
CSS

❖ CSS3 소개

◆ CSS3

- 스타일 시트 표준안
- 웹 문서에 글꼴, 색상, 정렬과 각 요소의 배치 방법 등과 같은 디자인 요소를 적용하는 데 사용

◆ CSS3의 구성

- 선택자(Selector): 스타일 시트를 적용할 대상을 지정
- 속성(Property): 어떤 속성을 적용할지 선택
- 속성값(Value): 속성에 어떤 값을 반영할지 선택

❖ CSS3의 모듈별 발전 과정

◆ CSS1

- 웹 문서의 단순한 글꼴, 텍스트 정렬 방식, 마진 등을 정의하는 데 사용

◆ CSS2

- 1998년에 발표되어 거의 모든 브라우저에서 사용
- 글꼴 규정 및 현재 사용되고 있는 CSS의 모든 규격 등이 이 버전에서 시작됨

◆ CSS3

- Text, fonts, color, backgrounds & borders, transforms, transitions, animations과 같은 종류의 모듈을 추가로 지원
- 기존의 CSS2가 갖지 못했던 화려하고 역동적인 표현을 추가하여 자바스크립트 같은 서버 측 기술에만 의존하던 영역을 지원

❖ CSS3의 필요성

◆ 문서 작성과 디자인을 분리

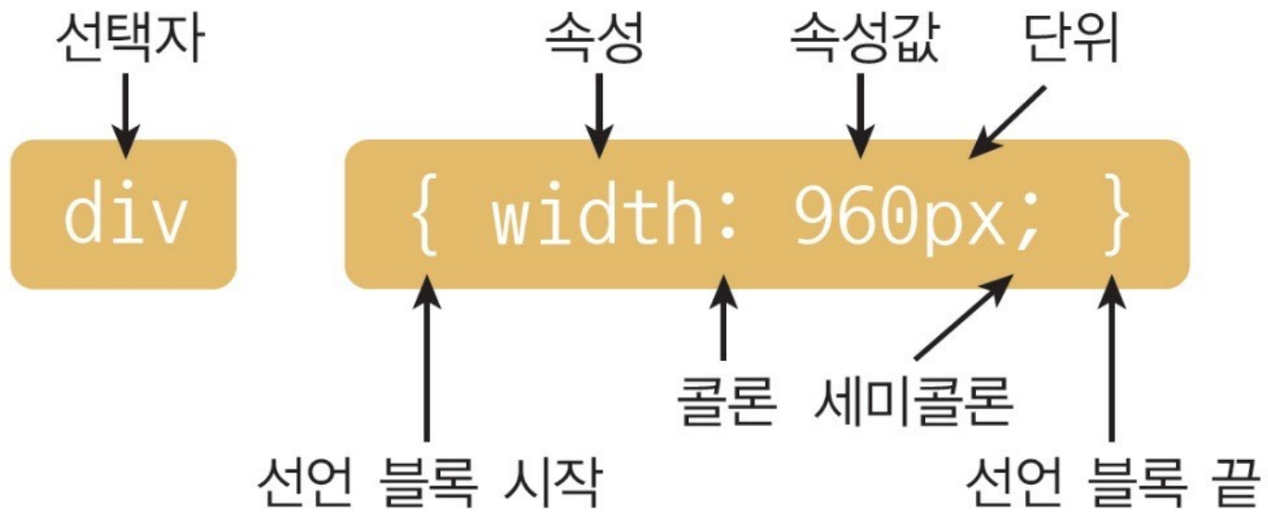
- 하나의 웹 문서에서 문서 작성은 HTML이, 디자인은 CSS가 담당

◆ 디자인을 분리했을 때 장점

- 내용과 디자인 수정이 용이
- 다양한 기능으로 확장 가능
- 통일된 문서 양식 제공
- 전송 및 로딩 시간 단축

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ CSS의 정의 문법



◆ 한줄 작성 방법

```
p { color:blue; background-color:yellow;}
```

◆ 여러줄 작성 방법

```
p {
    color:blue;
    background-color:yellow;
  }
```

❖ CSS의 적용 방법

◆ HTML 문서에 포함 시키는 방법 (Inline Styles)

1. 태그에 넣는 방법

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
</head>
<body>
    <h3 style=“color:blue; background-color:yellow;”>Hello Jennie!!!</h3>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ CSS의 적용 방법

- HTML 문서에 포함 시키는 방법 (Internal Style Sheet)
  - <head>에 넣는 방법

```
<!doctype html>
<html>
<head>
    <meta charset=“UTF-8">
    <style type=“text/css”>
        h3 {
            color:blue;
            background-color:yellow;
        }
    </style>
</head>
<body>
    <h3>Hello Jennie!!!</h3>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ CSS의 적용 방법

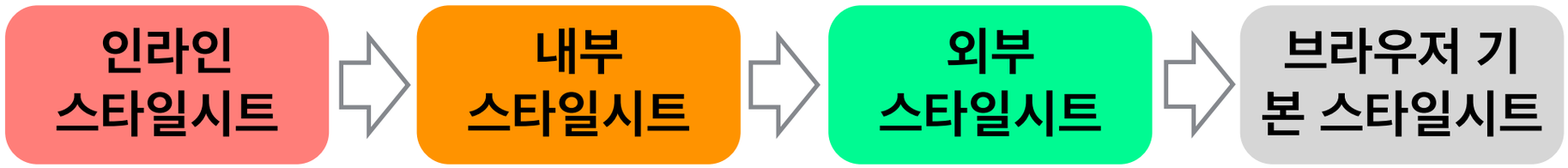
◆ HTML 외부 style sheet 를 사용하는 방법 (External Style Sheet)

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <link type=“text/css” rel=“stylesheet” href=“external_style.css” />
</head>
<body>
    <h3>Hello Jennie!!!</h3>
</body>
```

* HTML5 에서는 CSS를 HTML문서안에 기술하는 것을 권장하지 않는다.
  html 문서는 내용을 기술하고 css에서 스타일을 적용하는 것이 원칙이다.
  하지만 기존버전의 속성이나 기존 페이지의 호환성을 위해서 내부에 포함시킬수 밖에 없다.

