

HTML5 & CSS3 실전대비를 위한 웹표준

기초편

```
color: inherit;
text-decoration: none;
}
.seminar {width: 900px; margin: 0 auto; border: solid 1px #ccc; position: relative;
.seminar .tit {text-align: center; padding: 10px 0 20px; font-weight: bold;
.seminar .tit_comment { text-align: center; color: #999; margin-bottom: 20px;
.seminar .links { margin-bottom: 20px; }
.seminar .links:after { content: ""; display: block; clear: both; }
.seminar .links a { display: block; width: 25%; float: left; background-color: #eee; border: 1px solid #ccc; padding: 5px; text-align: center; margin-bottom: 5px; }
}
</style>
</head>
<body>
<section class="seminar">
<h2></h2>
<div class="links">
<a href="#">연세대학교</a>
<a href="#">고려대학교</a>
<a href="#">한양대학교</a>
<a href="#">연세대학교</a>
</div>
<ul class="list">
<li>

<ul>
<li>47th Annual Meeting International Neuropsychological Society</li>
<li>06 : New York City, USA</li>
<li>702 : 2020.01.01~2020.01.09</li>
</ul>
<a href="#">세종대학교</a>
</li>
<li>

<ul>
<li>47th Annual Meeting International Neuropsychological Society</li>
<li>06 : New York City, USA</li>
<li>702 : 2020.01.01~2020.01.09</li>
</ul>
<a href="#">세종대학교</a>
</li>
<li>

<ul>
<li>47th Annual Meeting International Neuropsychological Society</li>
<li>06 : New York City, USA</li>
<li>702 : 2020.01.01~2020.01.09</li>
</ul>
<a href="#">세종대학교</a>
</li>
</ul>
```

| | |
|------------------------------|-----------|
| 1장. HTML..... | 3 |
| 가. 웹의 표준화..... | 3 |
| 나. HTML | 5 |
| 다. HTML 기본구조 및 요소..... | 6 |
| 라. 시맨틱 웹(Semantic Web) | 23 |
| 마. HTML5 추가된 시맨틱 요소 | 24 |
| 2장 CSS..... | 25 |
| 가. CSS란?..... | 25 |
| 나. CSS 작성 방법 | 25 |
| 다. CSS 기본 구문 | 27 |
| 라. 선택자 우선순위..... | 34 |
| 마. 스타일 속성 살펴보기 | 35 |
| 3장. 활용문제 | 61 |
| 가. Html 마크업 작성하기 | 61 |
| 나. CSS로 스타일 적용하기. | 63 |

1장. HTML

가. 웹의 표준화

브라우저의 혼란

90대 모자이크, 넷스케이프, 익스플로러와 같은 브라우저 제조사는 웹 표준화 기구에 속하지 않은 채 폰트, 컬러, 테이블, 프레임과 같은 HTML 확장 기술을 했다. 이들은 10년 동안 표준화 기구 하나 없이 자바스크립트, DOM과 같은 기술을 발표해왔으며, 이 기간에 웹 개발자들은 혼란에 빠졌다.

이러한 혼돈 속에서 World Wide Web 창시자인 '팀 버너스리'의 주도 아래 웹과 관련 있는 기업, 협회 중심으로 W3C(World Wide Web Consortium)가 설립되었다.

웹 표준화의 탄생

W3C, ISO, IETF, Ecma International과 같은 표준화 기구는 브라우저 업체 간 분쟁을 조정하기 위해 웹의 근간이 되는 HTML, CSS 등 공동 웹 기술에 대한 표준을 정의하였다.

하지만, 웹 표준에 강제성이 없었기에 제대로 지켜지지않자 WaSP(The Web Standards Project)에서 웹 표준안을 활용하면서 이 프로젝트에 참여한 웹 개발자들이 웹 표준안을 활발하게 사용하고 전파하여 공식 표준화 기구, 일반 개발자들로 구성된 협회, 창조적인 개인 웹 개발자에 의해 현재 웹 표준안이 탄생되었다.

웹 표준화의 현재

현재 웹 표준화는 다음과 같은 기술로 이루어져있다.

- W3C의 HTML/XHTML
- W3C의 CSS (Cascading Style Sheets)
- W3C의 The DOM (Document Object Model)
- Ecma International의 자바스크립트 (standard EcmaScript라 부르기도 한다.)
- W3C의 PNG와 SVG, ISO의 JPEG 같은 이미지 파일 포맷
- W3C의 WAI-ARIA와 같은 접근성 향상 기술

하지만, 표준안에 기반을 둔 웹 개발이 위 기술만 사용하지는 않는다.

웹 표준화를 지켜야하는 이유

웹을 개발하고 콘텐츠를 제공할 때 도구 제한이 없어야하며, 원한다면 언제든지 도구를 교체할 수 있어야한다. 또한 특정 브라우저 제조사나 기술에 종속적이지 않아야 하며, 다른 브라우저를 사용할 수 있어야한다.

웹 표준화는 콘텐츠 공급자, 일반 사용자, 도구 개발자, 브라우저 개발자, 웹 개발자 모두에게 도움이 된다.

프런트엔드 구조

웹사이트의 프런트엔드에는 콘텐츠, 디자인, 기능 구조로 되어있다.

웹 콘텐츠에 의미를 부여하고 구조를 형성하는 것은 **마크업 언어(HTML)**가 담당하고, 콘텐츠 디자인은 **CSS**가 담당하며, 모든 프런트엔드의 브라우저 상호작용은 **자바스크립트**가 담당한다.

나. HTML

HTML이란?

하이퍼텍스트 마크업 언어(HyperText Markup Language)라는 의미로 웹 페이지를 사용되고있는 마크업 언어이다. HTML은 제목, 단락, 목록 등과 같은 본문을 위한 구조적 의미를 나타내는 것뿐만 아니라 링크, 인용과 그 밖의 항목으로 구조적문서를 만들 수 있는 방법을 제공한다. 그리고 이미지와 객체를 내장하고 대화형 양식을 생성하는데 사용될 수 있다.

HTML은 웹 페이지 콘텐츠 안의 <>에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한다.

요소 살펴보기

요소(Element)는 빈요소와 비어있지 않은 요소가 있다.

비어있지 않은 요소는 요소안에 다른 요소(또는 텍스트)를 담을 수 있다.

반대로 빈요소는 다른 요소를 담을 수 없다.

비어있지 않은 요소 예시

```
<p> p요소안에 텍스트나 <strong>또다른 요소</strong>을 담을 수 있다.</p>
```

빈 요소 예시

```

```

요소구성

```
<p> p요소안에 텍스트나 <strong>또다른 요소</strong>을 담을 수 있다.</p>
```



```

```



다. HTML 기본구조 및 요소

문서타입

HTML 파일의 시작 부분에 HTML 문서 타입을 선언하는데 그것을 **DOCTYPE** 선언이라 한다.

DOCTYP 선언은 문서가 사용하고 있는 마크업 언어가 무엇이니, 어떤 버전을 사용하고 있는지, 페이지를 출력할 때 구문 규칙을 얼마나 엄격히 준수할지에 대한 정보를 브라우저에 넘겨준다.

HTML5 이전 버전에는 위와 같은 내용을 넣었지만 HTML5에서는 `<!DOCTYPE html>` 으로 선언하면 된다.

최상위 요소 html

html 시작을 알리는 요소이다. html 요소는 문서의 root 요소라고도 한다. 문서 내의 모든 요소는 html 요소 안에 있어야 한다.

html은 head 요소와 body요소만 담을 수 있다. (html 자식 요소로는 head 요소와 body요소만 올 수 있다.)

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <title>문서 타이틀</title>
  <meta charset="utf-8">
</head>
<body>
  body 내용
</body>
</html>
```

HTML5 기본 구조

Head 요소

html 문서에서 필수이며, **문서 정보**를 담는 요소이다. title 요소는 브라우저의 타이틀 바에 출력되는 정보이다. meta 요소는 검색엔진을 위한 해당 문서의 검색 정보를 담는다. 그 외 인코딩 방식이나 내장스크립트와 css 정보, 스크립트와 css 파일 링크 정보를 담게 된다.

body 요소

body요소 안의 내용은 브라우저 화면에 출력되는 부분이다.

그 말은 웹페이지에서 보이는 부분은 body 요소안에 작성되어야 한다. 지금부터 배우게 될 요소들은 body 요소 안에 담겨야 한다.

웹페이지의 기본 구성은 글자 이미지 링크이다. 기본 구성에서 태그를 입히면 그 글자는 의미가 없는 글자가 아닌 의미를 갖게 되는 글자가 또는 이미지가 되는 것이다. 그리고 태그에 스타일을 입혀 디자인을 하게 된다.

그럼 기본 구성인 이미지, 링크를 알아보자.

이미지

웹페이지에 보여지는 이미지들은 포토샵이나 일러스트에서 만들어진 이미지를 img 태그를 통해서 불러온다.

```

```

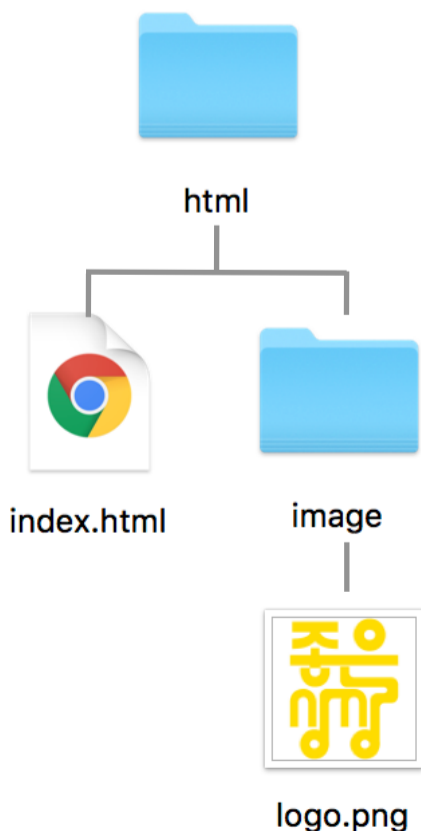


src 속성 : 저장되어있는 이미지의 위치 주소
(이미지 주소위치를 찾는 방법은 상대주소와 절대주소방법이있다.)

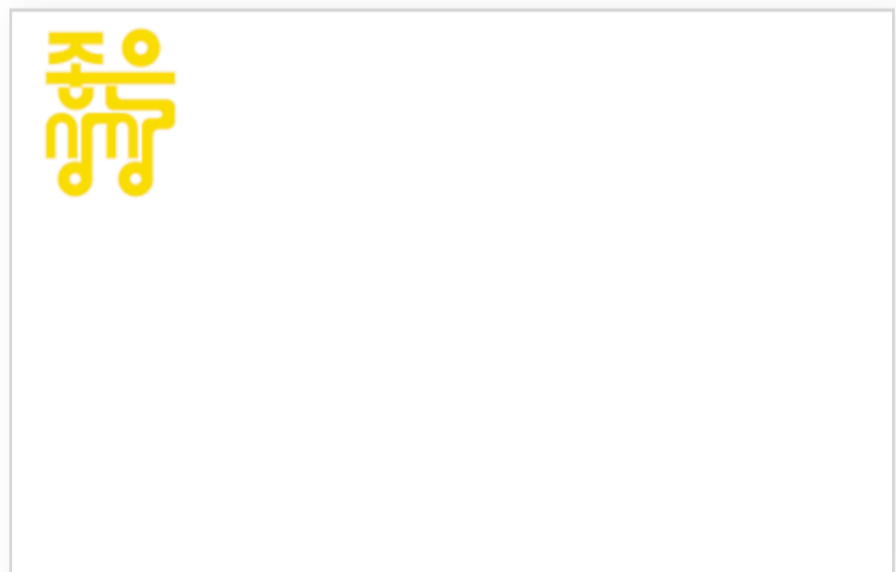
alt 속성 : 이미지 설명 (이미지 속 텍스트나 이미지의 설명을 기입한다.)

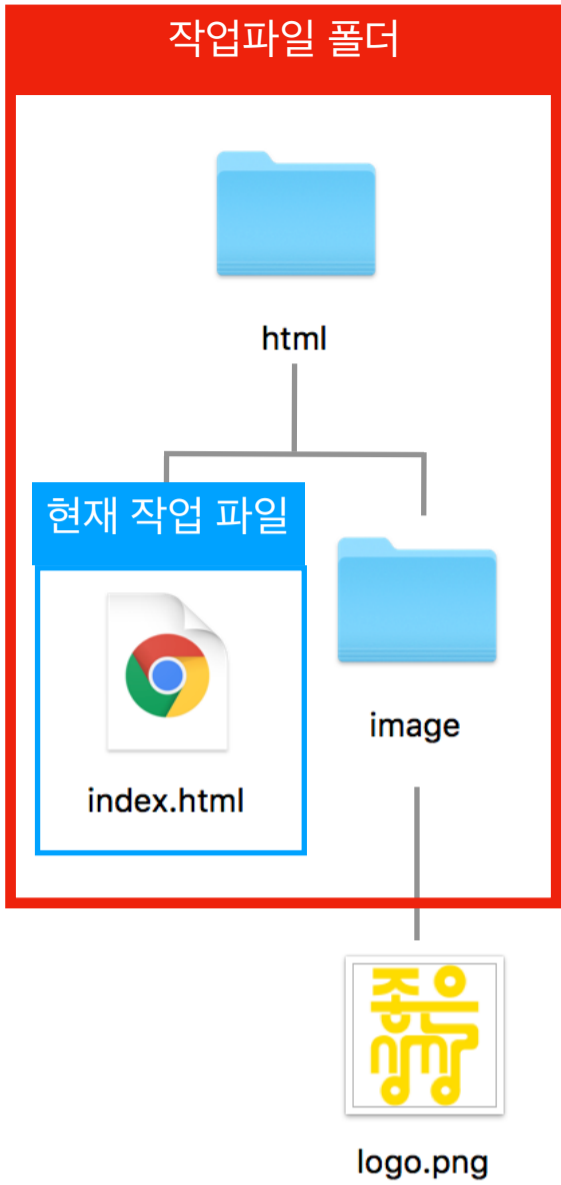
logo.png 이미지 불러오기

[파일 구조]



[브라우저]





상대주소로 불러오기

현재 파일위치를 기준으로 이미지의 위치를 찾는다.

현재 작업 파일(index.html)은 html 폴더에 있으므로 같은 폴더 안에 있는 image **폴더를 접근**할 때는 “**폴더명 /**”로 통해 폴더로 이동한다. 그럼 index.html 기준으로 봤을 때 logo.png 파일은 다음과 같다.

`images/logo.png`

[현재 파일 : index.html]

```

<!DOCTYPE html>
<html>
<head>
  <title>이미지 불러오기</title>
  <meta charset="utf-8">
</head>
<body>
  
</body>
</html>
  
```



index.html 파일은 web 폴더에 있으므로 image 폴더로 접근하기 위해서는 html 폴더 이동한 다음 image 폴더에 접근할 수 있다. web 폴더의 **상위폴더**인 html 폴더로 이동할 때는 “**../**”를 넣어 상위 폴더로 이동한다.

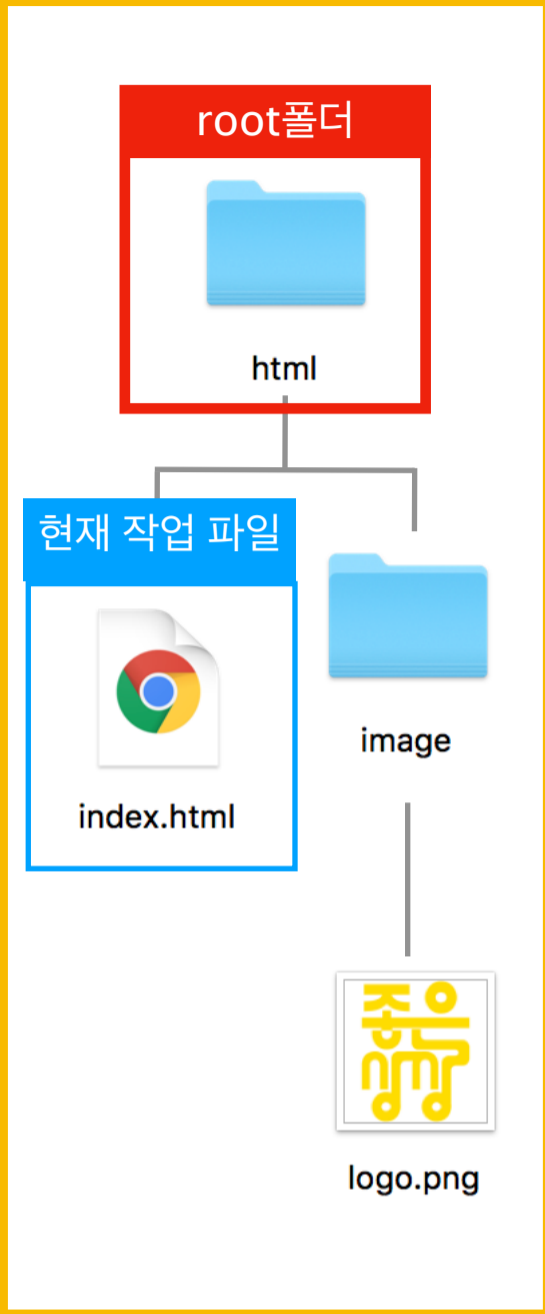
여기서 logo.png 파일 주소는 다음과 같다.

`../images/logo.png`

```

<!DOCTYPE html>
<html>
<head>
  <title>이미지 불러오기</title>
  <meta charset="utf-8">
</head>
<body>
  
</body>
</html>
  
```


서버(root폴더 : html)
도메인 : http://web.co.kr



절대주소로 불러오기

절대주소는 서버에 파일을 올렸을 때 사용한다.
파일의 최초 위치부터 파일을 불러온다.

상대주소는 작업 파일 위치를 기준으로 이미지 파일을 불러오는 거라면 절대주소는 최초의 위치(root 폴더)부터 파일을 불러온다. 여기서 root 폴더는 html 폴더이며 root 폴더는 브라우저에 도메인을 입력하고 들어갔을 때 처음으로 들어가는 폴더를 말한다.

다시 말해 브라우저에 http://web.co.kr를 입력해서 서버로 이동하면 html 폴더가 root 폴더이기 때문에 html 폴더로 이동한다는 것이다.

이때 루트폴더를 “/“ 같이 표현한다.

절대주소로 logo.png 파일을 불러올 경우 주소는 다음과 같다.

`/images/logo.png`

```
<!DOCTYPE html>
<html>
<head>
  <title>이미지 불러오기</title>
  <meta charset="utf-8">
</head>
<body>
  
  
</body>
</html>
```

링크 <a>..

웹사이트를 사용하면 이미지나 글자들을 클릭하여 다른 페이지로 이동하게 된다. 이때 웹사이트에서 기본적으로 사용되는 링크는 <a>요소로 표현할 수 있다.

<a>요소 사이에 글자를 넣어 링크를 걸거나 요소를 넣어 이미지에 링크를 걸 수 있다.

