    <h2>box1</h2>
  </div>
  <div id="box2">
    <h2>box2</h2>
  </div>
  <div id="box3">
    <h2>box3</h2>
  </div>
</div>
```



# 플렉서블 박스 레이아웃

## 플렉서블 박스 레이아웃 기본 속성

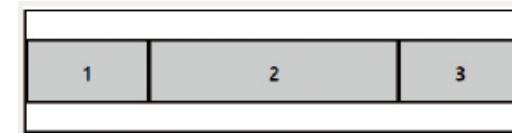
### flex 속성

- 플렉스항목의 크기 조절
- flex-grow와 flex-shrink, flex-basis 속성은 별개의 속성이지만 따로 쓰지 않고 flex 속성으로 묶어 사용.

```
<style>
  #box1 {
    flex:1 1 0;
  }
  #box2 {
    flex:2 2 0; /* 2배 늘이거나 2배 줄임 */
  }
</style>
```

기본형    flex : [<flex-grow> <flex-shrink> <flex-basis>] | auto | initial

속성 값	설명
<flex-grow>	플렉스 항목의 너비를 얼마나 늘일지 숫자로 지정합니다.
<flex-shrink>	플렉스 항목의 너비를 얼마나 줄일지 숫자로 지정합니다.
<flex-basis>	플렉스 항목의 기본 크기를 지정합니다. width 속성처럼 너비 값을 지정할 수도 있고 0이나 auto 를 지정할 수도 있습니다. 0일 경우, flex-grow와 flex-shrink의 인수 값을 함께 사용하고 auto일 경우, 플렉스 항목의 너비 값을 사용합니다.
initial	항목의 width/height 값에 의해 크기가 결정되는데 플렉스 컨테이너의 공간이 부족할 경우, 최소 크기까지 줄입니다.★
auto	항목의 width/height 값에 의해 크기가 결정되지만 플렉스 컨테이너의 공간에 따라 늘이거나 줄입니다.



# 플렉서블 박스 레이아웃

## 플렉스 항목 배치를 위한 속성

### justify-content 속성

플렉스 항목을 주축 방향으로 배치할 때의 배치 기준

기본형 justify-content : flex-start | flex-end | center | space-between | space-around

속성 값	설명	미리 보기
flex-start	주축의 시작점을 기준으로 배치합니다.	
flex-end	주축의 끝점을 기준으로 배치합니다.	
center	주축의 중앙을 기준으로 배치합니다.	
space-between	첫 번째 플렉스 항목과 마지막 플렉스 항목은 시작점과 끝점에 배치한 후 중앙 항목들은 같은 간격으로 배치합니다.	
space-around	모든 플렉스 항목들을 같은 간격으로 배치합니다.	

### align-items 속성, align-self 속성

교차축을 기준으로 하는 배치 방법 조절

기본형 align-items : stretch | flex-start | flex-end | center | baseline

속성 값	설명	미리 보기
stretch	플렉스 항목을 확장해 교차축을 꽉 채웁니다. 기본 값입니다.★	
flex-start	교차축의 시작점을 기준으로 배치합니다.	
flex-end	교차축의 끝점을 기준으로 배치합니다.	
center	교차축의 중앙을 기준으로 배치합니다.	
baseline	시작점과 글자 기준선이 가장 먼 플렉스 항목(미리보기에서는 2 번 항목의 글자 크기가 가장 크기 때문에 2번의 글자 기준선이 가장 멀리 떨어져 있음)을 시작점에 배치합니다. 그리고 그 글자의 기준선과 다른 항목의 기준선을 맞추어 배치합니다.	



# 플렉서블 박스 레이아웃

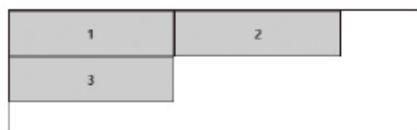
## 플렉스 항목 배치를 위한 속성

### align-content 속성

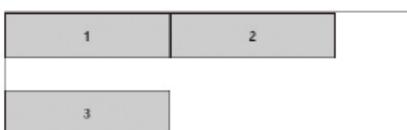
플렉스 항목이 여러 줄로 표시될 때 교차 축 기준의 배치 방법 지정

기본형 align-content : flex-start | flex-end | center | space-between | space-around

align-content : flex-start (기본 값)



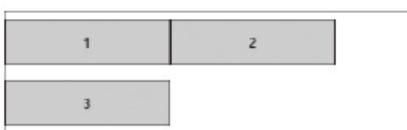
align-content : space-between



align-content : flex-end



align-content : space-around



align-content : center