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ CSS의 우선순위



- 하나의 요소에 인라인 스타일 시트가 중복 정의되면 제일 마지막에 설정된 값이 적용
- CSS 적용 우선순위와 상관없이 속성을 강제로 적용할 때는 (!important) 표시 사용

❖ CSS의 우선순위

- 외부 style sheet 를 정의하는 위치가 중요

```
p {
  color: green;
  background-color: yellow;
}
```

style01.css

```
<head>
    <meta charset="UTF-8">
    <!-- 외부 스타일 시트를 정의하는 위치가 중요 -->
    <link type="text/css" rel="stylesheet" href=“./css/style01.css” />
    <style>
        p { color: blue; }
        p { color: yellow; }
        p { color: red; }
    </style>
</head>
<body>
    <p>CSS는 정의하는 위치가 중요합니다.</p>
</body>
```

day01_04.html

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ CSS의 우선순위

- 외부 style sheet 를 정의하는 위치가 중요

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <!-- 외부 스타일 시트를 정의하는 위치가 중요 –>
    <style>
        p { color: blue; }
        p { color: yellow; }
        p { color: red; }
    </style>
    <link type="text/css" rel="stylesheet" href=“./css/style01.css” />
</head>
<body>
    <p>CSS는 정의하는 위치가 중요합니다.</p>
</body>
</html>
```

day01_05.html

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ CSS의 우선순위

- !important : 정의된 속성을 무시하고 적용

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <!-- !important —>
    <style>
        p { color: blue !important;}
        p { color: yellow; }
        p { color: red; }
    </style>
    <link type="text/css" rel="stylesheet" href=“./css/style01.css” />
</head>
<body>
    <p>CSS는 정의하는 위치가 중요합니다.</p>
</body>
</html>
```

day01_06.html

❖ 기본 선택자

- 선택자 : 특정한 HTML 태그를 선택할때 사용하는 기능

종류	사용 방법	설명
전체 선택자	* { 속성선언; }	모든 태그에 스타일을 적용한다.
타입 선택자	태그 { 속성선언; }	지정한 태그에 스타일을 적용한다.
클래스 선택자	.클래스 이름 { 속성선언; }	지정한 클래스에 스타일을 적용한다.
아이디 선택자	#아이디 { 속성선언; }	지정한 아이디에 스타일을 적용한다.
속성 선택자	[속성] { 속성선언; } [속성=값] { 속성선언; }	지정한 속성 또는 속성값이 있는 태그에 스타일을 적용한다.

❖ 전체 선택자

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Selector01</title>
    <style>
        * {
            color: red;
            background-color: yellow;
            font-size: 13px;
        }
    </style>
</head>
<body>
    <h1>Universal Selector</h1>
    <h2>모두 같은 색상, 같은 크기</h2>
    <h3>전체적으로 동시에 스타일 적용</h3>
    <p>모든 데이터에 적용</p>
```

selector02.html

❖ 타입 선택자

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Selector02</title>
    <style>
        h1 { background-color: yellow; }
        h2 { background-color: green; }
        h3 { background-color: pink; }
        p { color: red; }
    </style>
</head>
<body>
    <h1>Type Selector</h1>
    <h2>Type Selector</h2>
    <h3>Type Selector</h3>
    <p>각 요소에 다르게 적용</p>
</body>
</html>
```

selector02.html

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 클래스 선택자

selector03.html

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Selector03</title>
    <style>
        .class1 {
            color: blue;
            background-color: yellow;
        }
        .class2 {
            color: red;
            background-color: green;
        }
        h3.class1 {
            color: navy;
            background-color: pink;
        }
    </style>
</head>
<body>
    <h1 class="class1">Class 1 입니다.</h1>
    <p class="class1">Class 1 입니다.</p>
    <h1 class="class2">Class 2 입니다.</h1>
    <p class="class2">Class 2 입니다.</p>
    <h3 class="class1">Element+Class Selector</h3>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ ID 선택자

selector04.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8">
  <title>Selector04</title>
  <style>
        #id1 {
            color: blue;
            background-color: pink;
        }
        #id2 {
            color: blue;
            background-color: yellow;
        }
        h2#id1 {
            color: red;
            background-color: green;
        }
   </style>
</head>
<body>
    <h1 id="id1">ID1 선택자</h1>
    <p id="id1">ID1 선택자</p>
    <h1 id="id2">ID2 선택자</h1>
    <p id="id2">ID2 선택자</p>
    <h2 id="id1">Element+ID Selector</h2>
</body>
```

❖ 속성 선택자

selector05.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8”>
  <title>Selector05</title>
  <style type=“text/css”>
        input[type=“text”] {
            background-color: red;
        }
        input[type=“password”]{
            background-color:blue;
        }
  </style>
</head>
<body>
    <input type=“text” />
    <input type="password" />
</body>
```