```
<a href="링크 파일주소">링크명</a>
<a href="링크 파일주소"></a>
```

링크태그에 `target`속성으로 현재 창에서 파일을 열수도 있지만 새창(탭)으로도 파일을 열 수 있다.

[현재 창에서 파일열기]

```
<a href="/index.html">메인</a>
```

```
<a href="admin/login.html" target="_self">로그인</a>
```

[새 창(탭)에서 파일열기]

```
<a href="http://webweb.co.kr" target="_blank">웹사이트</a>
```

`href` 속성 : 링크 걸 파일의 위치 주소를 넣어준다.(넣는 방법은 이미지 파일과 동일하다.)

`target` 속성 : 링크 페이지가 열리게되는 창을 지정할 수 있다.(_self [기본값], _blank)

제목요소 <h1>..</h1> ~ <h6>..</h6>

웹문서 안에 제목이 될 수 있는 글자나 이미지를 감싸면 제목부분이라는 의미를 갖게된다.

6단계로 제목단계를 표현 할 수 있다. 가장 큰 제목은 <h1>요소이며 내려 갈 수록 다음단계의 제목을 의미한다.

실무에서 <h1>요소는 웹사이트의 가장 큰제목인 로고로 사용된다.

[html]

```
<!DOCTYPE html>
<html>
<head>
  <title>제목요소</title>
  <meta charset="utf-8">
</head>
<body>
  <h1></h1>
  <h2>제목</h2>
  <h3>소제목</h3>
  <h4>소소제목</h4>
  <h5>소소제목</h5>
  <h6>소소소제목</h6>
</body>
</html>
```

[브라우저]



제목

소제목

소소제목

소소제목

소소소제목

브라우저 화면 결과를 보면 낮은 단계로 갈수록 글자의 사이즈가 변하는 것을 확인할 수 있다. 이걸 브라우저 내에 요소마다 다른 스타일을 주고 있기 때문에 다음과 같은 결과가 나온 것이다. 브라우저 내에는 요소마다 기본적으로 스타일을 주고 있다. 하지만, 스타일은 내가 직접 변경할 수 있기 때문에 브라우저에서 지정된 스타일은 중요하지 않다.

✧ html요소를 공부할때는 요소의 의미를 잘 파악하셔야지 스타일은 중요한것이 아닙니다.

문단(단락)요소 <p>..</p>

단락은 유사한 주제를 갖는 **문자열의 묶음**이다.

문서를 보시면 단락을 구분을 하게되는데 이 단락을의미할때 <p> 요소를 사용한다.

```
<h1>아홉살 인생</h1>
```

```
<p>
```

```
    <아홉살 인생>은 위기철씨의 소설로, 1991년 출간된 이후 만화와 영화로도 만들어졌다.
```

```
</p>
```

```
<p>
```

```
    이 소설은 1960년대 경상도 산동네 마을을 배경으로 아홉 살 꼬마의 동심어린 눈을 통해  
    가난하고 소외된 이들의 고단한 일상을 정겹고 따뜻하게 그린 것으로, 1991년 출간되었으며  
    2003년 MBC '느낌표!' 선정 도서가 되어 상반기 화제의 베스트셀러에 오르기도 했다.
```

```
</p>
```

줄바꿈

문자열을 아래로 이동시키고 싶을때
요소로 줄바꿈을 해줄 수 있다.

[html 문서]

```
<h1>하여가</h1>
<p>
  이런들 어떠하며 저런들 어떠하리
  만수산 드령참이 얹어진들 어떠하리
  우리도 이같이 얹어져 백년까지 누리리라.
</p>
```

[브라우저]

하여가

이런들 어떠하며 저런들 어떠하리 만수산 드령참이 얹어진들 어떠하리 우리도 이같이 얹어져 백년까지 누리리라.

html 문서에서는 엔테로 줄바꿈을 했지만 브라우저에서는 줄바꿈을 표현하지 않는다. 브라우저에서 줄바꿈을 표현하고 싶으면 다음과 같이
요소를 넣어줘야 한다.

[html 문서]

```
<h1>하여가</h1>
<p>
  이런들 어떠하며 저런들 어떠하리<br>
  만수산 드령참이 얹어진들 어떠하리<br>
  우리도 이같이 얹어져 백년까지 누리리라.
</p>
```

[브라우저]

하여가

이런들 어떠하며 저런들 어떠하리
만수산 드령참이 얹어진들 어떠하리
우리도 이같이 얹어져 백년까지 누리리라.

비순서 목록 요소 ..

목록을 살펴보면 연관된 항목들이 묶여있는 형태를 목록이라 한다. 이때 항목들의 순서가 중요하지 않는 목록일 경우 로 표현하고 목록 안에 묶이게되는 항목들을 요소로 표현한다.

[html 문서]

```
<h1>브라우저 종류</h1>
<ul>
  <li>크롬</li>
  <li>사파리</li>
  <li>익스플로러</li>
  <li>파이어폭스</li>
</ul>
```

[브라우저]

브라우저 종류

- 크롬
- 사파리
- 익스플로러
- 파이어폭스

* 목록요소는 요소안에 요소만 자식요소로 들어갈 수 있다.

순서 목록 요소 ..

항목들의 순서가 중요시되는 목록은 로 표현하고 목록 안에 묶이게되는 항목들을 요소로 표현한다.

[html 문서]

```
<h1>실시간 뉴스</h1>
<ol>
  <li>새해인사</li>
  <li>경자년</li>
  <li>빅뱅이론</li>
  <li>스핀</li>
</ol>
```

[브라우저]

실시간 뉴스

1. 새해인사
2. 경자년
3. 빅뱅이론
4. 스피ن

그룹요소 <div>..</div>

제목요소(h1), 문단요소(p) 등은 의미가 있지만 <div>요소는 콘텐츠 의미가 따로 있지않고 요소들을 묶을때 사용된다.

[div 예시]

The image displays three distinct web components, each enclosed in a red border and labeled with a 'div' tag at the top. The first component is a news article titled '서원뉴스' (Seowon News), featuring an image of a group of people at a table, followed by an h3 heading '2020년도 신년교례회 개최' (2020 New Year Meeting) and a paragraph of text. The second component is a '공지사항' (Notice) page with a navigation menu (일반, 학사, 장학, 취업, 행사, 정보보호) and a list of various university events. The third component is a promotional banner for 'PPT콘텐츠서비스' (PPT Content Service) with icons for templates, education, and certificates, and a '전체보기' (View All) button.

[html 문서 - 서원뉴스 부분만 코딩]

```
<div>
  <h2>서원뉴스</h2>
  
  <h3>2020년 신년교례회 개최</h3>
  <p>
    2020년도 시년교례회 개최 지난 2일, 미래창조관
    세미나실에서 2020년도 시년교례회..
  </p>
  <a href="news.html">+더보기</a>
</div>
```

위와 같이 요소들을 묶을때 <div>요소로 사용되며 <div>요소는 연관된 콘텐츠 요소들을 묶어 하나의 그룹으로 스타일을 입힐 수 있다.

span요소 ..

요소는 <div> 요소와 마찬가지로 콘텐츠 의미는 없으나 특정 글자나 요소 스타일을 꾸미고자 할 경우 사용할 수 있다. <div>와 다른 점은 <div>는 여러개의 요소를 묶을때 활용이되며 요소는 글자나 인라인요소*에 사용된다.

[html 문서]

```
<p>  
  <span style="color:red">웹표준</span>을 지키면서  
  웹사이트를 제작해야합니다.  
</p>
```

[브라우저]

웹표준을 지키면서 웹사이트를 제작해야합니다.

 요소는 아무 의미도 없는 요소로써 특정 글자에 스타일을 꾸며주고자할때 사용된다. 다음 장에서 살펴볼 내용이지만 요소는 가장 대표적인 인라인요소*이며, <div>요소는 가장 대표적인 블록요소*로 사용된다.

강조요소 .. or ..

특정 콘텐츠(글자/이미지)를 강조하고자할때 요소나 요소로 의미를 부여한다. 여기서 은 좀 더 강한 강조로 사용되며 요소는 보다 약한 강조로 의미를 부여한다.

[html 문서]

```
<p>  
  <strong>웹표준</strong>과 <em>웹 접근성</em>을  
  지키면서 웹사이트를 만드셔야 좋은 사이트라 할 수 있습니다.  
</p>
```

* 여기에서 인라인 요소는 글자와 같은 형태이고 블록 요소는 박스와 같은 형태라고만 알아 두고 다음에 나올 css챕터의 블록 요소와 인라인 요소파 트에서 자세히 살펴보도록 하자.

[브라우저]

웹표준과 웹 접근성을 지키면서 웹사이트를 만드셔야 좋은 사이트라 할 수 있습니다.

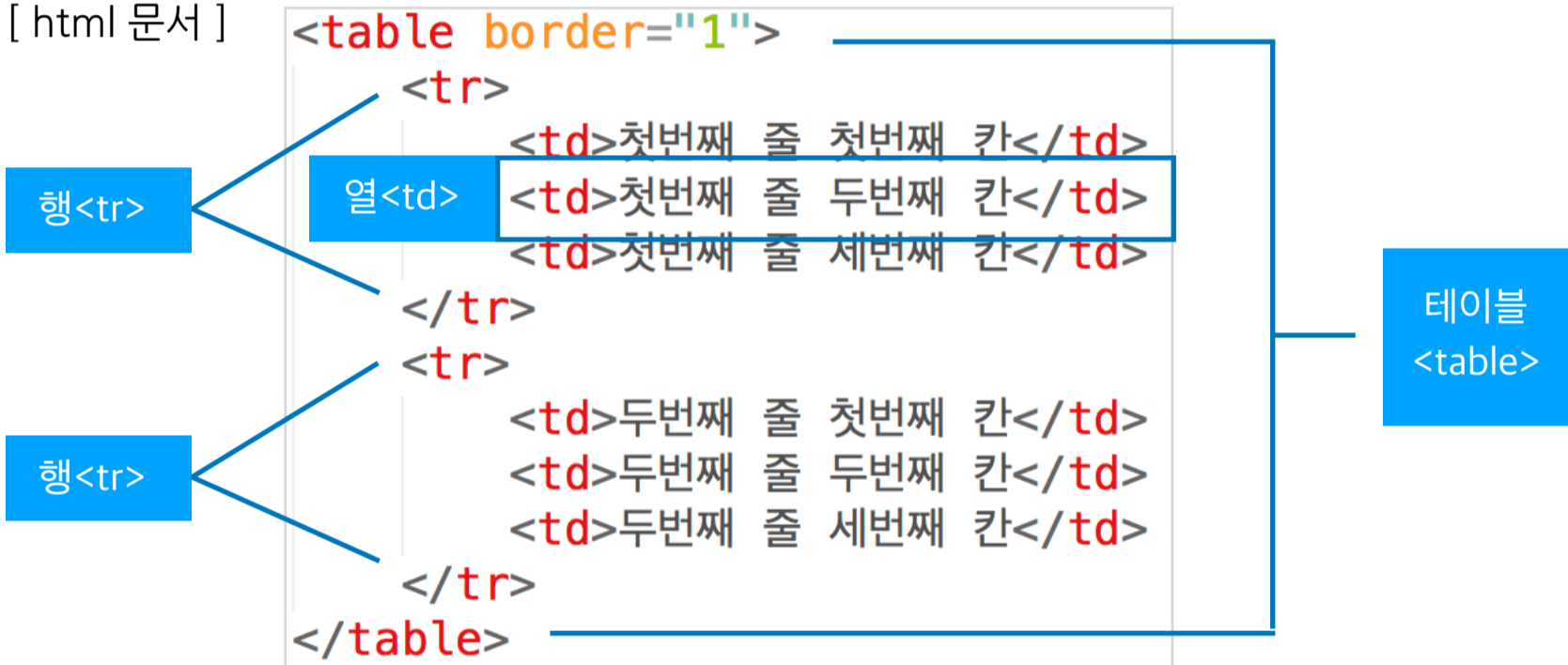
표 요소 <table>..</table>

표로 표현할만한 데이터 형태를 만들고자할때 <table>요소를 표현하면 좋습니다.

테이블 구조

테이블을 크게 행<tr> 안에 열<td>이 배치되어 있는 구조를 가졌다.

[html 문서]



[브라우저]

| | | |
|-------------|-------------|-------------|
| 첫번째 줄 첫번째 칸 | 첫번째 줄 두번째 칸 | 첫번째 줄 세번째 칸 |
| 두번째 줄 첫번째 칸 | 두번째 줄 두번째 칸 | 두번째 줄 세번째 칸 |

그리고 행은 테이블의 머리 행<thead>요소와, 몸체 행<tbody>요소, 테이블 바닥이 될 수 있는 바닥 행<tfoot>요소로 나눌 수 있으며. 테이블의 머리 행<thead>과 바닥 행<tfoot>요소는 고정되어 있다. 물론, <tfoot> 나 <thead>는 생략되는 테이블도 있기때문에 생략 가능하다.

[html 문서]

```

<table border="1">
  <thead>
    <tr>
      <th scope="col">날짜</th>
      <th scope="col">연필</th>
      <th scope="col">지우개</th>
      <th scope="col">필통</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th scope="row">합계</th>
      <td>100개 </td>
      <td>50개 </td>
      <td>30개 </td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>2020년 1월 1일</td>
      <td>50개</td>
      <td>20개</td>
      <td>10개</td>
    </tr>
    <tr>
      <td>2020년 1월 2일</td>
      <td>50개</td>
      <td>30개</td>
      <td>20개</td>
    </tr>
  </tbody>
</table>

```

행<tr>

제목칸<th>

scope속성

머리 행 <thead>

행<tr>

칸<td>

바닥 행 <tfoot>

테이블 <table>

행<tr>

바디 행 <tbody>

행<tr>

[브라우저]

| 날짜 | 연필 | 지우개 | 필통 |
|-------------|-----|-----|-----|
| 2020년 1월 1일 | 50개 | 20개 | 10개 |
| 2020년 1월 2일 | 50개 | 30개 | 20개 |

머리 행<thead>
 바디 행<tbody>
 바닥 행<tfoot>

scope 속성 : 제목 칸<th> 요소에서 행 제목이인지 열 제목인지를 알려주는 속성
 scope = "col" : 열 제목
 scope = "row" : 행 제목

웹 접근성 테이블 코딩

웹 접근성까지 고려하여 코딩하면 <caption> 요소와 <colgroup> 요소를 추가해야한다.

```
<table border="1">
  <caption> 날짜별 연필, 지우개, 필통 재고 합계 표</caption>
  <colgroup>
    <col style="width:40%">
    <col style="width:20%">
    <col style="width:20%">
    <col style="width:20%">
  </colgroup>
  <thead>
    <tr>
      <th scope="row">날짜</th>
      <th scope="row">연필</th>
      <th scope="row">지우개</th>
      <th scope="row">필통</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th scope="row">합계</th>
      <td>100개 </td>
      <td>50개 </td>
      <td>30개 </td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>2020년 1월 1일</td>
      <td>50개</td>
      <td>20개</td>
      <td>10개</td>
    </tr>
    <tr>
      <td>2020년 1월 2일</td>
      <td>50개</td>
      <td>30개</td>
      <td>20개</td>
    </tr>
  </tbody>
</table>
```

여기서 <caption> 요소에는 테이블의 상세설명을 기입하고, <colgroup> 요소에는 테이블 열의 개수만큼 <col> 요소를 추가시키고 width 값을 주어 칸의 크기를 지정해준다.

폼 (FORM)

웹사이트에서 회원가입이나 로그인 등 정보를 입력하고자할때 사용되는 폼들은 <form>요소 안에 폼 컨트롤 요소를 넣어 폼을 완성한다.

폼은 서버와의 통신으로 데이터를 주고 받는 역할까지 하지만 그 과정은 여기서 다루지는 않고, 폼에 의미를 부여하고 구조를 형성하는 방법만 살펴보도록하자.

폼은 반드시 <form>요소 안에 포함해야 한다. HTML은 다음과 같은 폼 컨트롤을 제공한다.

- 텍스트 필드 <input>
- 체크박스 <input>
- 라디오 버튼 <input>
- 버튼 <button>..</button>
- 셀렉트 박스 <select><option>..</option></select>
- 텍스트 에어리어 <textarea>..</textarea>

폼 요소 <form>..</form>

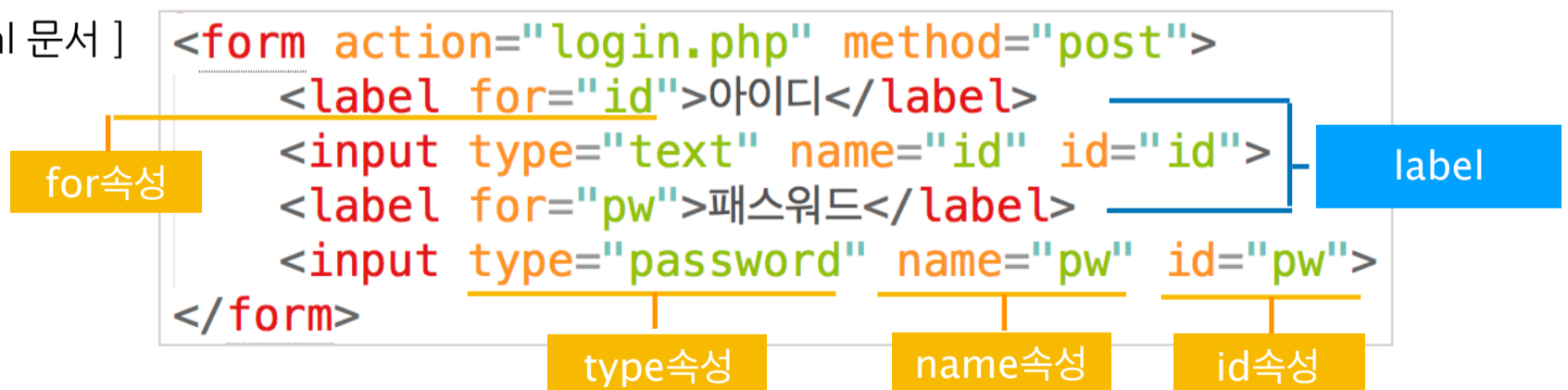
<form> 요소는 모든 웹 폼 컨트롤을 담는다. <form>요소는 **method** 속성을 통해 폼에 입력된 데이터가 전송될 때. 어떤 작업을 수행할지 지어한다. 또한, 필수 속성인 **action** 속성을 통해 전송된 데이터를 처리하고 반환 값을 보내줄 수 있는 서버 스크립트의 URL을 지정한다.

<input> 요소

text와 password

text,password필드는 한 줄의 텍스트 데이터를 입력 받을 수 있는 요소이다. text필드는 문자로 보여지고 password는 별(*)표시 등으로 숨겨 문자를 숨겨 표시한다.

[html 문서]



[브라우저]

아이디 패스워드

아이디 패스워드

type : 타입 속성 값에 따라 서로 다른 폼 컨트롤을 구분하게 된다.
name : 폼 데이터의 구분자(변수)가 된다.
id : 태그의 이름이 되며 id로 태그를 구별하여 접근할 수 있다.
for : label 요소는 폼 컨트롤 요소의 문구를 지정해주는 요소이며, label와 연결된 폼 컨트롤 요소를 for 속성 값과 같은 값을 갖는 id 속성의 요소와 연결하게 된다.

체크박스 , 라디오 버튼

여러개의 선택지 중 한개만 선택하는 그룹을 제공하고 싶을때는 라디오 버튼으로 작성하며, 다중 선택이 가능한 그룹을 제공하고 싶을때는 체크박스 요소를 작성한다.

[html 문서]