❖ 가상 선택자

- 웹 문서에는 보이지 않지만 동작에 영향을 주는 속성을 가상 선택자로 이용

```
a:link { color: blue; }
```

가상 선택자

사용 방법	설명	사용 예
: link 선택자	웹 문서에서 사용자가 방문하지 않았던 곳을 표시한다.	a : link { color: red; text-decoration: none; }
: visited 선택자	웹 문서에서 사용자가 방문한 곳을 표시한다.	a : visited { color: blue; }
: active 선택자	웹 문서에서 사용자가 링크를 클릭하는 순간을 나타낸다.	a : active { color: black; }
: hover 선택자	웹 문서에서 사용자가 링크에 마우스 포인터를 올리는 순간을 나타낸다.	a : hover { color: green; }
: focus 선택자	해당 요소에 초점이 맞춰졌을 때 적용된다.	a : focus { color: yellow; }

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 가상 선택자

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Selector06</title>
    <style>
        a:link { color: blue; text-decoration: underline; }
        a:visited { color: red; }
        a:hover { text-decoration: overline; }
        a:active { background-color: yellow; }
        div.d1 { border: 1px dashed red; width: 400px; padding: 5px; }
        div.d1:hover { background-color: yellow; }
        div.d2 { border: 1px dashed green; width: 400px; padding: 5px; }
        div.d2:hover { background-color: green; }
    </style>
</head>
<body>
    <h2>가상선택자</h2>
    <p><a href="http://www.w3.org" target="_blink">W3C 방문</a> : 마우스 이벤트에 따른 링크의 변화를 잘 보세요.</p>
    <div class="d1">
        <h3>가상 클래스 1 영역</h3>
        마우스 위치에 따른 박스의 스타일 변화를 보세요.
    </div>
    <div class="d2">
        <h3>가상 클래스 2 영역</h3>
        마우스 위치에 따른 박스의 스타일 변화를 보세요.
    </div>
</body>
</html>
```

selector06.html

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 이벤트 가상 클래스 선택자

◆ 반응 선택자 : 사용자의 반응으로 생성되는 특정한 상태를 선택

선택자 형태	설 명
:active	사용자가 마우스로 클릭한 태그 선택
:hover	사용자가 마우스 커서를 올린 태그 선택

◆ 상태 선택자 : 입력 양식의 상태를 선택할 때 사용

선택자 형태	설 명
:checked	체크 상태의 input 태그 선택
:focus	초점이 맞추어진 input 태그 선택
:enabled	사용 가능한 input 태그 선택
:disabled	사용 불가능한 input 태그 선택

❖ 반응 선택자

selector07.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Selector07</title>
  <style type="text/css">
        h1:hover { color: red; }
        h1:active { color: blue; }
 </style>
</head>
<body>
    <h1>반응 선택자</h1>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 상태 선택자

selector08.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8”>
  <title>Selector08</title>
  <style>
        /* input 태그가 사용 가능할 경우에
           background-color 속성에 white 키워드를 적용합니다. */
        input:enabled { background-color: white; }

        /* input 태그가 사용 불가능할 경우에
           background-color 속성에 gray 키워드를 적용합니다. */
        input:disabled { background-color: gray; }

        /* input 태그에 초점이 맞추어진 경우에
           background-color 속성에 orange 키워드를 적용합니다. */
        input:focus { background-color: orange; }
    </style>
</head>
<body>
    <h2>사용 가능</h2>
    <input />
    <h2>사용 불가능</h2>
    <input disabled="disabled"/>
</body>
</html>
```

❖ 이벤트 가상 클래스 선택자

◆ 상태 선택자 : 입력 양식의 상태를 선택할 때 사용

선택자 형태	설 명
:first-child	형제 관계에서 첫 번째로 등장하는 태그 선택
:last-child	형제 관계에서 마지막으로 등장하는 태그 선택
:nth-child(수열)	형제 관계에서 앞에서 수열 번째로 등장하는 태그 선택
:nth-last-child(수열)	형제 관계에서 뒤에서 수열 번째로 등장하는 태그 선택

* **:nth-child(수열)**
  수열에 2n+1 을 넣으면 홀수 2n을 넣으면 짝수 번째 위치의 태그의 스타일을 적용

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 구조 선택자

selector09.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Selector09</title>
  <style>
        ul { overflow: hidden; }
        li {
            list-style: none;
            float:left; padding: 15px;
        }
        li:first-child { border-radius: 10px 0 0 10px; }
        li:last-child { border-radius: 0 10px 10px 0; }
        li:nth-child(2n) { background-color: #FF0003; }
        li:nth-child(2n+1) { background-color:#800000; }
    </style>
</head>
<body>
    <ul>
        <li>첫 번째</li>
        <li>두 번째</li>
        <li>세 번째</li>
        <li>네 번째</li>
        <li>다섯 번째</li>
        <li>여섯 번째</li>
        <li>일곱 번째</li>
    </ul>
</body>
</html>
```

❖ 구조 선택자

selector10.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8”>
  <title>Selector10</title>
  <style>
        li > a:first-child { color: red; }
    </style>
</head>
<body>
    <ul>
        <li><a href="#">주의 사항</a></li>
        <li><a href="#">주의 사항</a></li>
        <li><a href="#">주의 사항</a></li>
        <li><a href="#">주의 사항</a></li>
        <li><a href="#">주의 사항</a></li>
    </ul>
</body>
</html>
```

❖ 구조 선택자

selector10_01.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Selector10_01</title>
  <style>
      li:first-child > a { color: red; }
  </style>
</head>
<body>
    <ul>
        <li><a href="#">주의 사항</a></li>
        <li><a href="#">주의 사항</a></li>
        <li><a href="#">주의 사항</a></li>
        <li><a href="#">주의 사항</a></li>
        <li><a href="#">주의 사항</a></li>
    </ul>
</body>
</html>
```

❖ 조합 선택자

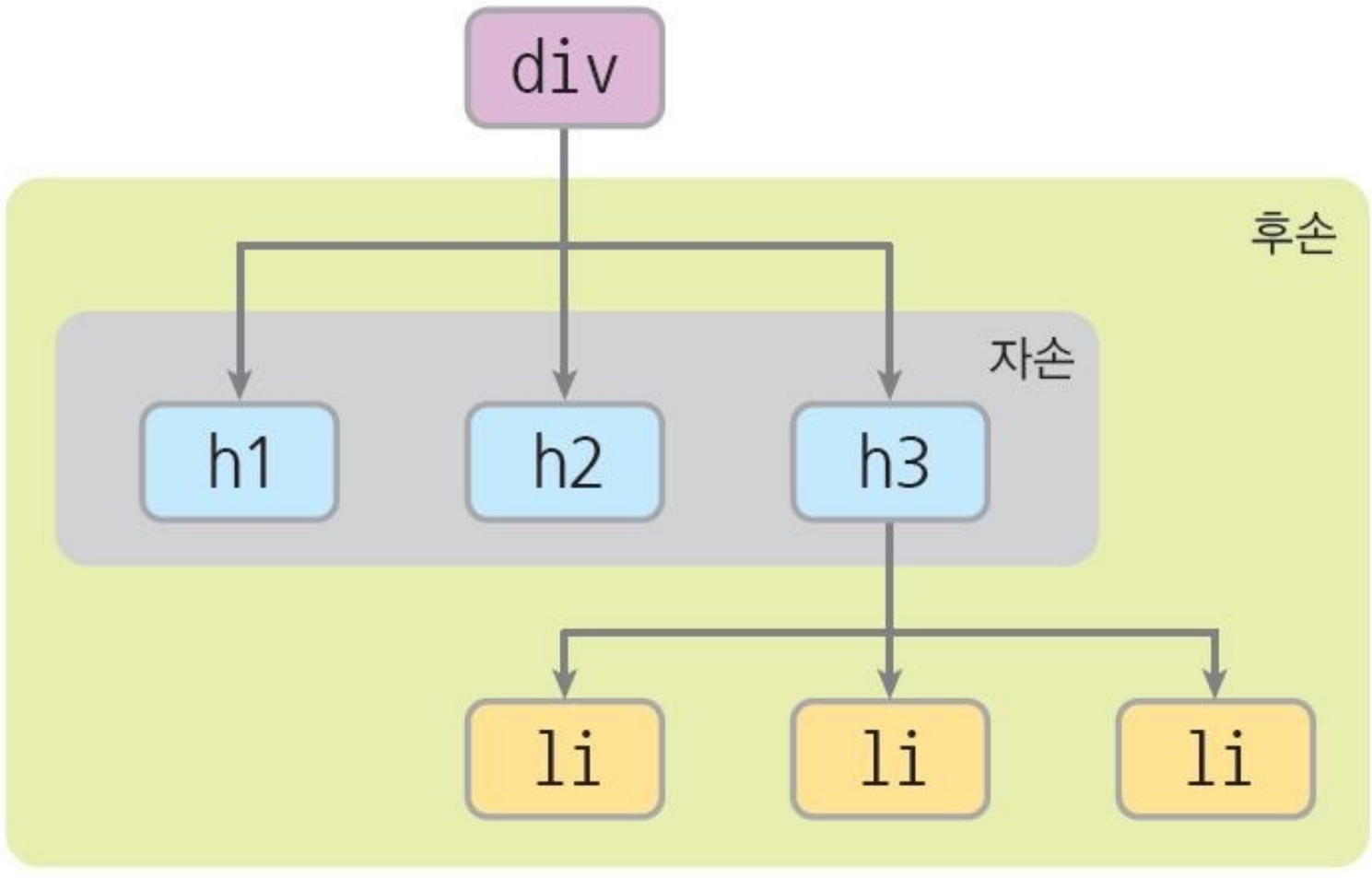
◆ 기존의 여러 선택자를 복합적으로 조합하는 방법 제공

구 분	조합 방법	설 명
후손 선택자	선택자A 선택자B	선택자B가 선택자A에 반드시 포함되어 있을 경우 선택
자손 선택자	선택자A > 선택자B	부모 선택자A의 직계 자손인 선택자B를 선택한다.
인접 형제 선택자	선택자A + 선택자B	선택자A 바로 다음에 위치한 선택자B를 선택한다.
일반 형제 선택자	선택자A ~ 선택자B	선택자A 뒤에 인접하여 나타나는 모든 선택자B를 선택한다.
그룹 선택자	선택자A, 선택자B	선택자A와 선택자B를 모두 선택한다.

조합 선택자의 종류

❖ 조합 선택자

◆ 조합 선택자의 이해



HTML 문서 트리구조

❖ 조합 선택자

selector11.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Selector11</title>
  <style>
        div ul { color: red; }
        div h2 { color: yellow;
                          background-color: purple; }
    </style>
</head>
<body>
    <div>
        <h2>Descendant Selector_1</h2>
                <ul>후손 선택자
                        <li>자식의 자식 (후손1)</li>
                        <li>자식의 자식 (후손2)</li>
         </ul>
    </div>
    <h2>Descendant Selector_2</h2>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 조합 선택자

selector12.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Selector12</title>
  <style>
        body > div > h3 { color: red; }
        body > p { color: green; background-color: yellow; }
        body > h3 > tel > home { color: blue; }
    </style>
</head>
<body>
    <div>
        <h3>Child Selector_1</h3>
    </div>
    <p>자식 선택자</p>
    <h3>
      <sno>123456</sno><br>
      <std>제니</std><br<
      <tel>
          <office>02-4567-1010</office><br>
          <home>010-1234-5678</home>
       </tel>
    </h3>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 조합 선택자

selector13.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Selector13</title>
  <style>
        h1 + h2 + ul { color: blue; }
        div + h3 { color: red; }
        h3 + p { color: purple; background-color: yellow; }
    </style>
</head>
<body>
    <div>
        <h1>인접 형제 선택자1</h1>
        <h2>인접 형제 선택자2</h2>
        <ul>목록
            <li>주제1</li>
            <li>주제2</li>
        </ul>
    </div>
    <h3>Adjacent Selector_1</h3>
    <p>인접 형제 선택자에 의한 스타일 적용</p>
    <h3>Adjacent Selector_2</h3>
</body>
</html>
```