```
<form action="login.php" method="post">
  <div>
    <label for="male">남성</label>
    <input type="radio" name="gender" id="male" value="남성" checked="checked">
    <label for="female">여성</label>
    <input type="radio" name="gender" value="여성" id="female">
  </div>
  <div>
    <label for="reading">독서</label>
    <input type="checkbox" name="reading" id="reading" value="독서">
    <label for="music">음악감상</label>
    <input type="checkbox" name="music" id="music" value="음악감상" checked="checked">
    <label for="movie">영화감</label>
    <input type="checkbox" name="movie" value="영화감상" id="movie">
  </div>
</form>
```

checked

value 속성

[브라우저]

남성 여성
독서 음악감상 영화감

여기서 라디오 버튼과 체크박스의 차이는 라디오버튼의 name 속성 값은 동일 해야하며, 체크박스의 name 속성값은 동일하면 안된다.

checked : 라디오 버튼과 체크박스는 checked 속성을 넣어 선택된 상태로 지정할 수 있다. (선택됨)
value : 선택된 항목의 전달 되는 데이터 값을 넣어준다.

<button>.. <button> 요소

버튼을 만들어주는 요소로서 type속성 값에 따라 기능을 달라진다. 기본 type 속성값은 submit이며 submit은 폼을 전송하는 기능이 있다. type 속성값이 reset일 경우 폼 내용을 초기화 시키며, type 속성값이 button 일때는 아무 기능이 없다.

[html 문서]

```
<form action="login.php" method="post">
  <button type="submit">전송하기</button>
  <button type="reset">초기화하기</button>
  <button type="button">버튼</button>
</form>
```

[브라우저]



<select><option>..</option></select> 요소

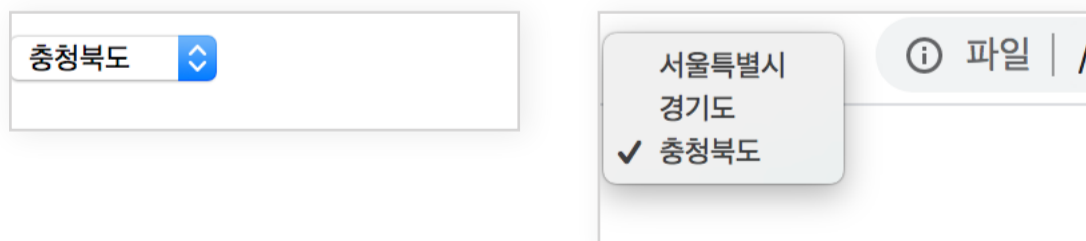
라디오버튼이나 체크박스 처럼 선택할때 사용되며 리스트 형태로 선택항목이 나타난다. <select> 요소 안에는 선택항목을 <option></option> 요소로 작성한다.

[html 문서]

```
<form action="login.php" method="post">
  <label for="region">지역선택</label>
  <select name="region" id="region">
    <option value="서울">서울특별시</option>
    <option value="경기">경기도</option>
    <option value="충북" selected="selected">충청북도</option>
  </select>
</form>
```

Selected 속성

[브라우저]



selected : 셀렉트박스에 selected 속성을 넣어 선택된 상태로 지정할 수 있다. (선택됨)

<textarea> .. </textarea> 요소

장문의 텍스트 데이터를 입력 받을 수 있는 요소이다.

[html 문서]

```
<form action="login.php" method="post">
  <label for="self-int">자기소개</label>
  <textarea name="self" id="self-int" rows="5" cols="100"></textarea>
</form>
```

rows속성 cols 속성

[브라우저]

자기소개

rows : 행 수 (5줄)

cols : 가로 글자 수 (100자)

* rows 속성과 cols 속성으로 textarea 크기를 지정할 수도 있지만 style로 지정해줄 수도 있다.

폼 그룹 <fieldset>..</fieldset>

폼 안에서 여러 그룹의 폼을 표현할때 폼안에서는 <fieldset> 요소로 그룹을 묶어주며 <legend>요소를 통해 폼 그룹의 제목을 작성한다.

[html 문서]

```
<form action="login.php" method="post">
  <fieldset>
    <legend>로그인 폼</legend>
    <label for="id">아이디</label>
    <input type="text" name="id" id="id">
    <label for="pw">패스워드</label>
    <input type="password" name="pw" id="pw">
  </fieldset>
  <fieldset>
    <legend>비회원 로그인 폼</legend>
    <label for="phone">전화번호</label>
    <input type="text" name="phone" id="phone">
    <label for="pw">패스워드</label>
    <input type="password" name="pw" id="pw"></legend>
  </fieldset>
</form>
```

특수문자

HTML 문서 내에서 띄어쓰기나 카피라이트 등과 같은 특수문자, 숫자형 문자 참조 등을 사용하여 특수기호를 표현한다.

자주 사용되는 특수 문자 표시법은 다음과 같다.

- 띄어쓰기 :
- 앰퍼샌드(&) : &
- copyright(저작권) : ©
- 부등호 (< , >) : < , >
- 작은따옴표 (' , ') : ‘ , ”

라. 시맨틱 웹(Semantic Web)

대부분의 인터넷 사용자는 원하는 정보를 취득하기 위해 Google이나 Naver와 같은 검색사이트를 이용한다. 웹사이트는 검색엔진에의 노출이 매우 중요하다. 당연한 것이 검색엔진에 노출되지 않는 웹사이트에는 접속하는 이도 없기 때문이다. 검색엔진은 이 시대의 가장 중요한 도구이다.

SEO(검색엔진 최적화: Search Engine Optimization)같은 마케팅 도구를 사용하여 검색엔진이 본인의 웹사이트를 검색하기 알맞은 구조로 웹사이트를 조정하기도 하는데, 이것은 기본적으로 검색엔진이 웹사이트 정보를 어떻게 수집하는지 아는 것으로 부터 시작된다.

검색엔진은 로봇(Robot)이라는 프로그램을 이용해 매일 전세계의 웹사이트 정보를 수집한다.(이것을 크롤링이라 하며 검색엔진의 크롤러가 이를 수행한다.) 그리고 검색 사이트 이용자가 검색할 만한 키워드를 미리 예상하여 검색 키워드에 대응하는 인덱스(색인)을 만들어 둔다.(이것을 인덱싱이라 하며 검색엔진의 인덱서가 이를 수행한다.)

인덱스를 생성할 때 사용되는 정보는 검색 로봇이 수집한 정보인데 결국 웹사이트의 HTML 코드이다. 즉, 검색 엔진은 HTML 코드만으로 그 의미를 인지하여야 하는데 이때 시맨틱 요소(Semantic element)를 해석하게 된다.

시맨틱 웹이란 웹에 존재하는 수많은 웹페이지들에 메타데이터(Metadata)를 부여하여, 기존의 잡다한 데이터 집합이었던 웹페이지를 '의미'와 '관련성'을 가지는 거대한 데이터베이스로 구축하고자 하는 발상이다.

non-semantic 요소

div, span 등이 있으며 이들 태그는 content에 대하여 어떤 설명도 하지 않는다.

semantic 요소

H1,p,form, table, img 등이 있으며 이들 태그는 content의 의미를 명확히 설명한다,

마. HTML5 추가된 시맨틱 요소

| | |
|------------------------------|----------------------------|
| <code><header></code> | |
| <code><nav></code> | |
| <code><section></code> | <code><aside></code> |
| <code><article></code> | |
| <code><footer></code> | |

| | |
|----------------|--|
| Header | 문서 또는 섹션 헤더 내용.(로고,검색,일 메뉴등) |
| Nav | 문서의 네비게이션 |
| Section | 문서의 섹션 (서로 관계있는문서를 분리) |
| Article | 내용이 독립적이고, 홀로 설 수 있는 내용.(예 : 블로그, 포럼 글, 뉴스 기사) |
| Footer | 문서 또는 섹션 바닥 (주소,카피라이터 등) |
| Aside | 문서의 주요 내용이 아닌 부수적인 내용. (예 : 광고) |

2장 CSS

가. CSS란?

CSS 언어

CSS는 HTML 마크업 문서의 요소를 화면에 어떤 모습으로 출력할지를 서술하는 언어이다. 스타일시트^{css}는 요소한테만 사용할 수 있다.

나. CSS 작성 방법

스타일은 요소에 직접 넣는 인라인 스타일 방식, HTML문서 <head>에 넣는 내부 스타일 방식, 스타일시트^{css} 파일을 따로 만들어 HTML에서 링크를 걸어 넣는 외부 스타일 방식 이렇게 3가지로 방법이 있다.

인라인 스타일

인라인 스타일은 각 개별 요소에 style 속성으로 스타일을 넣어 꾸며주는 방식이다.

인라인 스타일시트^{css} 언어는 다음과 같은 형태로 서술한다. 세미콜론(;) 으로 여러 개의 스타일로 꾸밀 수 있다.

```
style = "스타일 속성 : 속성 값; 스타일 속성 : 속성 값;"
```

[html 문서]

```
<p style="color: red; font-size:24px">  
인라인 스타일로 요소 꾸미기  
</p>
```

Style 속성

[브라우저]

인라인 스타일로 요소 꾸미기

style : style 속성 값으로 css 언어를 서술해 직접 스타일을 적용할 수 있다.

인라인 스타일 방식은 동일한 스타일이 필요로 하는 요소들에게 반복적인 style 을 서술해야 하기 때문에 좋지 않은 방법이다.

내부 스타일

HTML 문서 내의 <head></head> 요소에 스타일시트^{css} 언어를 서술하는 방식이다.

스타일시트^{css} 다음과 같이 서술한다.

```
선택자 { 스타일 속성 : 속성 값 ; 스타일 속성 : 속성 값 ; }
```

인라인 직접 선택한 요소에 스타일을 주기때문에 선택자가 없지만 요소에 직접 주지않는 내부스타일이나 외부 스타일 경우 선택자가 필요하다.

```
<head>
  <title>내부 스타일</title>
  <meta charset="utf-8">
  <style type="text/css">
    선택자 p { color : red; font-size : 24px; }
  </style>
</head>
```

선택자는 <p> 요소이며 color 속성은 글자의 색을 변경할 수 있는 속성이다 red 는 color 속성의 값으로써 <p> 요소 안에 있는 글자의 색을 빨강으로 꾸며주는 스타일이다.

내부 스타일 방식은 선택된 요소(선택자)들이 스타일과 같은 HTML 문서 내에 있으면 모두 한꺼번에 적용된다.

HTML 문서 내에 <p> 요소가 여러 개일 경우 내부 스타일은 문서 내 모든 <p> 요소에 스타일이 적용되지만 인라인 스타일 같은 경우 <p> 요소마다 style을 모두 서술해줘야 한다.

외부(링크) 스타일

HTML 문서에 CSS를 적용하는 방식 중 가장 일반적으로 사용도는 방식이며, 외부 CSS파일을 따로 만들어 HTML의 <head>요소 안에 <link>요소에 CSS파일을 링크하여 HTML 요소에 스타일을 적용시키는 방법이다.

이렇게 하면 HTML에서 디자인 요소를 분리하여 얻을 수 있는 모든 장점을 취할 수 있다.

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

rel속성

type속성

href속성

rel : 링크할 파일과의 관계를 지정. 링크할 파일과의 관계를 문서의 스타일 시트로 지정한다.
href : 링크할 CSS 파일의 위치 정보를 지정.
type : 브라우저에 어떤 언어의 스타일 시트를 사용할지를 알려준다. (HTML5에서는 생략 가능)

다. CSS 기본 구문

선택자 종류

스타일을 구성하는 부분 중 선택자는 여러 방법으로 요소를 선택할 수 있다. 태그 선택자, 클래스 선택자, 아이디 선택자, 하위 선택자, 자식 선택자 등 수많은 선택자들이 있으나 이 책에서 다 살펴볼 수는 없기 때문에 가장 많이 사용되어 꼭 알아야 하는 선택자들만 살펴보고자 한다.

태그(tag)선택자

HTML 요소의 태그이름으로 지칭하는 선택자가 태그 선택자이다. 이 선택자는 h1, p, div 처럼 HTML 요소의 이름 지칭함으로써 선택자가 지칭하는 이름을 갖는 모든 요소에게 스타일을 적용시킬 수 있다.

```
h2 { color : #00f }
```

* #00f는 파란색을 의미하며 CSS에서 컬러를 지정하는 방법은 #을 시작해서 16진법으로 색상 코드를 입력하는 방법과 rgb(0,0,0)으로 색상을 지정할 수 있다. 웹에서는 red, green, blue (rgb 컬러) 3개의 색을 0~255 수치로 표현하며 16진 법의 FF는 255를 이기 때문에 #FFFFFF는 흰색이 되고 #FF0000(빨강), #00FF00(녹색), #0000FF(파랑)이 된다. 이때 반복되는 숫자는 하나로 줄여 쓸 수 있다. #FF0000 -> #F00과 같다. 또는 rgb(255,0,0) 이런 식으로 10진 법의 숫자를 넣어서 색을 표현할 수 있다.

클래스 선택자

HTML요소의 이름과는 상관없이 스타일에 이름을 부여하여 요소에 class 속성에 스타일 이름을 넣어 스타일을 적용 시킬 수 있다. 이때 스타일의 이름을 만들어 사용하는데 dot(.)으로 시작해서 이름을 작명해서 선택자를 만들어 줄 수 있는데 이 선택자를 클래스 선택자라 한다.

```
.point { color : #f00 ; background-color: #ff0 ; }
```

point 클래스 선택자의 스타일을 요소에 적용 시키고자 할 때에는 HTML 요소에 class = "point" 라고 넣어주면 스타일이 적용된다.

```
<span class="point">태그 선택자</span>
```

class 속성

클래스 선택자 사용예시

[스타일]

```
.point { color : #f00 ; background-color: #ff0 ; }
```

[html 문서]

```
<h2>선택자의 종류</h2>
<ul>
  <li>태그선택자</li>
  <li class="point">클래스선택자</li>
  <li>아이디선택자</li>
</ul>
<p>
  <span class="point">태그 선택자</span>는 요소 이름,
  <span class="point">클래스 선택자</span>는 스타일의 이름이다.
</p>
```

[결과 (브라우저)]

선택자의 종류

- 태그선택자
- 클래스선택자
- 아이디선택자

태그 선택자는 요소 이름, 클래스 선택자는 스타일의 이름이다.

아이디 선택자

아이디는 1장에 <input> 요소에 id 속성을 넣어 요소를 구별할 수 있는 이름을 만들 수 있다 언급했는데, 이처럼 요소에 이름(id)이 있는 경우 아이디를 선택자로 지칭하여 스타일을 꾸밀 수 있다.

```
#pw { color : #f00 ; background-color: #ff0 ; }
```

pw 라는 이름을 갖고 있는 요소에 스타일을 적용시킨다.

[html 문서]

```
<form action="login.php" method="post">
  <label for="id">아이디</label>
  <input type="text" name="id" id="id">
  <label for="pw">패스워드</label>
  <input type="password" name="pw" id="pw">
</form>
```

[브라우저]

아이디 패스워드

그룹 선택자

여러개의 선택자가 같은 스타일을 적용되기 원할때 콤마(,)로 선택자들을 묶어 동시에 스타일을 적용 시킬 수 있다.

```
#pw,#id,p,.point { color : #f00 ; background-color: #ff0 ; }
```

하위 선택자

하위 선택자는 특정 선택자 요소 안에서만 선택자를 찾고자할때 선택자와 선택자 사이에 띄어쓰기를 넣어 특정 요소에 제한을 두어 선택자를 찾을 수 있다.

[html 문서]

```
<div class="menus">
  <a href="#">홈</a>
  <ul>
    <li><a href="#">회사소개</a></li>
    <li><a href="#">사업소개</a></li>
    <li><a href="#">공지사항</a></li>
  </ul>
</div>
<div class="links">
  <a href="#">이전링크</a>
  <a href="#">다음링크</a>
  <ul>
    <li><a href="#">배너1</a></li>
    <li><a href="#">배너2</a></li>
    <li><a href="#">배너3</a></li>
  </ul>
</div>
```

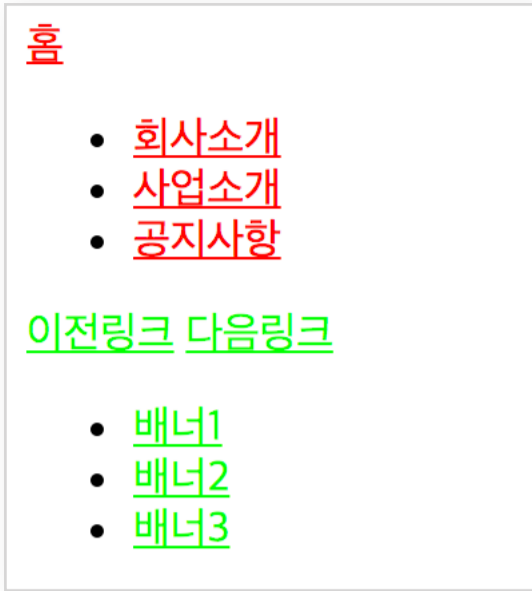
위의 HTML에 문서에서 .menus <div>요소와 .links <div>요소 안에 있는 <a> 요소에 다른 스타일을 적용하고자 할 때 하위 선택자를 활용할 수 있다.

.menus 안에 있는 <a>요소는 빨간색 글자가 적용되고, .links 안에 있는 <a>요소는 초록색 글자가 적용되고자 할 때 다음과 같이 서술할 수 있다.

[CSS]

```
.menus a { color : #f00; }
.links a { color : #0f0; }
```

[결과]



자식 선택자

자식 선택자는 특정 선택자의 자식 요소를 선택하고자 할 때 “>”를 넣어 자식 요소를 선택하여 스타일을 적용시킬 수 있다. 하위 선택자와 비슷하게 생각할 수 있지만 하위 선택자는 특정 선택자 안에 있는 모든 요소들에 접근할 수 있지만 자식 선택자는 자식으로만 있는 요소에만 적용이 된다.

[CSS]

```
.menu>a{ color : #f00; }  
.links>a{ color : #0f0; }
```

위 하위 선택자 [html 문서] 와 같은 요소에서 `.menu` 요소와 `.links` 요소 안에 있는 모든 `<a>` 요소에 color 스타일이 적용되었지만 자식 선택자를 사용하게 되면 다음과 같은 결과가 나온다.

[결과]



하위 선택자와 다른 점은 `.menu` 요소 안에 있던 모든 `<a>` 요소에 빨강 글자로 변경되었지만 위의 결과는 ` 홈 ` 요소만 적용된 것을 확인할 수 있다. `.links` 요소도 마찬가지로 ` 이전링크 ` 와 ` 다음링크 ` 요소에만 스타일이 적용된 것을 확인할 수 있다.

속성 선택자

요소의 속성의 속성이름, 속성값으로 선택하고자 할 때 “요소[속성명]” 을 선택하여 스타일을 적용시킬 수 있다.