❖ 조합 선택자

selector14.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Selector14</title>
  <style>
        h1 ~ ul { color: blue; }
        div ~ h3 { color: red; }
        h3 ~ p { color: purple; background-color: yellow; }
    </style>
</head>
<body>
    <div>
        <h1>형제 선택자1</h1>
        <h2>형제 선택자2</h2>
        <ul>목록
            <li>주제1</li>
            <li>주제2</li>
        </ul>
    </div>
    <h3>Sibling Selector-1</h3>
    <h4>같은 레벨 형제</h4>
    <p>일반 형제 선택자에 의한 스타일 적용</p>
    <h3>Sibling Selector_2</h3>
    <h3>Sibling Selector_3</h3>
</body>
```

❖ 그룹 선택자

```
h1 { font-family: D2Coding;}
h2 { font-family: D2Coding;}
h3 { font-family: D2Coding;}
```

```
h1, h2, h3 {
        font-family: D2Coding;
}
```

selector15.html

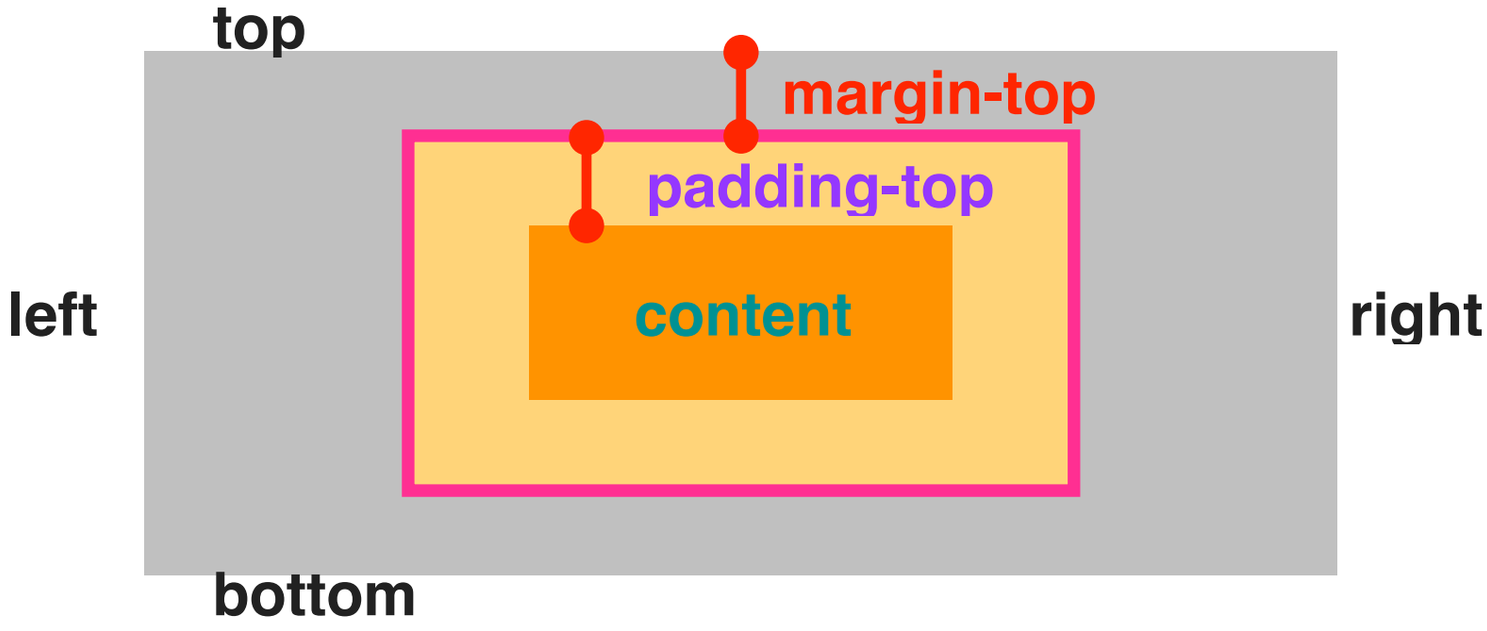
```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Selector15</title>
  <head>
    <style>
        h3, p, h2 {
                color: red;
                background-color: yellow;
        }
    </style>
</head>
<body>
    <h2>Group Selector_1</h2>
    <p>스타일 지정을 그룹으로 적용</p>
    <div>
        <h3>Group Selector_2</h3>
    </div>
</body>
</html>
```

❖ 박스 모델

- 웹 문서에 텍스트, 이미지, 테이블 등의 요소를 배치하기 위해 사용
- 웹 문서의 전체 레이아웃을 정의
- 각종 요소들을 원하는 위치에 배치

❖ 박스의 속성

- **content : 실제 내용이 표현되는 곳**
- **padding : 콘텐츠와 테두리 사이의 여백**
- **border : 박스의 테두리 두께**
- **margin : 테두리와 박스의 최종 경계 사이의 여백**



훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 박스 모델

box01.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Box Model</title>
  <style>
        div {
            background-color: yellow;
            width: 300px;
            padding: 25px;
            border: 15px solid navy;
            margin: 25px;
        }
    </style>
</head>
<body>
    <p>CSS3 박스 모델은 content, padding, border, margin으로 구성되어 있다.</p>
    <div>박스 모델의 padding, border, margin 속성의 기본 값은 0이며, 상하좌우 네 가지 방향을 top, bottom, left,
    right를 이용하여 지정할 수 있습니다.</div>
</body>
</html>
```

❖ Content

- 실제 내용이 표현되는 곳
- 속성: width(너비 지정), height(높이 지정)

속성	설명
value	실제로 측정한 데이터 값이다.
min, max	데이터가 인식하는 최솟값과 최댓값이다. 기본값은 0~1이다.
low, high	허용되는 범위의 최솟값과 최댓값이다. low~high 값은 항상 min~max 값 범위 내에 있다.

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Content

content01.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8”>
  <title>Box Model</title>
  <style>
    p {
        background-color: yellow;
        color: red;
        font-weight: bold;
        font-size: 16pt;
    }
    p.c1 {
        width: 700px;
        height: 80px;
        color: green;
    }
    p.c2 {
        width: 50%;
        height: 150%;
        color: purple;
    }
    p.c3 {
        width: 10cm;
        height: 3cm;
        color: blue;
    }
</style>
</head>
<body>
    <p>박스 모델의 내용 영역 크기 지정</p>
    <p class="c1">(1) 박스 모델의 크기를 픽셀(px) 단위로 지정</p>
    <p class="c2">(2) 박스 모델의 크기를 퍼센트(%) 단위로 지정</p>
    <p class="c3">(3) 박스 모델의 크기를 센티미터(cm) 단위로 지정</p>
</body>
```

❖ padding, margin

- padding : 박스의 안쪽 여백 설정
- Margin : 박스의 바깥쪽 여백 설정

속성값	설 명
수 치	여백을 픽셀(px), 포인트(pt), 센티미터(cm), 배수(em) 같은 단위로 지정
백분율	여백을 부모 요소를 기준으로 하여 백분율로 지정
auto	여백을 웹 브라우저가 자동으로 지정 / 기본값

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Content

content01.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8”>
  <title>Content</title>
  <style>
        p {
           background-color: yellow;
           color: red;
           font-weight: bold;
           font-size: 16pt;
        }
        p.pad {
           color: purple;
           padding: 20px;
        }
        p.mar {
           color: green;
           margin: 30px;
        }
        p.mp {
           color: blue;
           padding: 5%;
           margin: 5%;
        }
    </style>
</head>
<body>
    <p>박스의 안쪽 여백과 바깥쪽 여백 지정</p>
    <p class="pad">(1) 안쪽 여백 지정 - padding 20px</p>
    <p class="mar">(2) 바깥쪽 여백 지정 - margin 30px</p>
    <p class="mp">(3) 안쪽, 바깥쪽 여백 지정 - padding 5%, margin 5%</p>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 속성값의 개수에 따라 적용되는 위치

```
h1 { margin: 5px 10px 5px 10px; }
```

top right bottom left

```
h1 { margin: 5px 10px 5px ; }
```

top right & left bottom

```
h1 { margin: 5px 10px; }
```

top & bottom right & left

```
h1 { margin: 5px; }
```

top & bottom & right & left

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Content

content02.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8”>
  <title>Content02</title>
  <style>
       body { font-weight: bold;  font-size: 16pt; }
       .mp1 { background-color: aqua;
                    color: red; margin-bottom: 40px;
                    padding-left: 50px; }
       .mp2 { background-color: silver;  color: green;
                    margin: 20px;  padding: 10px, 20px;  }
       .mp3 { background-color: gray;  color: purple;
                    margin: 50px, 15px, 20px; padding: 20px; }
    </style>
</head>
<body style="background-color: lightyellow;">
    <p>박스 모델의 네 방향 여백 지정</p>
    <p class="mp1">mp1</p>
    <p class="mp2">mp2</p>
    <p class="mp3">mp3</p>
</body>
</html>
```

❖ border-width

- 테두리 두께 설정
- Top, bottom, left, right를 이용하여 네 방향으로 설정 가능

속성값	설 명
수 치	여백을 픽셀(px), 포인트(pt), 센티미터(cm) 같은 단위로 지정
thin	얇은(1px) 두께의 테두리 지정
medium	중간(3px) 두께의 테두리 지정
thick	굵은(5px) 두께의 테두리 지정

❖ border-color

◆ 테두리 색상 지정 (W3C의 칼라 코드 표 참고)