```
[ html ] <a href="#">링크1</a>
<a href="#" target="_self">링크2</a>
<a href="#" target="_blank">링크3</a>
```

```
[ style ] a[target]{ color : red; }
```

```
[ 결과 ] 링크1 링크2 링크3
```

```
[ style ] a[target="_blank"]{ color : red; }
```

```
[ 결과 ] 링크1 링크2 링크3
```

| 선택자 | 예 | 설명 |
|-------------|--------------------|---|
| [속성] | a[target] | a요소 중 target 속성을 갖는 요소 |
| [속성 = 값] | a[target = _blank] | a요소 중 target속성 값으로 _blank 값을 갖는 요소 |
| [속성 ~ = 값] | [title ~ = abc] | 모든 요소 중 title속성 값 중 abc단어를 포함한 요소 |
| [속성 = 값] | [title = abc] | 모든 요소 중 title 속성 값이 abc이거나 abc로 시작하는 요소 |
| [속성 ^ = 값] | [title ^ = abc] | 모든 요소 중 title속성 값이 abc로 시작하는 요소 |
| [속성 \$ = 값] | [title \$ = abc] | 모든 요소 중 title 속성 값이 abc로 끝나는 요소 |
| [속성 * = 값] | [title * = abc] | 모든 요소 중 title 속성 값이 abc를 포함한 요소 |

전체 선택자

HTML 내에 사용되는 모든 요소를 선택하고자 할 때 전체 선택자 (*) 를 사용할 수 있다.

```
* { margin : 0 ; }
```

가상클래스 선택자

상황에 따라 분류하여 선택하거나, 특정 요소의 첫번째, 마지막 요소 및 특정 순서의 요소를 선택하여 스타일을 적용시키고자 할때 사용되며 “:” 로 구별한다.

:link, :hover, :visited, :active, :focus

[html] ` 웹사이트 링크`

링크된 파일에 방문했을 때

```
a:link { color : red }
```

[결과]

[웹사이트 링크](#)

a 태그에 링크가 걸렸을 때

```
a:visited {color : green;}
```

[결과]

[웹사이트 링크](#)

요소에 마우스 오버했을 때

```
a:hover { color : skyblue; }
```

[결과]

[웹사이트 링크](#)

요소가 활성화 되었을 때 (요소에 마우스 다운 했을때)

```
a:active { background : yellow; }
```

[결과]

[웹사이트 링크](#)

요소가 포커스에서 나왔을때

```
a:focus { background : black; }
```

[결과]

[웹사이트 링크](#)

:first-child, :last-child, :nth-child(n)

[html]

```
<ul>
  <li>리스트1</li>
  <li>리스트2</li>
  <li>리스트3</li>
  <li>리스트4</li>
  <li>리스트5</li>
</ul>
```

특정 형제 요소들 중 첫번째 요소를 선택하고자 할때

```
li:first-child { color : red; }
```

[결과]

- 리스트1
- 리스트2
- 리스트3
- 리스트4
- 리스트5

특정 형제 요소들 중 마지막 요소를 선택하고자 할때

```
li:last-child { color : red; }
```

[결과]

- 리스트1
- 리스트2
- 리스트3
- 리스트4
- 리스트5

특정 형제 요소들 중 특정 순서에 있는 요소를 선택하고자 할때

```
li:nth-child(2) { color : red; }
```

[결과]

- 리스트1
- 리스트2
- 리스트3
- 리스트4
- 리스트5

가상요소선택자

특정 요소의 첫번째 자식(:before), 마지막 자식(:after)으로 요소를 추가 할 수 있다. 스타일 속성으로 content 속성은 필수로 입력해야한다.

[html] `<p> p요소 컨텐츠 입니다. </p>`

[style] `p:before { content : '시작'; color : red; }
p:after { content : '끝'; color : blue; }`

[결과] 시작 p요소 컨텐츠 입니다. 끝

라. 선택자 우선순위

한 요소에 여러 선택자들로 똑같은 스타일 속성이 적용될때 하나의 스타일만 표현 될 수 있다. 이때 선택자가 태그선택자, 클래스 선택자, 아이디 선택자에 따라 우선순위가 달라진다.

[html 문서] `<h1 class="redColor" id="title"> 타이틀 </h1>`

[CSS] `#title { color : blue ; }
.redColor { color : red ; }
h1 { color : blue ; }`

위 HTML 문서의 <h1> 요소에는 클래스, 아이디, 태그 선택자로 각 color 스타일 속성이 적용된다. 하지만 <h1>요소는 모든 글자색을 표현 할 수 없으므로 하나의 글자색만 표현된다. 이 중 우선순위가 가장 높은 것은 #title 선택자인 아이디 선택자가 가장 우선에 있게되어 글자 색은 파란색으로 적용 된다. 아이디가 없을 경우 .redColor인 클래스 선택자가 적용이 된다. 가장 우선순위가 낮은건 태그 선택자이다.

`<h1 style="color: green" class="redColor" id="title"> 타이틀 </h1>`

여기서 위와 같이 <h1> 요소에 인라인 스타일도 있으면 이때는 인라인 스타일이 최우선이 되어 <h1> 요소에는 녹색 글자로 표현된다.

특정 속성을 최우선으로 두고싶을때는속성 값 안에 !important 를 넣어주면 가장 우선 적용된다.

`#title { color : blue ; }
.redColor { color : red ; }
h1 { color : blue !important; }`

위와 같이 h1 선택자의 color 속성 값 옆에 !important 를 추가시키면 <h1> 요소의 글자색은 파란색으로 적용된다.

우선순위 순서

!important > 인라인스타일 > 아이디 선택자 > 클래스 선택자 > 태그 선택자

우선순위는 개별성 규칙에 의해 결정된다. 선택자별 개별성은 다음과 같다.

| 선택자 | 사용 예 | 개별성 |
|-------------|-------------------|------|
| 태그(type)선택자 | P | 1 |
| 클래스 선택자 | .point | 10 |
| 아이디 선택자 | #gnb | 100 |
| 인라인 선택자 | style="color:red" | 1000 |
| 가상요소 | :first-child | 1 |
| 가상클래스 | :before | 10 |

마. 스타일 속성 살펴보기

글자관련 속성

| 속성 | 속성 값 | 설명 |
|-----------------|---|--|
| Font-style | font-style: normal italic oblique initial inherit; | 이탤릭체 등의 글꼴 스타일 |
| font-weight | font-weight: normal bold bolder lighter number initial inherit; | 글자 두께 |
| font-variant | font-variant: normal small-caps initial inherit | 소문자를 작은 대문자 등의 스타일 |
| font-size | font-size: medium xx-small x-small small large x-large xx-large smaller larger length initial inherit ; | 글자 크기 (px,pt,%,em 단위로 크기를 지정 할 수 있다.) |
| line-height | line-height: normal number length initial inherit; | 줄 간격 (px,cm,%,em 단위로 크기를 지정 할 수 있다.) |
| font-family | font-family : dotum, 돋움, sans-serif (글꼴) | 글꼴 |
| [축약형] font | font: font-style font-variant font-weight font-size/line-height font-family | |
| text-decoration | text-decoration: overline line-through underline none initial inherit; | 밑줄, 취소선 등의 스타일 |
| text-indent | text-indent: length initial inherit; | 들여쓰기 |

속성값으로 initial (초기값), inherit (부모 요소 값 상속) 이 모든 속성 값으로 지정할 수 있다.

색상표현 법 (RGB) - red / green / blue

기본 색상 키워드 : aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow

16진수 표현 : #ff3300

10진수 표현 : rgb(255,33,0)

알파(투명)값 표현 : rgba(0,0,0,.5) -> 1은 불투명, 0은 투명, 0.5또는 .5는 50%투명값을 갖는다.

css 단위(units)

css에서는 절대 단위와 상대 단위로 가 있다. 절대 단위는 pt(포인트), cm(센티미터), pc(파이카), in(인치)가 있고, 상대 단위에는 해상도 기준에따라 결정되는 픽셀(px), font-size 기준에 따라 결정되는 em, 부모 기준으로 결정되는 %(백분율) 단위가 있다.

해상도 기준 px

모니터 해상도에 기준으로 사이즈를 px단위로 표현하며 웹에서는 픽셀(px)로 사용된다.

font-size 기준 em

css에서는 폰트 크기를 기준으로 사이즈 지정할 수 있다. 이때 사용되는 단위가 em이다. 웹에서 기본 font-size는 16px이다.

부모 기준 %

부모의 크기를 기준으로 상대적으로 사이즈 변경을 하고자할때 사용되는 단위는 %이다.

웹 폰트

font-family 로 지정된 글꼴은 클라이언트 컴퓨터에 글꼴이 설치되어있을때 적용되기 때문에 제한적인 글꼴 표현을 서버에서 글꼴을 불러와 글꼴을 표현할 수 있다.

기본 사용법

웹 폰트는 @font-face 규칙을 사용해 적용한다.

```
@font-face {  
  font-family: 'NanumSquare';  
  font-weight: 400;  
  src: url(NanumSquareR.eot);  
  src: url(NanumSquareR.eot?#iefix) format('embedded-opentype'),  
       url(NanumSquareR.woff) format('woff'),  
       url(NanumSquareR.ttf) format('truetype');  
}
```

서버경로

폰트 지정 : `body { font-family : 'NaumSquare', sans-serif; }`

요즘은 무료로 제공되는 웹폰트를 외부스타일로 연결하여 사용할 수 있다.

[무료 웹폰트 사이트 정보]

- 눈누 - <https://noonnu.cc/>
- 구글웹폰트 - <https://fonts.google.com/>

박스모델

css의 중요한 개념으로 **블록 요소**와 **인라인 요소**가 있으며 화면에 표시되는 영역과 배치 등의 다른 특징을 갖는다. 블록요소인지 인라인 요소인지는 **display** 속성으로 결정된다.

* 블록요소 - **display : block.** * 인라인 요소 - **display : inline**

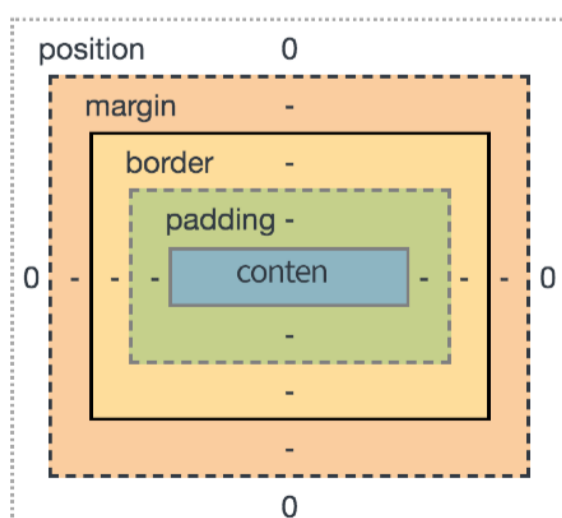
블록 요소의 특징

- 가로(width)/세로(height) 크기를 갖는다.
(블록요소의 width 기본값은 부모 상속, height 기본값은 콘텐츠(자식요소)크기만큼 갖는다.)
- 아래로 쌓인다. (요소 옆으로 위치하지 않는다.)

인라인 요소의 특징

- 가로(width)/세로(height) 크기를 갖지 않는다.
- 옆으로 나열된다. (텍스트와 같은 개념)

박스 모델 구성도



요소 안에 들어가는 부분을 콘텐츠(content)라 지칭한다.

`<p> 이부분이 콘텐츠 부분입니다. </p>`

박스 모델과 관련된 속성

| 속성 | 속성 값 | | 설명 |
|-----------------------------------|--|---|---|
| display | display: value ; 값 : none, inline, block, inline-block 등 | | 소소 |
| width | width: auto value initial inherit; | | 가로 크기 (%, em, px 단위로 표현) |
| height | height: auto value initial inherit; | | 세로 크기 (%, em, px 단위로 표현) |
| border | border-style | border-style: none hidden dotted dashed solid double groove ridge inset outset initial inherit; | 테두리 스타일 지정 |
| | border-color | border-color: color transparent initial inherit; | 테두리 색상 지정 |
| | border-width | border-width: medium thin thick length initial inherit; | 테두리 두께 |
| | 축약형 (값의 순서중요하지 않음) | border { border-style border-color border-width } | |
| margin | margin-top | margin-top: length auto initial inherit; | 요소 바깥여백 (위 여백) |
| | margin-right | margin-top: length auto initial inherit; | 요소 바깥여백 (오른쪽 여백) |
| | margin-bottom | margin-top: length auto initial inherit; | 요소 바깥여백 (아래 여백) |
| | margin-left | margin-top: length auto initial inherit; | 요소 바깥여백 (왼쪽 여백) |
| | 축약형 (값의 순서중요) | margin { top/bottom left/right } | 값이 2개일 경우 첫번째 값은 top과 bottom여백, 두번째 값은 left와 right 여백이다. |
| | | margin { top left/right bottom } | 값이 3개일 경우 첫번째 값은 top 여백, 두번째 값은 left와 right 여백, 세번째 값은 bottom 여백이다. |
| margin { top right bottom left } | | 값이 4개일 경우 첫번째 값은 top 여백, 두번째 값은 right 여백, 세번째 값은 bottom 여백, 네번째 값은 left 여백이다. | |
| padding | padding-top | padding-top: length auto initial inherit; | 요소 안쪽여백 (위 여백) |
| | padding-right | padding-top: length auto initial inherit; | 요소 안쪽여백 (오른쪽 여백) |
| | padding-bottom | padding-top: length auto initial inherit; | 요소 안쪽여백 (아래 여백) |
| | padding-left | padding-top: length auto initial inherit; | 요소 안쪽여백 (왼쪽 여백) |
| | 축약형 (값의 순서중요) | padding { top/bottom left/right } | 값이 2개일 경우 첫번째 값은 top과 bottom여백, 두번째 값은 left와 right 여백이다. |
| | | padding { top left/right bottom } | 값이 3개일 경우 첫번째 값은 top 여백, 두번째 값은 left와 right 여백, 세번째 값은 bottom 여백이 다. |
| padding { top right bottom left } | | 값이 4개일 경우 첫번째 값은 top 여백, 두번째 값은 right 여백, 세번째 값은 bottom 여백, 네번째 값은 left 여백이다. | |

display 속성

| | |
|--------------------|---|
| inline | 요소를 인라인 요소 (예 :)로 표시합니다. 높이 및 너비 속성은 영향을 미치지 않습니다 |
| block | 요소를 <p>와 같은 블록 요소로 표시합니다. 새 줄에서 시작하여 전체 너비를 차지합니다. |
| contents | 컨테이너를 사라지게하여 DOM에서 요소의 하위 요소를 다음 단계로 만듭니다. |
| flex | 요소를 블록 레벨 플렉스 컨테이너로 표시 |
| grid | 요소를 블록 레벨 그리드 컨테이너로 표시 |
| inline-block | 요소를 인라인 레벨 블록 컨테이너로 표시합니다. 요소 자체는 인라인 요소로 형식화되지만 높이 및 너비 값을 적용 할 수 있습니다 |
| inline-flex | 요소를 인라인 레벨 플렉스 컨테이너로 표시 |
| inline-grid | 요소를 인라인 레벨 그리드 컨테이너로 표시 |
| inline-table | 요소가 인라인 레벨 테이블로 표시됩니다 |
| list-item | 요소가 요소처럼 동작하게하십시오 |
| run-in | 상황에 따라 요소를 블록 또는 인라인으로 표시 |
| table | 요소가 <table> 요소처럼 동작하게하십시오 |
| table-caption | 요소가 <caption> 요소처럼 동작하게하십시오 |
| table-column-group | 요소가 <colgroup> 요소처럼 동작하게하십시오 |
| table-header-group | 요소가 <thead> 요소처럼 동작하게하십시오 |
| table-footer-group | 요소가 <tfoot> 요소처럼 동작하게하십시오 |
| table-row-group | 요소가 <tbody> 요소처럼 동작하게하십시오 |
| table-cell | 요소가 <td> 요소처럼 동작하게하십시오 |
| table-column | 요소가 <col> 요소처럼 동작하게하십시오 |
| table-row | 요소가 <tr> 요소처럼 동작하게하십시오 |
| none | 요소가 완전히 제거 |
| initial | 이 속성을 기본값으로 설정 |
| inherit | 부모 요소에서이 속성을 상속 |

display 속성은 요소의 특징을 나타내며 위에서 기본적으로 inline 요소일 경우의 특징과 block 요소의 특징을 살펴봤지만 사실, 더 많은 특징을 갖는 요소들이 있다. 이 중 inline-block 요소는 inline 요소와 block 요소의 특징이 같이 나타난다.

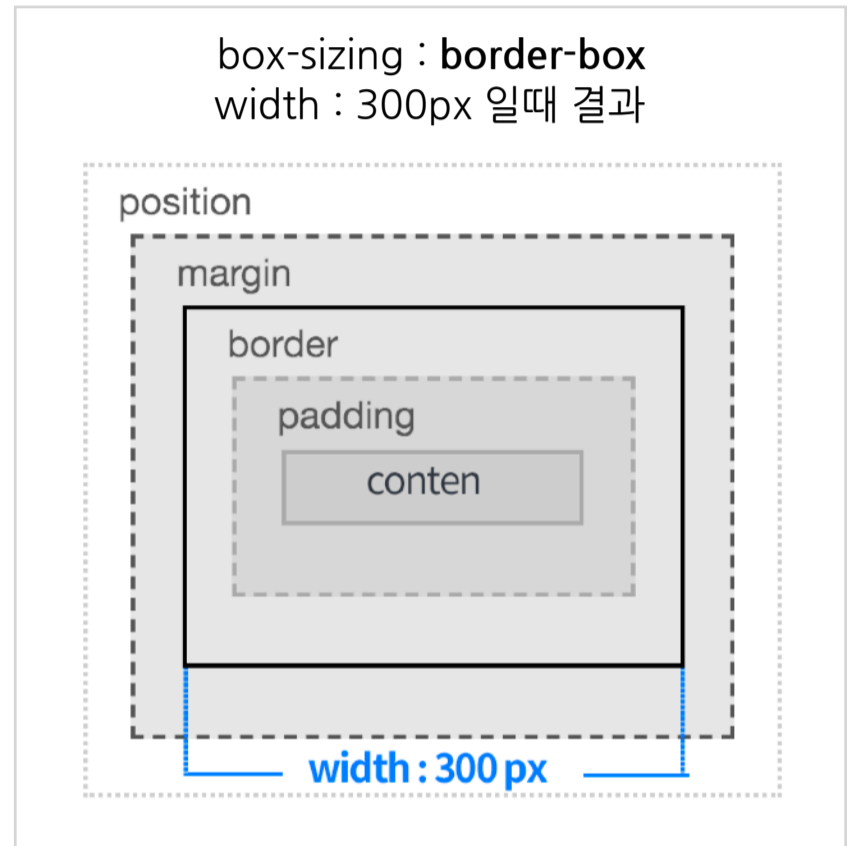
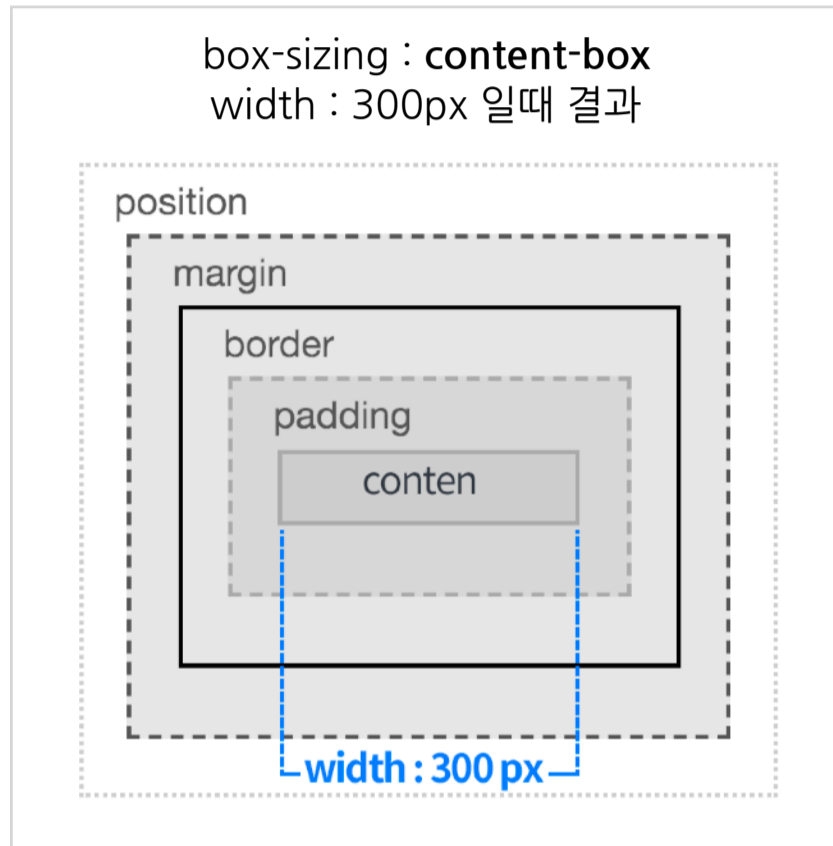
- 가로(width)/세로(height) 크기를 갖는다. (block 요소 특징)
- 옆으로 나열된다. (inline 요소의 특징)

width/height 속성

width/height 값을 지정할때 **box-sizing** 속성이 content기준인지 border까지의 기준인지에 따라 크기가 달라진다.

| 속성 | 속성 값 | 설명 |
|------------|---|------------------------------------|
| box-sizing | box-sizing: content-box border-box initial inherit; | width값이 콘텐츠크기인지 border까지의 크기인지 지정. |

box-sizing 기본값은 content-box 이다.



단위 px , % , em 등으로 지정 할 수 있다.

border 속성

요소의 테두리 스타일 적용할때 border 속성을 사용한다.

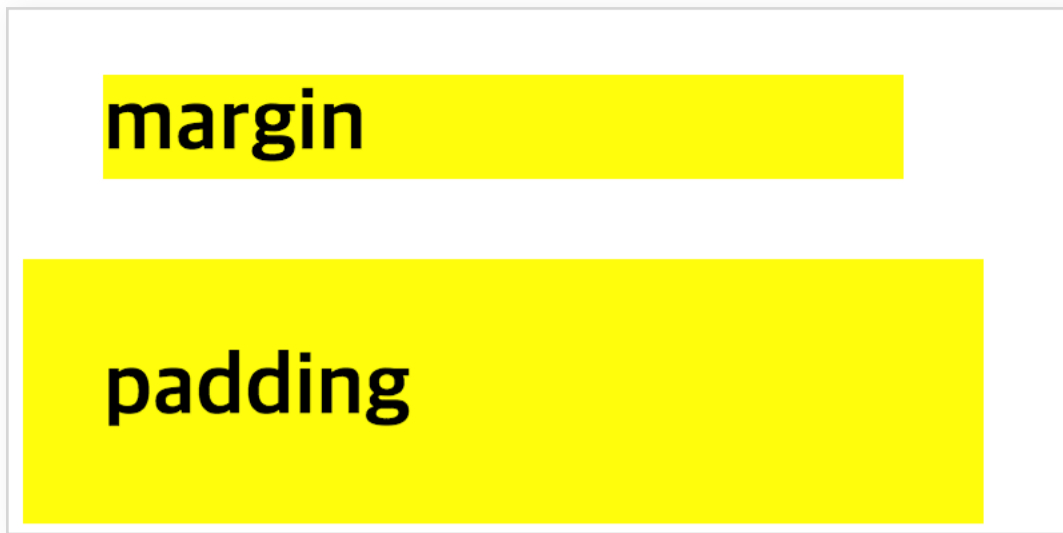
요소 여백 - margin, padding

요소의 안쪽의 여백은 padding 속성으로, 요소의 바깥 여백은 margin으로 스타일을 적용한다.