❖ border-style

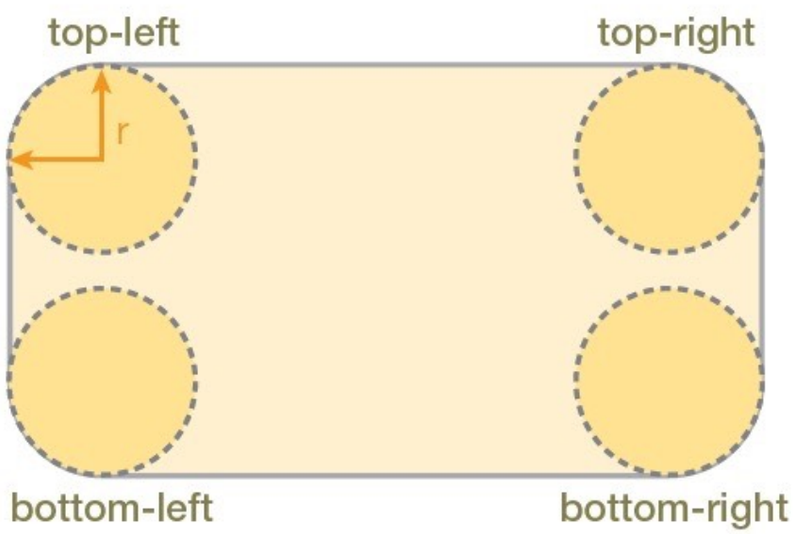
- 테두리 스타일 지정

속성값	설 명
none	테두리가 나타나지 않는다. 기본값
hidden	테두리를 감춘다.
dotted	테두리를 점선으로 지정
dashed	테두리를 파선으로 지정
solid	테두리를 실선으로 지정
double	테두리를 이중선으로 지정
groove	테두리를 오목한 선(홈이 파인듯 입체적으로) 지정
ridge	테두리를 볼록한 선(튀어 나온듯 입체적으로) 지정
inset	테두리 안쪽이 오목한 선으로 지정
outset	테두리 안쪽이 볼록한 선으로 지정

❖ border-radius

- 테두리의 모서리를 둥글게 설정

형 식	설 명
none	테두리가 나타나지 않는다. 기본값
hidden	테두리를 감춘다.
dotted	테두리를 점선으로 지정
dashed	테두리를 파선으로 지정
solid	테두리를 실선으로 지정



❖ Border

border01.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Border01</title>
  <style>
       body { font-weight: bold; font-size: 12pt; }
      .br1 { background-color: lime; color: red;
          border-style: solid; border-width: thick;
          border-color: green; border-radius: 30%; }
      .br2 { background-color: maroon; color: yellow;
          border-style: dotted; border-width: 2px;
          border-color: white; border-radius: 15px 35px; }
      .br3 { background-color: teal;  color: aqua;
          border-style: dashed; border-width: 5px;
          border-color: red; border-radius: 5px 15px 25px 35px; }
      .br4 { border: 3px solid red;
              border-top-left-radius: 30px; }
   </style>
</head>
<body>
   <h1 class="br1">둥근 모서리 스타일링 예제 1</h1>
   <h1 class="br2">둥근 모서리 스타일링 예제 2</h1>
   <h1 class="br3">둥근 모서리 스타일링 예제 3</h1>
   <h1 class="br4">둥근 모서리 스타일링 예제 4</h1>
</body>
```

❖ box-shadow

- 박스에 그림자 효과 적용
- 수평 그림자(h-shadow): 그림자의 수평 거리 지정
- 수직 그림자(v-shadow): 그림자의 수직 거리 지정
- 그림자 흐림(blur): 그림자의 흐림 정도 지정
- 그림자 번짐(spread): 그림자의 번짐 정도 지정
- 그림자 색상(color): 그림자의 색상 지정
- 삽입 효과(inset): 박스 외부로 표현되는 그림자를 박스 안쪽으로 표현하는 효과

```
{ box-shadow: 수평그림자(필수) I 수직그림자(필수) I 그림자흐림 I
                그림자번짐 I 그림자 색상 I 삽입효과; }
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Box-shadow

box-shadow01.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8”>
  <title>Box Shadow 1</title>
  <style>
        body { font-weight: bold; font-size: 12pt; }
        .boxshadow1 { background-color: yellow;
            border: 5px solid blue; box-shadow: 10px 10px 0px teal; }
        .boxshadow2 { background-color: orange;
            border: 5px solid red; box-shadow: 20px 20px 50px red; }
        .boxshadow3 { background-color: silver;
            border: 5px solid black;
            box-shadow: 20px 20px 30px -20px maroon; }
        .boxshadow4 { background-color: lime;
            border: 5px solid olive;
            box-shadow: 10px 10px 0px 10px fuchsia inset;
        }
    </style>
</head>
<body>
    <h1 class="boxshadow1">박스 그림자 스타일링 예제 1</h1>
    <h1 class="boxshadow2">박스 그림자 스타일링 예제 2</h1>
    <h1 class="boxshadow3">박스 그림자 스타일링 예제 3</h1>
    <h1 class="boxshadow4">박스 그림자 스타일링 예제 4</h1>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Box-shadow