```
[ html ] <h1 class="margin"> margin </h1>  
<h1 class="padding"> padding </h1>
```

```
[ style ] .margin { background-color:#ff0; width:300px; margin:30px; }  
.padding { background-color:#ff0; width:300px; padding:30px; }
```


[결과]



노란색 박스가 h1 요소이며, margin 적용 시 요소의 바깥 부분에 여백이 생기므로 요소가 자체가 아래 왼쪽으로 30px씩 이동한 걸 알 수 있다. 여기서 margin이 바깥 여백임과 동시에 요소의 위치도 변경된다는 걸 이해하자.

padding 같은 경우 요소의 위치가 변하지는 않지만 안쪽으로 여백이 생기므로 content 주의에 여백이 생기고 content의 위치가 변경된다.

(margin과 padding을 여백으로만 생각하지 말고 여백을 줌으로써 요소나 content의 위치가 변경된다는 것도 같이 이해하자.)

블록요소 가운데 정렬 - margin-left : auto , margin-right : auto

위에 margin은 요소의 바깥 여백이며 요소의 위치가 변경된다는 걸 알 수 있다. 여기서 margin으로 요소(블록 요소)의 위치를 가운데로 오게 할 수 있는데 margin-left와 margin-right 값을 auto로 넣어주면 부모 width를 기준으로 왼쪽과 오른쪽 여백이 같은 크기를 갖게 되면서 가운데 오게 된다.

인라인요소 정렬 - text-align

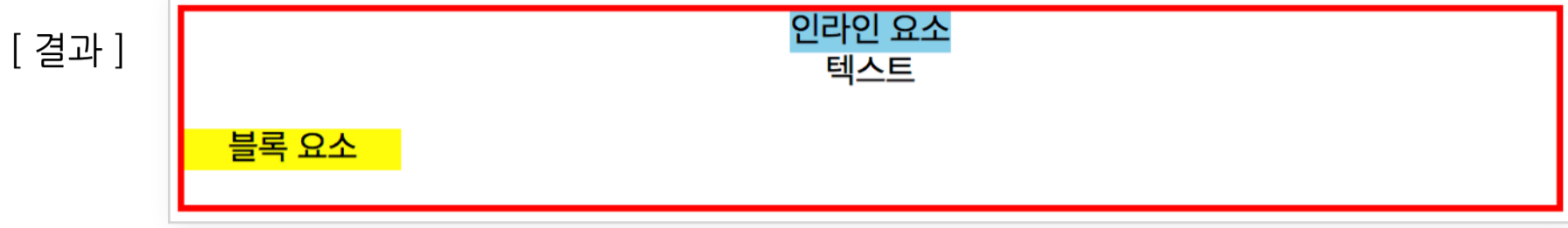
블록 요소는 margin으로 가운데 정렬을 시킬 수 있다면, 인라인 요소나 텍스트는 text-align 속성으로 가운데/왼쪽/오른쪽 정렬이 가능하다.

| 속성 | 속성 값 | 설명 |
|------------|--|---------------|
| text-align | text-align: left right center justify initial inherit; | 인라인 요소 텍스트 정렬 |

[html]

```
<div class="block">  
  <span class="inline">인라인 요소</span> <br>  
  텍스트  
  <p>블록 요소</p>  
</div>
```

```
[ style ]
.block { border:solid 3px red; text-align: center;}
.inline{ background-color: skyblue }
.block p { background-color: yellow; width: 100px;}
```



<div class="block">...</div> 요소에 text-align : center를 넣으면 div 요소가 정렬되는 것이 아니라 div 요소 content 중 인라인 요소 (요소)와 텍스트 글자만 정렬이 되고 블록 요소(<p>요소)는 정렬되지 않는 것을 확인할 수 있다.

float 속성

| 속성 | 속성 값 | 설명 |
|-------|---|-----------------|
| float | float: left right initial inherit; | 왼쪽 오른쪽으로 요소 띄우기 |
| clear | clear: left right both initial inherit; | float 초기화 |

float는 말 그대로 요소를 띄우는 속성이다. 이 속성으로 왼쪽 오른쪽으로 위치 시킬 수 있으며 float를 사용하면 특정 콘텐츠가 주변 콘텐츠와 자연스럽게 어울리도록 띄울 수 있다. 문서를 만들 때 이미지를 왼쪽으로 오른쪽으로 글자들이 채워지는 레이아웃을 만들고자 할 때 사용할 수 있다.

```
[ html ]

<p>제1항의 탄핵소추는 국회재적의원 3분의 1 이상의 발의가 있어야 하며, 그 의결은 국회재적의원 과반수의 찬성이 있어야 한다. 다만, 대통령에 대한 탄핵소추는 국회재적의원 과반수의 발의와 국회재적의원 3분의 2 이상의 찬성이 있어야 한다.</p>
<p>국교는 인정되지 아니하며, 종교와 정치는 분리된다. 대통령의 임기는 5년으로 하며, 중임할 수 없다. 모든 국민은 법 앞에 평등하다. 누구든지 성별·종교 또는 사회적 신분에 의하여 정치적·경제적·사회적·문화적 생활의 모든 영역에 있어서 차별을 받지 아니한다.</p>
<p>혼인과 가족생활은 개인의 존엄과 양성의 평등을 기초로 성립되고 유지되어야 하며, 국가는 이를 보장한다. 모든 국민은 근로의 의무를 진다. 국가는 근로의 의무의 내용과 조건을 민주주의원칙에 따라 법률로 정한다.</p>
<p>사회적 특수계급의 제도는 인정되지 아니하며, 어떠한 형태로도 이를 창설할 수 없다. 대통령후보자가 1인일 때에는 그 득표수가 선거권자 총수의 3분의 1 이상이 아니면 대통령으로 당선될 수 없다.</p>
<p>국가는 농업 및 어업을 보호·육성하기 위하여 농·어촌종합개발과 그 지원등 필요한 계획을 수립·시행하여야 한다. 국회는 의장 1인과 부의장 2인을 선출한다.</p>
```

[화면]



제1항의 탄핵소추는 국회재적의원 3분의 1 이상의 발의가 있어야 하며, 그 의결은 국회재적의원 과반수의 찬성이 있어야 한다. 다만, 대통령에 대한 탄핵소추는 국회재적의원 과반수의 발의와 국회재적의원 3분의 2 이상의 찬성이 있어야 한다.

국교는 인정되지 아니하며, 종교와 정치는 분리된다. 대통령의 임기는 5년으로 하며, 중임할 수 없다. 모든 국민은 법 앞에 평등하다. 누구든지 성별·종교 또는 사회적 신분에 의하여 정치적·경제적·사회적·문화적 생활의 모든 영역에 있어서 차별을 받지 아니한다.

혼인과 가족생활은 개인의 존엄과 양성의 평등을 기초로 성립되고 유지되어야 하며, 국가는 이를 보장한다. 모든 국민은 근로의 의무를 진다. 국가는 근로의 의무의 내용과 조건을 민주주의원칙에 따라 법률로 정한다.

사회적 특수계급의 제도는 인정되지 아니하며, 어떠한 형태로도 이를 창설할 수 없다. 대통령후보자가 1인일 때에는 그 득표수가 선거권자 총수의 3분의 1 이상이 아니면 대통령으로 당선될 수 없다.

img 요소에 float으로 왼쪽으로 띄움.

```
[ style ] img { float: left; }
```

[결과]



제1항의 탄핵소추는 국회재적의원 3분의 1 이상의 발의가 있어야 하며, 그 의결은 국회재적의원 과반수의 찬성이 있어야 한다. 다만, 대통령에 대한 탄핵소추는 국회재적의원 과반수의 발의와 국회재적의원 3분의 2 이상의 찬성이 있어야 한다.

국교는 인정되지 아니하며, 종교와 정치는 분리된다. 대통령의 임기는 5년으로 하며, 중임할 수 없다. 모든 국민은 법 앞에 평등하다. 누구든지 성별·종교 또는 사회적 신분에 의하여 정치적·경제적·사회적·문화적 생활의 모든 영역에 있어서 차별을 받지 아니한다.

혼인과 가족생활은 개인의 존엄과 양성의 평등을 기초로 성립되고 유지되어야 하며, 국가는 이를 보장한다. 모든 국민은 근로의 의무를 진다. 국가는 근로의 의무의 내용과 조건을 민주주의원칙에 따라 법률로 정한다.

사회적 특수계급의 제도는 인정되지 아니하며, 어떠한 형태로도 이를 창설할 수 없다. 대통령후보자가 1인일 때에는 그 득표수가 선거권자 총수의 3분의 1 이상이 아니면 대통령으로 당선될 수 없다.

국가는 농업 및 어업을 보호·육성하기 위하여 농·어촌종합개발과 그 지원등 필요한 계획을 수립·시행하여야 한다. 국회는 의장 1인과 부의장 2인을 선출한다.

img 요소에 float으로 오른쪽으로 띄움.

```
[ style ] img { float: right; }
```

[결과]

제1항의 탄핵소추는 국회재적의원 3분의 1 이상의 발의가 있어야 하며, 그 의결은 국회재적의원 과반수의 찬성이 있어야 한다. 다만, 대통령에 대한 탄핵소추는 국회재적의원 과반수의 발의와 국회재적의원 3분의 2 이상의 찬성이 있어야 한다.

국교는 인정되지 아니하며, 종교와 정치는 분리된다. 대통령의 임기는 5년으로 하며, 중임할 수 없다. 모든 국민은 법 앞에 평등하다. 누구든지 성별·종교 또는 사회적 신분에 의하여 정치적·경제적·사회적·문화적 생활의 모든 영역에 있어서 차별을 받지 아니한다.

혼인과 가족생활은 개인의 존엄과 양성의 평등을 기초로 성립되고 유지되어야 하며, 국가는 이를 보장한다. 모든 국민은 근로의 의무를 진다. 국가는 근로의 의무의 내용과 조건을 민주주의원칙에 따라 법률로 정한다.

사회적 특수계급의 제도는 인정되지 아니하며, 어떠한 형태로도 이를 창설할 수 없다. 대통령후보자가 1인일 때에는 그 득표수가 선거권자 총수의 3분의 1 이상이 아니면 대통령으로 당선될 수 없다.

국가는 농업 및 어업을 보호·육성하기 위하여 농·어촌종합개발과 그 지원등 필요한 계획을 수립·시행하여야 한다. 국회는 의장 1인과 부의장 2인을 선출한다.



위 와 같은 문서 형태의 레이아웃을 만들고자 할 때 활용될 수 있다.

clear 속성 - float 속성 초기화

float이 적용되는 것을 원치 않는 요소에 clear : left/right/both 속성을 넣어 초기화 할 수 있다.

[html]

```


<p>제1항의 탄핵소추는 국회재적의원 3분의 1 이상의 발의가 있어야 하며, 그 의결은 국회재적의원 과반수의 찬성이 있어야 한다. 다만, 대통령에 대한 탄핵소추는 국회재적의원 과반수의 발의와 국회재적의원 3분의 2 이상의 찬성이 있어야 한다.</p>

<p style="clear:both">국교는 인정되지 아니하며, 종교와 정치는 분리된다. 대통령의 임기는 5년으로 하며, 중임할 수 없다. 모든 국민은 법 앞에 평등하다. 누구든지 성별·종교 또는 사회적 신분에 의하여 정치적·경제적·사회적·문화적 생활의 모든 영역에 있어서 차별을 받지 아니한다.</p>

<p>혼인과 가족생활은 개인의 존엄과 양성의 평등을 기초로 성립되고 유지되어야 하며, 국가는 이를 보장한다. 모든 국민은 근로의 의무를 진다. 국가는 근로의 의무의 내용과 조건을 민주주의원칙에 따라 법률로 정한다.</p>

<p>사회적 특수계급의 제도는 인정되지 아니하며, 어떠한 형태로도 이를 창설할 수 없다. 대통령후보자가 1인일 때에는 그 득표수가 선거권자 총수의 3분의 1 이상이 아니면 대통령으로 당선될 수 없다.</p>

<p>국가는 농업 및 어업을 보호·육성하기 위하여 농·어촌종합개발과 그 지원등 필요한 계획을 수립·시행하여야 한다. 국회는 의장 1인과 부의장 2인을 선출한다.</p>

```

[style]

```
img { float: left; }
```

두 번째 <p> 요소에 clear : both를 넣어 위에 float 속성을 갖는 요소의 영향을 초기화 시킬 수 있다.

[결과]

제1항의 탄핵소추는 국회재적의원 3분의 1 이상의 발의가 있어야 하며, 그 의결은 국회재적의원 과반수의 찬성이 있어야 한다. 다만, 대통령에 대한 탄핵소추는 국회재적의원 과반수의 발의와 국회재적의원 3분의 2 이상의 찬성이 있어야 한다.



국교는 인정되지 아니하며, 종교와 정치는 분리된다. 대통령의 임기는 5년으로 하며, 중임할 수 없다. 모든 국민은 법 앞에 평등하다. 누구든지 성별·종교 또는 사회적 신분에 의하여 정치적·경제적·사회적·문화적 생활의 모든 영역에 있어서 차별을 받지 아니한다.

혼인과 가족생활은 개인의 존엄과 양성의 평등을 기초로 성립되고 유지되어야 하며, 국가는 이를 보장한다. 모든 국민은 근로의 의무를 진다. 국가는 근로의 의무의 내용과 조건을 민주주의원칙에 따라 법률로 정한다.

사회적 특수계급의 제도는 인정되지 아니하며, 어떠한 형태로도 이를 창설할 수 없다. 대통령후보자가 1인일 때에는 그 득표수가 선거권자 총수의 3분의 1 이상이 아니면 대통령으로 당선될 수 없다.

국가는 농업 및 어업을 보호·육성하기 위하여 농·어촌종합개발과 그 지원등 필요한 계획을 수립·시행하여야 한다. 국회는 의장 1인과 부의장 2인을 선출한다.

float 요소의 부모 요소 변화 살펴보기

float 속성을 갖게 될 때 부모 요소는 자식 요소 중 float 요소는 인식하지 못한다.

[html]

```
<div class="block">
  <span class="inline">인라인 요소</span> <br>
  텍스트
  <p>블록 요소</p>
</div>
```

[style]

```
.container{ border:solid 2px red; }
```

[결과]



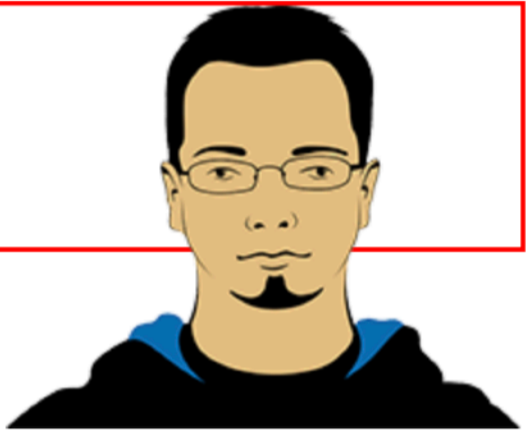
제1항의 탄핵소추는 국회재적의원 3분의 1 이상의 발의가 있어야 하며, 그 의결은 국회재적의원 과반수의 찬성이 있어야 한다. 다만, 대통령에 대한 탄핵소추는 국회재적의원 과반수의 발의와 국회재적의원 3분의 2 이상의 찬성이 있어야 한다.

<div class="container"> 요소 content로 요소와 <p> 요소가 있을 경우<div> 요소의 높이는 자식만큼 늘어난다. 이때, 요소에 float : right 속성을 넣게 되면 이미지는 오른쪽으로 띄어지면서 부모 요소는 요소를 인식 못 하고 <p> 요소만 인식하게 된다.

```
[ style ] img { float: right; }  
.container{ border:solid 2px red; }
```

[결과]

제1항의 탄핵소추는 국회재적의원 3분의 1 이상의 발의가 있어야 하며, 그 의결은 국회재적의원 과반수의 찬성이 있어야 한다. 다만, 대통령에 대한 탄핵소추는 국회재적의원 과반수의 발의와 국회재적의원 3분의 2 이상의 찬성이 있어야 한다.




이때, <div> 요소(부모 요소)가 요소(float 요소) 만큼 높이를 늘려주기 위해서는 요소 밑에 clear (float 초기화) 속성이 적용된 요소를 추가시켜 <div> 요소 (부모 요소)가 초기화된 요소를 인식하게 되어 이미지 아래까지 늘릴 수 있다.

```
[ html ] <div class="container">  
    
  <p>제1항의 탄핵소추는 국회재적의원 3분의 1 이상의 발의가 있어야 하며, 그 의결은 국회재적의원 과반수의 찬성이 있어야 한다. 다만, 대통령에 대한 탄핵소추는 국회재적의원 과반수의 발의와 국회재적의원 3분의 2 이상의 찬성이 있어야 한다.</p>  
  <span style="display: block; clear: both"></span>  
</div>
```

[결과]

제1항의 탄핵소추는 국회재적의원 3분의 1 이상의 발의가 있어야 하며, 그 의결은 국회재적의원 과반수의 찬성이 있어야 한다. 다만, 대통령에 대한 탄핵소추는 국회재적의원 과반수의 발의와 국회재적의원 3분의 2 이상의 찬성이 있어야 한다.



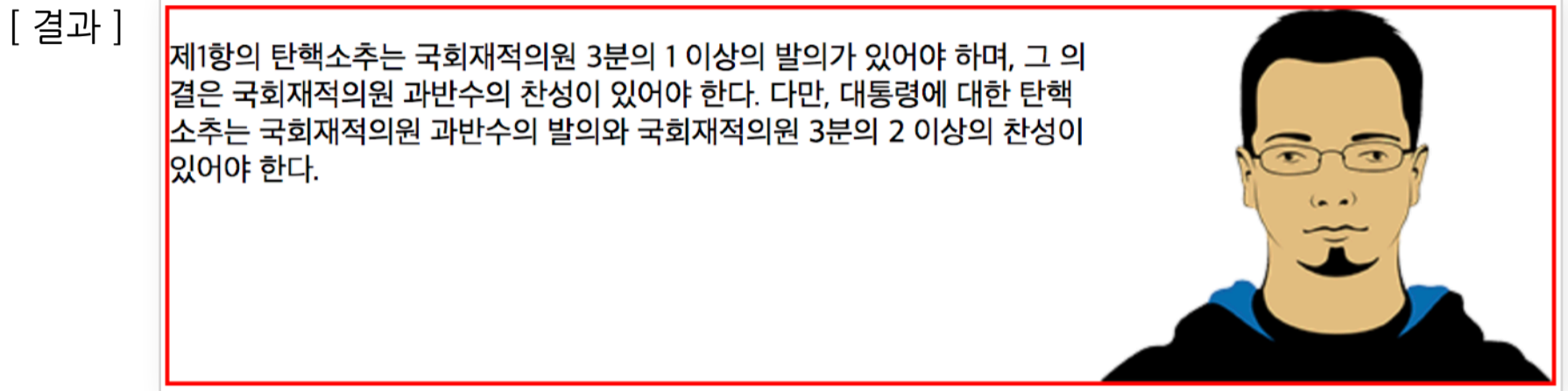
맞춤법 검사를 원하는 단어나 문장을 입력해주세요. 위 내용처럼 요소에 clear 속성을 넣어 주면 <div> 요소는 요소를 인식하여 요소까지 높이가 늘어난다.

하지만 아무 의미 없는 요소를 html에 추가시키지는 건 좋지 않고, 스타일로 요소를 추가시켜 위와 같은 결과를 만들어 낼 수 있다.

스타일 선택자 중 가상 요소 선택자(:before, :after) 활용하여 스타일에서 요소를 추가할 수 있는 float를 content(자식 요소)로 갖는 부모 요소에 :after 가상 요소 선택자를 추가하면 된다.

```
[ html ] <div class="container">
  
  <p>제1항의 탄핵소추는 국회재적의원 3분의 1 이상의 발의가 있어야 하며, 그 의결은 국회재적의원 과반수의 찬성이 있어야 한다. 다만, 대통령에 대한 탄핵소추는 국회재적의원 과반수의 발의와 국회재적의원 3분의 2 이상의 찬성이 있어야 한다.</p>
</div>
```

```
[ style ] img { float: right; }
.container { border:solid 2px red; }
.container:after { content: ''; display: block; clear: both; }
```

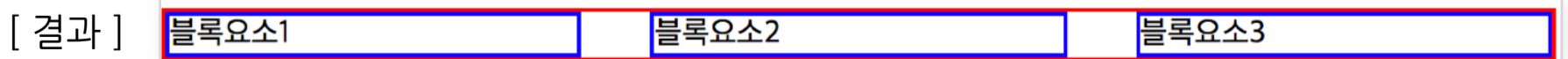


float 속성 요소로 블록요소 옆으로 나열하기

float 속성은 블록 요소, 인라인 요소 모두 적용이 되며 블록요소의 특징 중 아래로 쌓이고 옆으로 나열되지 않기때문에 블록요소를 옆으로 나열하고자 할때 많이 활용된다.

```
[ html ] <div class="container">
  <div class="block1"> 블록요소1 </div>
  <div class="block2"> 블록요소2 </div>
  <div class="block3"> 블록요소3 </div>
</div>
```

```
[ style ] .container { border:solid 2px red; }
.container:after { content: ''; display: block; clear: both; }
.block1, .block2, .block3 {
  box-sizing: border-box;
  width: 30%;
  border:solid 2px blue;
  float: left;
}
.block2 {margin-left: 5%; margin-right: 5%}
```



margin과 padding 값의 %는 부모의 width를 기준으로 백분율을 갖는다.
 .block1, .block2, .block3의 width는 각각 30%이기 때문에 90%에서 margin-left의 5%, margin-right의 5%를 합치면 100%로 부모 요소 크기만큼 갖는다.

background 속성

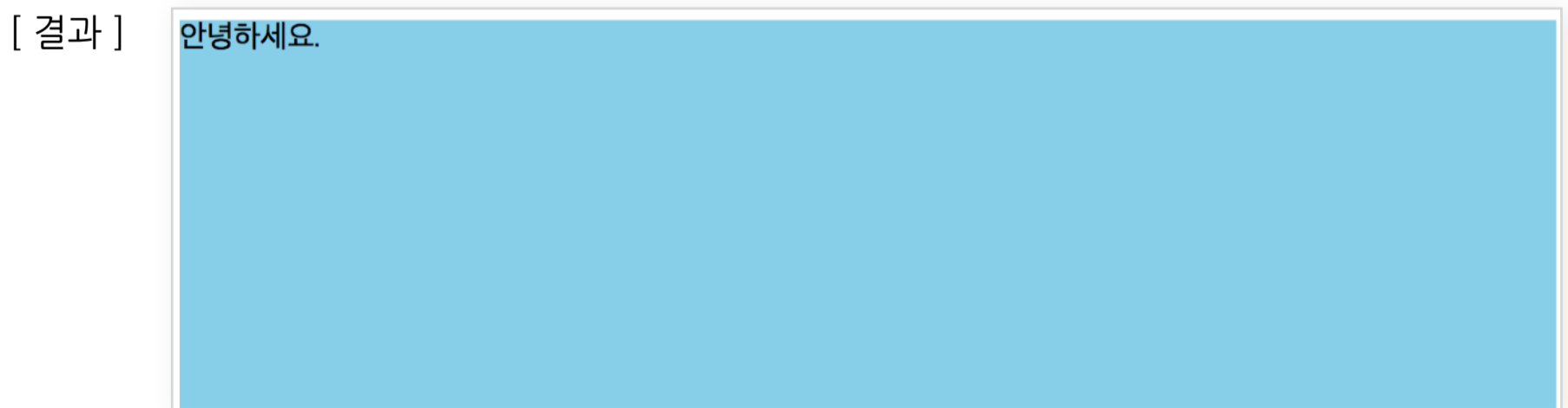
| 속성 | 속성 값 | 설명 |
|-----------------------|---|--|
| background-color | background-color: color transparent initial inherit; | 배경 색 |
| background-image | background-image : url(이미지주소) ; | 배경 이미지 |
| background-repeat | background-repeat: repeat repeat-x repeat-y no-repeat initial inherit; | 배경 이미지 반복 |
| background-position | [background-position: x y;] background-position: value initial ; | 배경 이미지 위치 (% ,px) left top left center left bottom right top right center right bottom center top center center center bottom |
| background-size | [background-size: width height;] background-size: auto length cover contain initial inherit; | 배경 이미지 크기 |
| background-attachment | background-attachment: scroll fixed local initial inherit; | 배경 이미지 스크롤/고정 여부 |
| background-origin | background-origin: padding-box border-box content-box initial inherit; | 배경 이미지 중심 위치 |
| background-clip | background-clip: border-box padding-box content-box initial inherit; | 배경 색 또는 이미지의 요소 내 확장 방법 |

background-color (배경 색)

요소의 배경으로 색을 넣을 수 있다.

```
[ html ] <div class="background">  
  안녕하세요.  
</div>
```

```
[ style ] .background { background-color: skyblue; height: 200px; }
```



background-image (배경 이미지)

요소 배경에 이미지로 채우고자 할 때 사용하게 된다.

background-image를 넣게 되면 이미지 사이즈, 이미지 위치 등 다양한 방법으로 이미지를 설정할 수 있다.

```
[ html ] <div class="background">
  안녕하세요.
</div>
```

```
[ style ] .background {
  background-image: url(image/donut.png);
  height: 200px;
}
```



background-repeat

배경으로 이미지를 넣게 되면 요소의 크기만큼 반복(background-repeat 기본값은 repeat이다.)적으로 이미지가 채워진다.

가로(X축), 세로(Y 축)을 선택하여 반복하거나, 반복적으로 채워지지 않고 한 번만 나오게 할 때는 background-repeat 속성으로 설정할 수 있다.

가로 방향으로 이미지 채우기

```
[ style ] .background {
  background-image: url(image/donut.png);
  background-repeat: repeat-x;
  height: 200px;
  border:solid 2px black;
}
```

[결과]

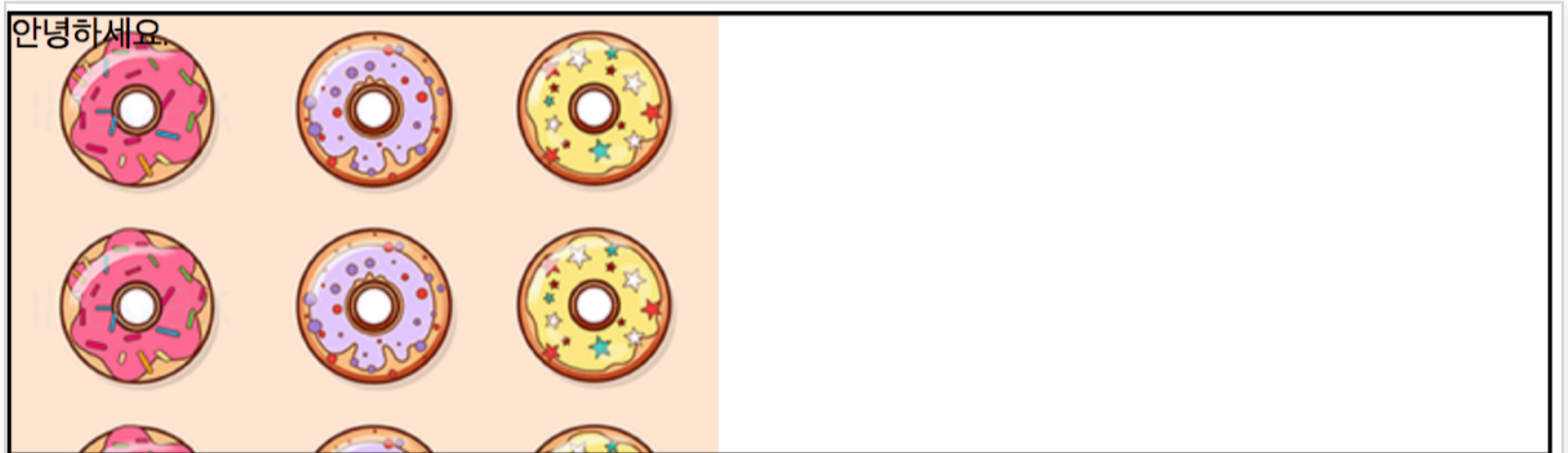


세로 방향으로 이미지 채우기

[style]

```
.background {  
  background-image: url(image/donut.png);  
  background-repeat: repeat-y;  
  height: 200px;  
  border:solid 2px black;  
}
```

[결과]



채우기 없애기

[style]

```
.background {  
  background-image: url(image/donut.png);  
  background-repeat: no-repeat;  
  height: 200px;  
  border:solid 2px black;  
}
```

[결과]



background-position

위 와 같이 요소 안에 배경 이미지가 하나만 나올 경우 이미지의 위치가 요소 안에서 상단 왼쪽에 정렬된다. 이미지의 위치를 변경하고 할 때는 background-position 속성으로 설정할 수 있다.

```
[ style ] .background {  
    background-image: url(image/donut.png);  
    background-repeat: no-repeat;  
    background-position: 50% 100%;  
    height: 200px;  
    border:solid 2px black;  
}
```

[결과] 안녕하세요.



background-position 값으로 두 개의 값이 지정할 수 있다. 첫 번째 값은 x 축이고, 두 번째 값은 y 축이 된다. 하나의 값만 넣게 되면 x축과 y 축 동시에 적용된다.

위의 스타일처럼 %로 넣어 위치를 지정할 수도 있고 left/right/center/top/bottom 값을 넣어 위치를 지정할 수도 있다 고정적으로 넣고자 할 때는 px 단위로 넣을 수 있다.

background-attachment

- scroll : body 스크롤시 선택한 요소와 같이 스크롤되지만, 요소 스크롤시 배경 이미지는 스크롤되지 않는다.
- fixed : body에 이미지가 고정되어 스크롤해도 스크롤되지 않는다.
- local : body 스크롤시 선택한 요소와 같이 스크롤되고, 요소 스크롤시 배경 이미지도 같이 스크롤된다.

[html]

```

<div class="background bg1">
  <ul>
    <li>리스트1</li>
    <li>리스트2</li>
    <li>리스트3</li>
    <li>리스트4</li>
    <li>리스트5</li>
    <li>리스트6</li>
    <li>리스트7</li>
    <li>리스트8</li>
    <li>리스트9</li>
    <li>리스트10</li>
    <li>리스트11</li>
    <li>리스트12</li>
  </ul>
</div>
<div class="background bg2">
  <ul>
    <li>리스트1</li>
    <li>리스트2</li>
    <li>리스트3</li>
    <li>리스트4</li>
    <li>리스트5</li>
    <li>리스트6</li>
    <li>리스트7</li>
    <li>리스트8</li>
    <li>리스트9</li>
    <li>리스트10</li>
    <li>리스트11</li>
    <li>리스트12</li>
  </ul>
</div>
<div class="background bg3">
  <ul>
    <li>리스트1</li>
    <li>리스트2</li>
    <li>리스트3</li>
    <li>리스트4</li>
    <li>리스트5</li>
    <li>리스트6</li>
    <li>리스트7</li>
    <li>리스트8</li>
    <li>리스트9</li>
    <li>리스트10</li>
    <li>리스트11</li>
    <li>리스트12</li>
  </ul>
</div>

```

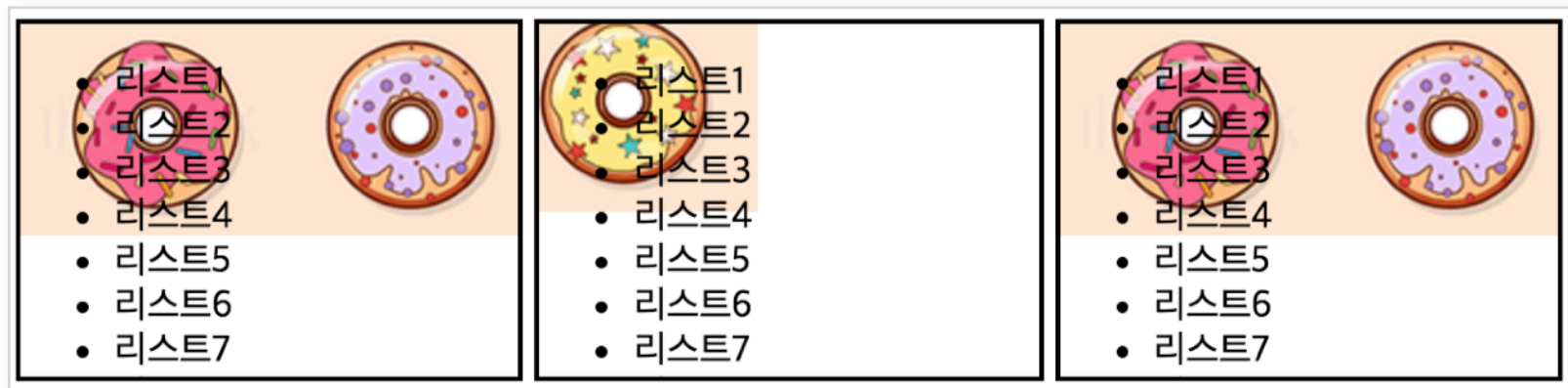
[style]

```

body{
  padding-bottom: 3000px;
}
.background {
  background-image: url(image/donut.png);
  background-repeat: no-repeat;
  background-position: 0 0;
  height: 150px;
  width: 30%;
  overflow: auto;
  float: left;
  margin-right: 5px;
  border:solid 2px black;
}
.bg1{ background-attachment: scroll; }
.bg2{ background-attachment: fixed; }
.bg3{ background-attachment: local; }

```

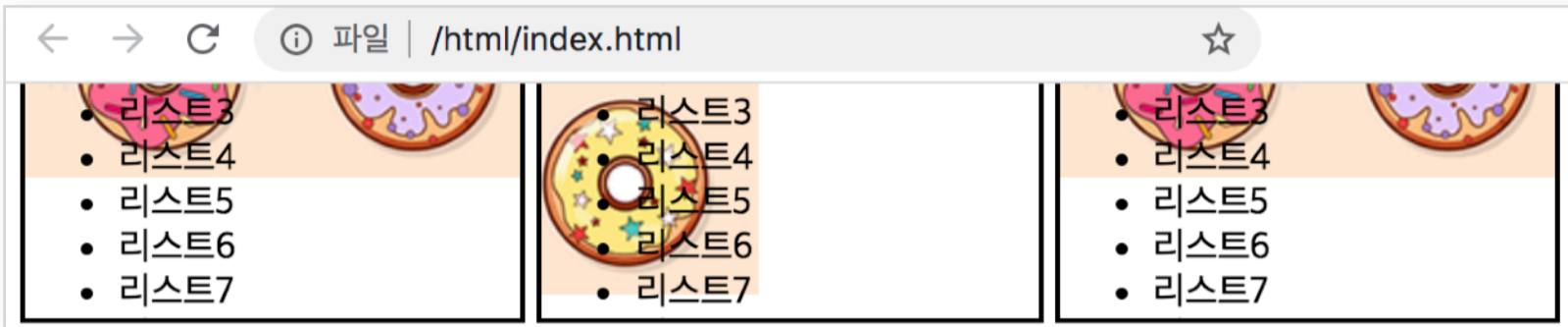
[결과]



각각의 요소 스크롤시 결과



body 스크롤시 결과



overflow

<body> 요소의 내용이 넘어가면 스크롤이 생기듯 특정 크기의 요소의 내용(content)이 오버해도 스크롤이 생성될 수 있다. overflow 속성을 적용시키면 된다.

| 속성 | 속성 값 | 설명 |
|----------|---|-----------------------|
| overflow | overflow: visible scroll auto hidden initial inherit; | 오버된 내용 노출 여부 및 스크롤 여부 |

- visible (기본값) : 오버된 내용이 노출된다.
- hidden : 오버된 내용이 숨겨진다.
- scroll : 오버된 내용은 클리핑 되고, 스크롤바가 생성되어 오버된 내용을 스크롤 할 수 있다.
- auto : scroll과 같이 클리핑 되며 스크롤바가 생성되어 스크롤 하여 내용을 볼 수 있으나, 오버된 내용이 없을 때는 스크롤바가 생성되지 않는다.

background (축약형)

background-color, background-image, background-repeat, background-position, background-attachment 속성은 background 속성이름으로 축약해서 사용할 수 있다.

```
[ html ] <div class="background">
          안녕하세요.
        </div>
```

[style]

```
.background{  
  height: 200px;  
  background: red url(image/donut.png) no-repeat 50% 50% scroll;  
}
```

[결과]



background-size

요소 배경 전체에 채우고 싶거나, width 또는 height 기준으로 채우고자 할 때, 또는 특정 크기로 사이즈를 조절하고자 할 때 사용할 수 있다.

[html]

```
<div class="background">  
  안녕하세요.  
</div>
```

[style]

```
.background{  
  height: 200px;  
  background: url(image/donut.png) no-repeat 50% 50%;  
  background-size: 100% 100%;  
}
```



background-size 값 중 첫번째 100%는 width 값이고 두번째 100% 값은 height 값이된다. 비율 상관없이 .background 요소의 크기만큼 늘어나게 된다.

width기준으로 비율에 맞춰 크기를 늘리거나 줄이고 할때는
background-size : 100%; 또는 background-size : 100% auto; 로 지정해줄 수 있고

height 기준으로 비율에 맞춰 크기를 늘리거나 줄이고자 할때는
background-size: auto 100%; 로 지정해준다.

자동으로 width나 height 기준에 맞춰 이미지를 늘리거나 줄이고자하면
background-size : contain; 또는 background-size : cover; 로 지정해 줄 수 있다.

contain 은 이미지가 잘리지 않게 채워주고, cover는 요소 크기에 꼭 채우고자할때 적용하면된다.

```
[ html ] <div class="background contain">  
    배경 사이즈 contain  
</div>  
<div class="background cover">  
    배경 사이즈 cover  
</div>
```

```
[ style ] .background{  
    height: 100px;  
    background: url(image/donut.png) no-repeat 50% 50%;  
    border:solid 2px red;  
    margin-bottom: 5px;  
}  
  
.contain{  
    background-size: contain;  
}  
  
.cover{  
    background-size: cover;  
}
```



position 관련속성

| 속성 | 속성 값 | 설명 |
|----------|--|---|
| position | position: static absolute fixed relative sticky initial inherit ; | 요소의 정적, 절대 또는 상대 고정 위치 설정 |
| top | top : auto length initial inherit ; | 상단 위치 지정 (absolute, relative, fixed 경우 적용) |
| bottom | bottom : auto length initial inherit ; | 하단 위치 지정 (absolute, relative, fixed 경우 적용) |
| left | left : auto length initial inherit ; | 좌측 위치 지정 (absolute, relative, fixed 경우 적용) |
| right | right : auto length initial inherit ; | 우측 위치 지정 (absolute, relative, fixed 경우 적용) |
| z-index | z-index : auto number initial inherit ; | 배경 이미지 스크롤/고정 여부 |

position 속성

지금까지 요소의 위치는 display 속성 값에 따른 요소들의 특징이 정해지고, margin 또는 padding, float, clear 속성을 통해 레이아웃을 구성하고 위치를 지정해주었다. 또한 html 마크업 순서도 위치 선정에 중요 부분이었었는데, 이는 모든 요소에 position 속성이 static 값으로 지정되어 있을때 위치 선정 방식이었다. 하지만 position 속성의 다른 값들을 알고나면 좀 더 정교하고 쉽게 레이아웃 및 위치를 선정 할 수 있다.

position 속성 값으로는 static(기본값), absolute, relative, fixed가 있다.

- static (정적위치) : position 속성의 기본값이며, html 마크업 출력 위치를 정하는 방식. top/bottom/left/right 속성에는 **영향을 미치지 않는다**.
- relative (상대위치) : 요소의 기본위치(static)를 기준으로 위치 지정.
- absolute (절대위치) : 부모요소(단, static 제외)를 기준으로 위치 지정.
- fixed (절대위치). : 브라우저 창을 기준으로 위치 지정.

top, bottom, left, right 속성

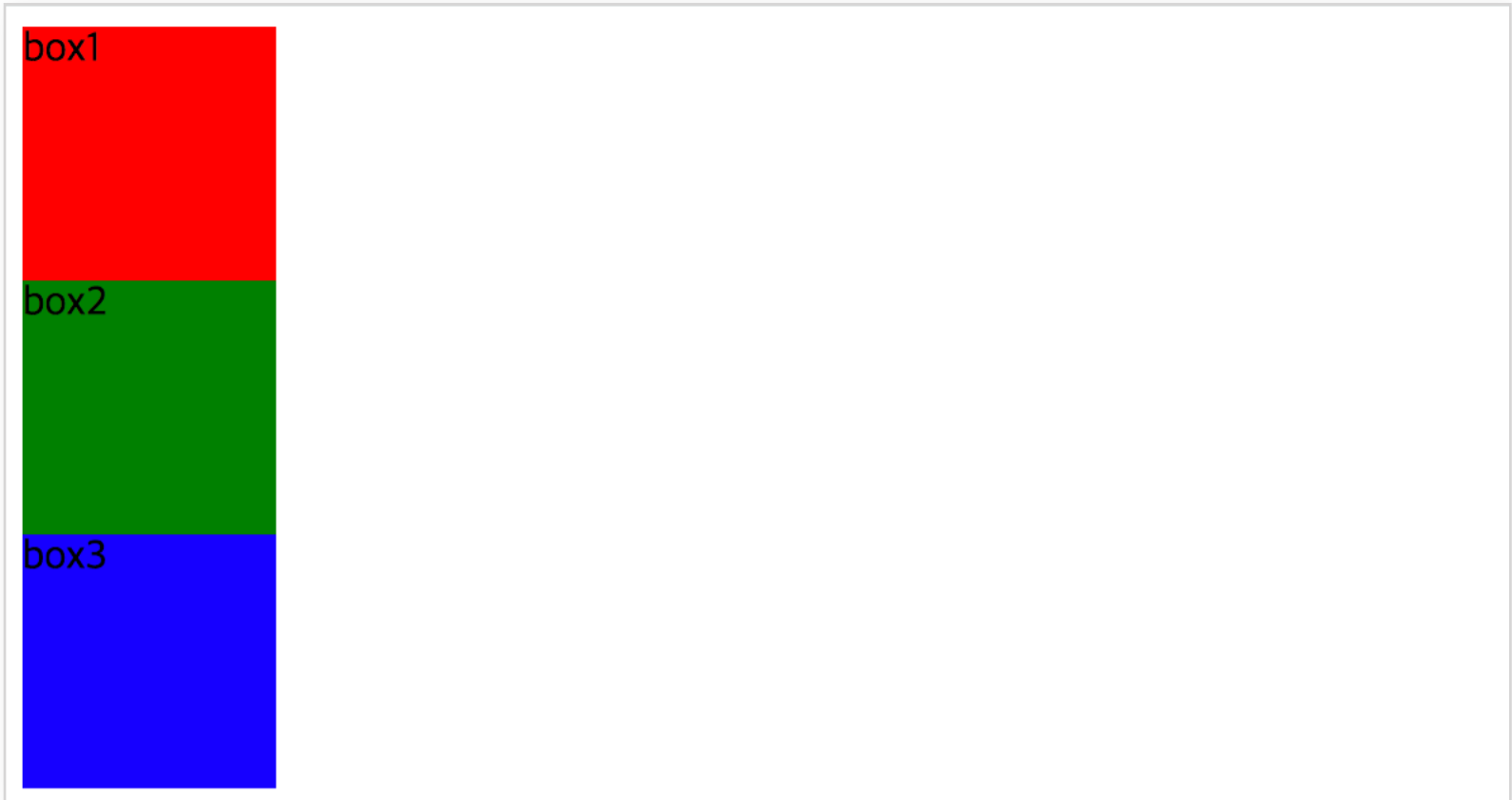
position 속성값이 absolute, relative, fixed, sticky 일 경우 각 속성값에 따른 기준에 따라 상단, 하단, 우측, 좌측 위치를 지정할 수 있다.

static 포지션

```
[ html ] <div class="box1"> box1 </div>
<div class="box2"> box2 </div>
<div class="box3"> box3 </div>
```

```
[ style ] .box1, .box2, .box3 { position: static; width: 100px; height: 100px; }
.box1 { background: red }
.box2 { background: green }
.box3 { background: blue }
```


[결과]



마크업 순서와 display 속성 특징 margin,padding 등의 속성에 따라 위치가 결정된다.
스타일의 position:static 은 생략 가능하다.

relative 포지션

[html]

```
<div class="box1"> box1 </div>  
<div class="box2"> box2 </div>  
<div class="box3"> box3 </div>
```

[style]

```
.box1, .box2, .box3 { position: static; width: 100px; height: 100px; }  
.box1 { background: red }  
.box2 { background: green; position: relative; top: 50px; left: 50px; }  
.box3 { background: blue }
```

[결과]



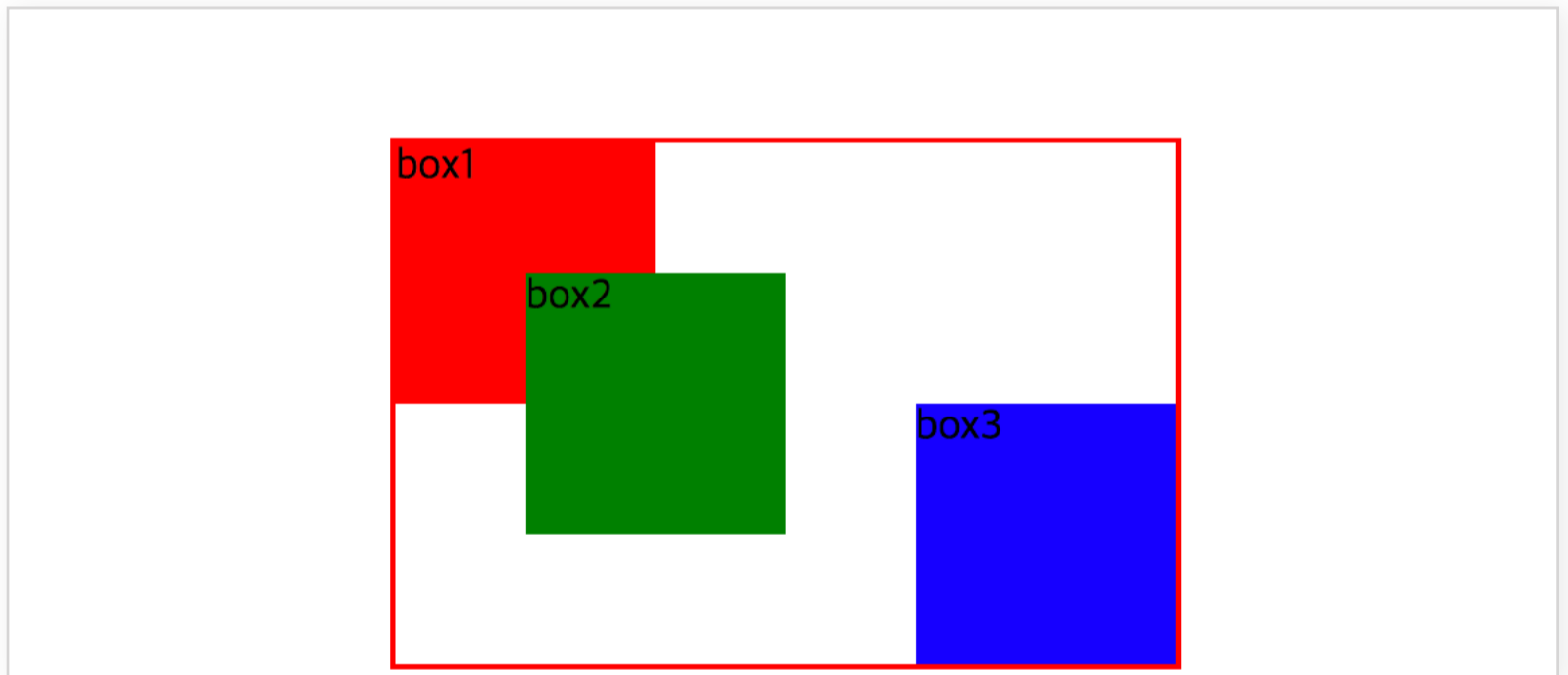
static이었을때 위치가 기준이되며 기본위치(static)에서 아래로 50px(top:50px) 오른쪽으로 50px(left:50px) 이동하였다.

absolute 포지션

```
[ html ] <div class="wrap">
  <div class="box1"> box1 </div>
  <div class="box2"> box2 </div>
  <div class="box3"> box3 </div>
</div>
```

```
[ style ] .wrap {
  position: relative;
  width: 300px;
  height: 200px;
  border:solid 2px red;
  margin: 50px auto 0;
}
.box1, .box2, .box3 { position: absolute; width: 100px; height: 100px;}
.box1 { background: red; top: 0px; left: 0;}
.box2 { background: green; top: 50px; left: 50px;}
.box3 { background: blue; bottom: 0px; right: 0px;}
```

[결과]



absolute 요소인 .box1, .box2, .box3 요소는 부모 요소인 .wrap 요소를 기준으로 각각의 위치 값을 갖는다.

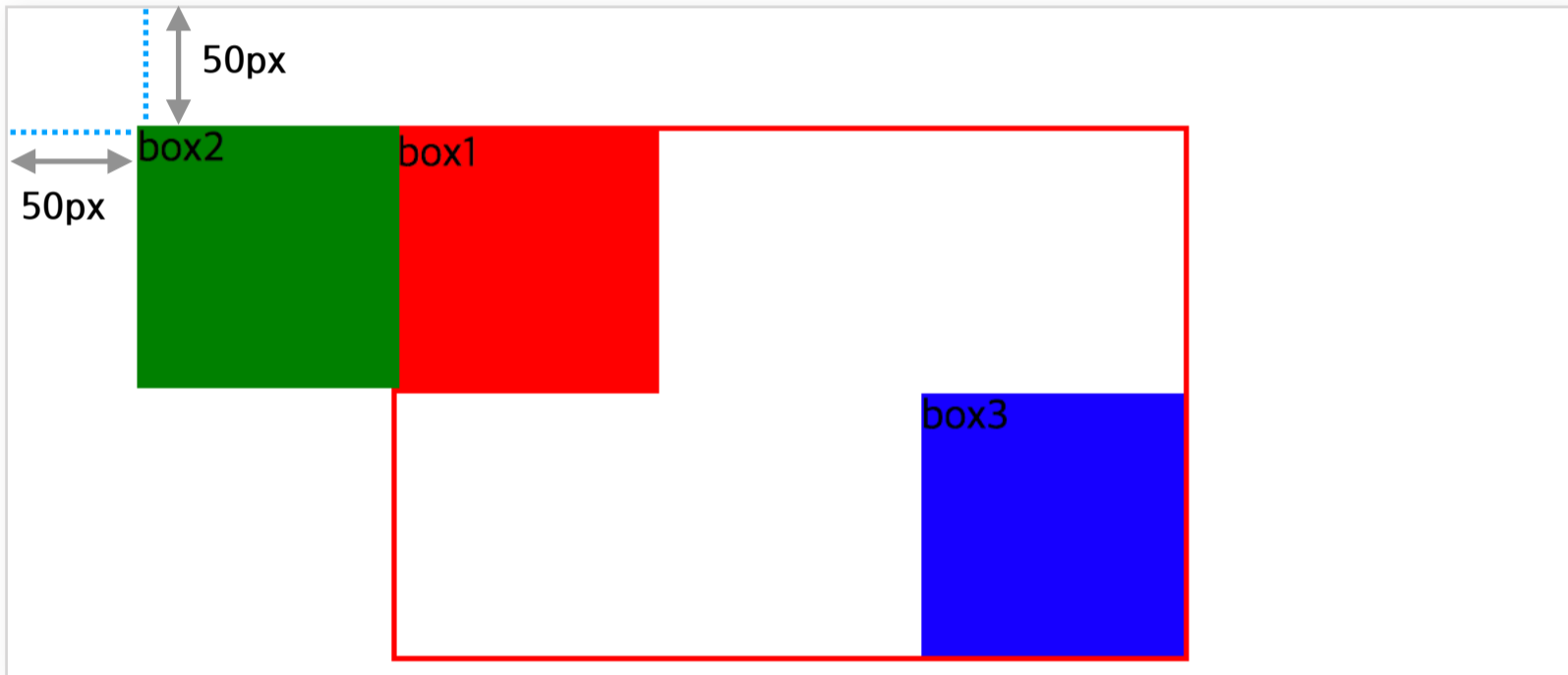
이때 부모 요소인 .wrap 요소의 position이 static 일 경우 부모로 인식하지 못하여 다른 결과가 나온다.

fixed 포지션

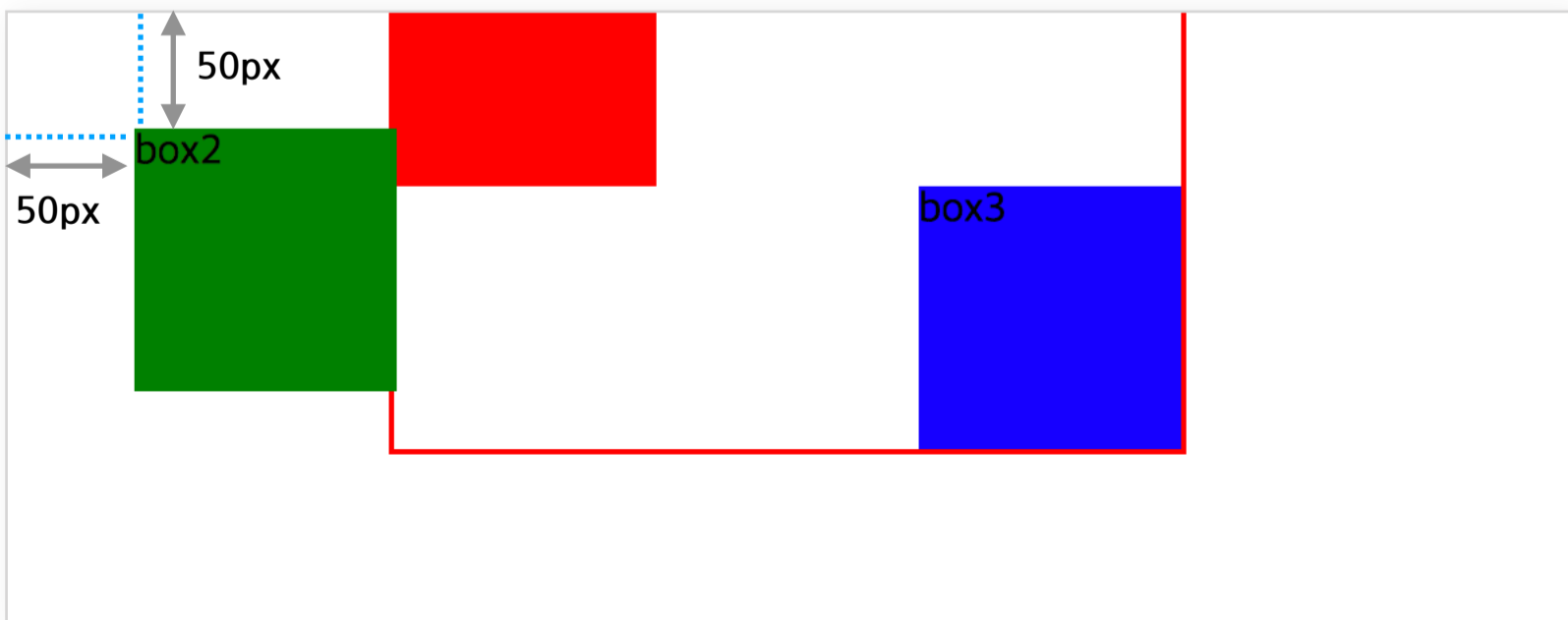
```
[ html ] <div class="wrap">
  <div class="box1"> box1 </div>
  <div class="box2"> box2 </div>
  <div class="box3"> box3 </div>
</div>
```

```
[ style ] .wrap {
  position: relative;
  width: 300px;
  height: 200px;
  border:solid 2px red;
  margin: 50px auto 0;
}
.box1, .box2, .box3 { position: absolute; width: 100px; height: 100px;}
.box1 { background: red; top: 0px; left: 0;}
.box2 { background: green; position: fixed; top: 50px; left: 50px;}
.box3 { background: blue; bottom: 0px; right: 0px;}
```

[결과] 스크롤 내리지 않았을 때



[결과] 스크롤 내렸을 때



.box2 요소만 fixed로 변경했을때 브라우저를 기준으로 위에서 50px 왼쪽에서 50px 이동되었고, 스크롤이 되어도 고정되어 있는것을 확인 할 수 있다.

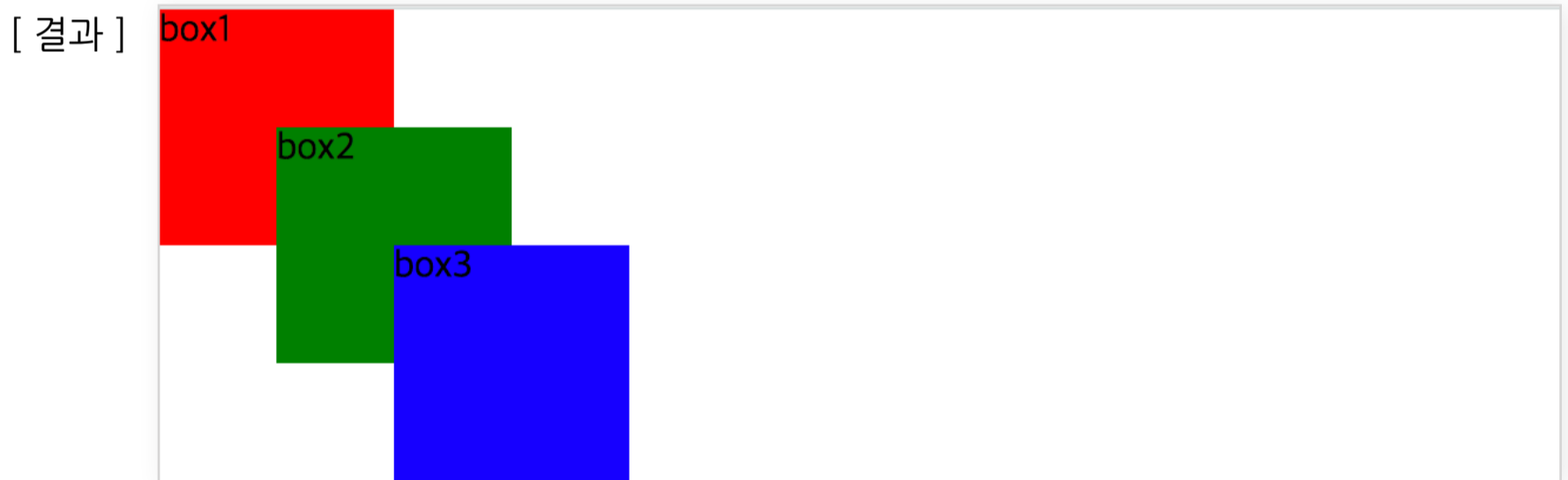
z-index 속성

포지션이 absolute, relative, fixed 일 경우 요소들이 겹치는 경우가 발생한다. 이때 z 위치를 지정 할 수 있다.

기본값은 마크업 순서에 따라 마지막 요소가 위로 올라가게 된다.

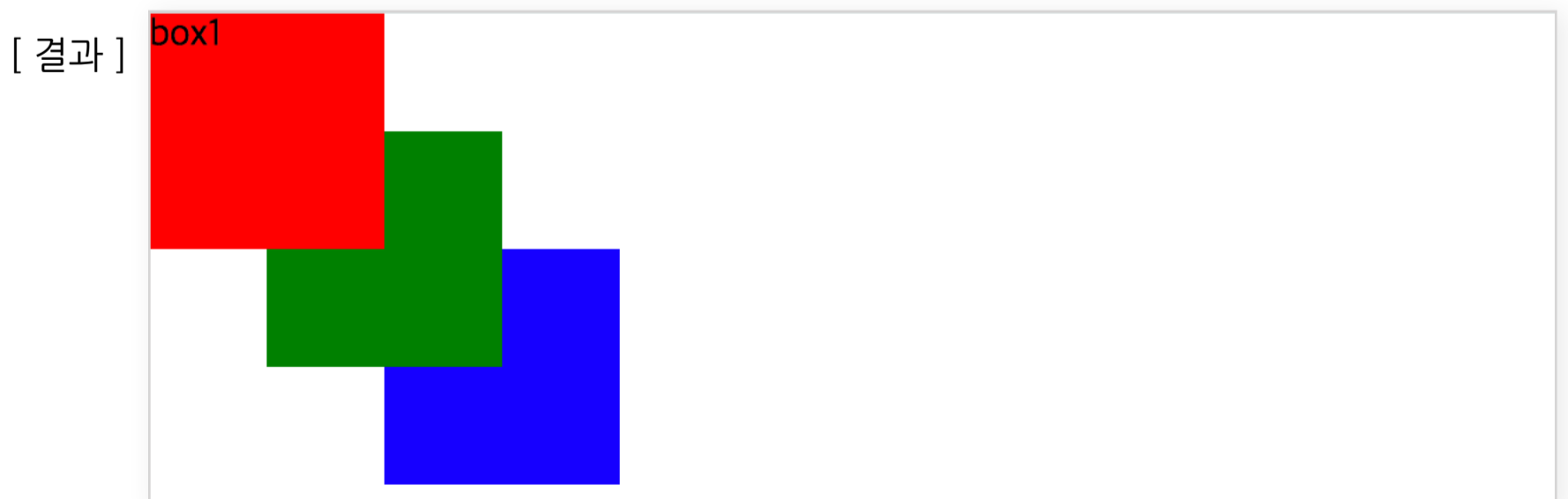
```
[ html ] <div class="box1"> box1 </div>
<div class="box2"> box2 </div>
<div class="box3"> box3 </div>
```

```
[ style ] .box1, .box2, .box3 { position: absolute; width: 100px; height: 100px; }
.box1 { background: red; top: 0px; left: 0; }
.box2 { background: green; top: 50px; left: 50px; }
.box3 { background: blue; top: 100px; left: 100px; }
```



z-index 속성이 없을 경우 기본값에따라 .box3가 맨 위에 .box2가 .box3 뒤에 .box1은 .box2 뒤에 위치하는데 z-index속성 값에 따라 z위치 값을 변경 할 수 있다. 숫자가 클 수록 위에 위치한다.

```
[ style ] .box1, .box2, .box3 { position: absolute; width: 100px; height: 100px; }
.box1 { background: red; top: 0px; left: 0; z-index: 3 }
.box2 { background: green; top: 50px; left: 50px; z-index: 2 }
.box3 { background: blue; top: 100px; left: 100px; z-index: 1 }
```



3장. 활용문제

가. Html 마크업 작성하기

1. 네비게이션 마크업을 완성하시오.

- UI 요소로 메뉴 작성하시오.
- 링크 주소는 임의로 #을 넣어 작성하시오.

메뉴

- [회사소개](#)
 - [인사말](#)
 - [연혁](#)
 - [조직도](#)
- [사업소개](#)
- [소식통](#)
 - [공지사항](#)
 - [자료실](#)
 - [QNA](#)



유틸메뉴

[로그인](#) [회원가입](#) [사이트맵](#)

주메뉴

- [세미나](#)
- [회사소개](#)
- [마이페이지](#)
 - [회원정보관리](#)
 - [예약내역관리](#)
 - [1:1문의내역관리](#)
 - [회원탈퇴](#)
- [고객지원](#)

2. 헤더 영역 마크업을 완성하시오.

- 로고는 사이트의 큰 제목을 의미하는 요소.
- 유틸메뉴 영역
(로그인, 회원가입, 사이트맵은 임의링크 #으로 표현)
- 주메뉴 영역 (navi)
(링크는 #으로 표현)

3. 메인 콘텐츠 중 추천세미나 영역 마크업을 완성하시오.

- 추천세미나 소제목 영역
- 추천세미나 설명부분 영역
- 전체보기/내과/비뇨기과/산부인과 탭메뉴 영역.
- 3개의 추천세미나 목록 영역
(바로가기링크는 임의링크 #으로표현)
- 더보기 버튼으로 링크.
(링크는 임의링크 #으로 표현)

추천세미나

AZATRIP에서 제공하는 학과별 추천 세미나를 확인하실 수 있습니다.

[전체보기](#) [내과](#) [비뇨기과](#) [산부인과](#)



- - 47th Annual Meeting InternationalNeuropsychological Society
 - 장소 : New York City, USA
 - 기간 : 2020.01.01~2020.01.08

[바로예약하기](#)



- - 47th Annual Meeting InternationalNeuropsychological Society
 - 장소 : New York City, USA
 - 기간 : 2020.01.01~2020.01.08

[바로예약하기](#)



- - 47th Annual Meeting InternationalNeuropsychological Society
 - 장소 : New York City, USA
 - 기간 : 2020.01.01~2020.01.08

[바로예약하기](#)

[더보기](#)

4. 푸터 영역 마크업을 완성하시오.

- 주소 / 전화 / 카피라이트 영역
- sns 영역
(페이스북, 트위터, 네이버, 카카오 임의링크 #으로 표현)

서울시 구로구 구로 중앙로 57 남정빌딩 3층
tel : 02-1234-1234
© AZATRIP CO., LTD. All Rights Reserved.
[페이스북](#) [트위터](#) [네이버](#) [카카오](#)

나. CSS로 스타일 적용하기.

1. 네비게이션 스타일 적용하시오. (내부 스타일로 작성하시오.) [sample/sample_01.html] 파일 다운받아 작업하시오.

- reset.css 스타일 파일 적용하기. (스타일 초기화)
- <ul class="navi"> 요소의 가로크기 300px, 테두리 스타일 적용하기.
- 모든 <a> 요소에 밑줄 삭제, 위/아래 여백 10px, 가로크기 300px 스타일 적용하기.
- 1차메뉴 배경색 #333 / 글자색 #FFF
- 2차메뉴 배경색 #FFF / 글자색 #333, 왼쪽으로 20px 여백 지정.



2. Header 섹션을 완성하시오.. (내부 스타일로 작성하시오.) [sample/sample_02.html] 파일 다운받아 작업하시오.



- <header class="header"> 가로사이즈 : 900px, 가운데 정렬, 테두리(색:#ccc)

- <h1 class="logo"> 왼쪽 정렬
- <div class="utilMenu"> 오른쪽 정렬 , 위/오른쪽 20px 여백
- <h2 class="tit"> 화면에 안보이게 숨기기.
- <nav class="navi"> 가로사이즈 100%, 배경색 #efefef
- <nav> 안 옆으로 나열, width 25%, 글자 가운데 정렬
- <nav> 안 <a> 위/아래 여백 15px, 가로사이즈 100%

3. 추천세미나 섹션을 완성하시오.. (내부 스타일로 작성하시오.)
 [sample/sample_03.html] 파일 다운받아 작업하시오.






- <section class="seminar"> 가로사이즈 900px, 가운데 정렬, 테두리(색:#ccc)
- <h2 class="tit"> 글자 가운데 정렬, 기본 폰트사이즈 2배, 글자굵게, 위 여백 50px, 아래 여백 20px
- <p class="tit_comment"> 글자 가운데 정렬, 글자 색:#999, 아래 여백 30px
- <div class="links"> 아래 여백 20px
- <div class="links"> 안 <a> 가로사이즈 25%, 오른쪽 테두리(색:#ccc), 옆으로 나열, 글자 가운데 정렬
- <ul class="list"> 안 옆으로 나열, 가로 사이즈 33.333%, 오른쪽 테두리(색:#ccc), 위/아래/왼쪽/오른쪽 안쪽 여백 20px

- 위/아래 여백 10px, 왼쪽/오른쪽 10px, 배경색:#00f, 글자색 : #fff, 위치 : <section class="seminar"> 기준으로 오른쪽 10px, 위에서 100px 위치로 지정
- <div class="links"> 안 <a> 마우스 오버시 배경색 #fff

추천세미나

AZATRIP에서 제공하는 학과별 추천 세미나를 확인하실 수 있습니다.

[더보기](#)

| 전체보기 | 내과 | 비뇨기과 | 산부인과 |
|--|---|--|------|
|  <p>47th Annual Meeting InternationalNeuropsychological Society 장소 : New York City, USA 기간 : 2020.01.01~2020.01.08 바로예약하기</p> |  <p>47th Annual Meeting InternationalNeuropsychological Society 장소 : New York City, USA 기간 : 2020.01.01~2020.01.08 바로예약하기</p> |  <p>47th Annual Meeting InternationalNeuropsychological Society 장소 : New York City, USA 기간 : 2020.01.01~2020.01.08 바로예약하기</p> | |