box-shadow02.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8”>
  <title>Box Shadow 2</title>
  <style>
        img {
            padding: 20px;
            margin: 20px;
        }
        .shadow1 {    /* Bottom right coner */
            box-shadow: 5px 5px 10px #000;
        }
        .shadow2 {    /* Up right coner */
            box-shadow: 5px -5px 10px #000;
        }
    </style>
</head>
<body>
    <img src="pic1.jpg" class="shadow1">
    <img src="pic2.jpg" class="shadow2">
</body>
</html>
```

❖ position

- 텍스트, 이미지, 표 등의 요소를 웹 문서에 배치할 때 사용하는 속성

구분	속성값	설명
정적 위치 설정	position: static;	각종 요소를 웹 문서의 흐름에 따라 배치한다.
상대 위치 설정	position: relative;	웹 문서의 정상적인 위치에서 상대적으로 얼마나 떨어져 있는지 표시하여 배치하는 방법이다.
절대 위치 설정	position: absolute;	전체 페이지를 기준으로 top, right, bottom, left의 속성을 이용하여 원하는 위치에 배치하는 방법이다.
고정 위치 설정	position: fixed;	요소의 위치를 '절대 위치 설정'과 똑같은 방법으로 배치하되, 창의 스크롤을 움직여도 사라지지 않고 고정된 위치에 그대로 있다.

❖ position
- 정적위치

position_static.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Position Static</title>
  <style>
      body { font-weight: bold; font-size: 12pt; }
      .sp1 { position: static; top: 100px;    /* 적용되지 않음 */
         background-color: cyan; width: 400px; height: 50px; }
      .sp2 { position: static; left: 30px;    /* 적용되지 않음 */
         background-color: orange; width: 400px; height: 50px; }
       .sp3 { background-color: lightgreen;
         width: 400px; height: 50px; }
    </style>
</head>
<body>
    <h1>positioning style1</h1>
    <p class="sp1">정적 위치 설정 적용1</p>
    <div class="sp2">정적 위치 설정 적용2</div>
    <p class="sp3">기본 위치 설정</p>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ position
- 상대위치

position_relative.html

```
<!doctype html>
<html>
<head>
 <meta charset=“UTF-8”>
 <title>Position Relative</title>
 <head>
 <style>
    body { font-weight: bold; font-size: 12pt; }
    .sp { position: static; left: 30px;    /* 적용되지 않음 */
          background-color: cyan; width: 400px; height: 50px; }
    .rp1 { position: relative; left: 30px; top: -10px;
          background-color: orange; width: 400px; height: 50px; }
     .rp2 { position: relative; left: 60px; top: 20px;
          background-color: lightgreen; width: 400px;
                height: 50px; }
   </style>
</head>
<body>
   <h1>positioning style2</h1>
   <p class="sp">정적 위치 설정 적용</p>
   <div class="rp1">상대 위치 설정 적용 - left 30px, top -10px</div>
   <p class="rp2">상대 위치 설정 적용 - left 60px, top 20px</p>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ position
- 절대위치

position_absolute.html

```
<!doctype html>
<html>
<head>
 <meta charset=“UTF-8”>
 <title>Position Absolute</title>
 <head>
 <style>
    body { font-weight: bold; font-size: 12pt; }
    .ap1 { position : absolute; left: 30px; top: 70px;
         background-color: yellow; width: 400px; height: 50px; }
    .ap2 { position: absolute; left: 40px; top: 90px;
         background-color: lightgreen; width: 400px;
                height: 50px; }
     .rp { position: relative; left: 50px; top: 80px;
         background-color: cyan; width: 400px; height: 50px; }
   </style>
</head>
<body>
   <h1>positioning style3</h1>
   <div class="ap1">절대 위치 설정 적용 - left 30px, top 70px</div>
   <div class="ap2">절대 위치 설정 적용 - left 40px, top 90px</div>
   <div class="rp">상대 위치 설정 적용 - left 50px, top 80px</div>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ position
- 고정위치

position_fixed.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8”>
  <title>Position Fixed</title>
  <head>
  <style>
    body { font-weight: bold; font-size: 12pt; }
    .p { background-color: yellow; width: 300px; height: 50px; }
    .fp { position: fixed; right: 5px; top: 5px;
        background-color: lightgreen; width: 300px; height: 50px; }
  </style>
</head>
<body>
    <h1>positioning style4</h1>
    <p class="p">기본 위치 설정 박스1</p>
    <p class="p">기본 위치 설정 박스2</p>
    <p class="p">기본 위치 설정 박스3</p>
    <p class="p">기본 위치 설정 박스4</p>
    <p class="p">기본 위치 설정 박스5</p>
    <p class="fp">고정 위치 설정 박스 : 오른쪽 스크롤
    위아래로 이동해보기</p>
</body>
```

❖ float

- 화면을 구성하는 요소 간의 관계를 고려하여 각 요소를 배치하는 방법

속성값	설명
inherit	요소를 감싸는 부모 요소의 float 속성을 상속받는다.
left	요소를 왼쪽으로 떠 있는 상태로 만든다.
right	요소를 오른쪽으로 떠 있는 상태로 만든다.
none	float 속성을 적용하지 않는다(요소를 떠 있지 않게 한다).

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ float

float_position.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8”>
  <title>Float Position</title>
  <head>
        <style>
              img { float: right; margin: 0 0 10px 10px; }
        </style>
</head>
<body>
    <p>float 속성은 웹 문서의 레이아웃을 설계하는 과정에서 많이 사용하는 속성입니다.</p>
    <p><img src="pic1.jpg" alt="image" width="140" height="140">
    float 속성은 특정 요소를 떠 있게 해줍니다. 여기서 '떠 있다'라는 말의 의미는 특정 요소가 기본 레이아웃에서 벗어나 웹 문서의 왼쪽이나 오른쪽에 이동하는 것을 말합니다. float 속성은 복잡한 형태의 레이아웃을 구성하는 데 필요한 핵심 속성으로, 특정 요소가 주변 요소와 자연스럽게 어울리도록 해줍니다. 주의할 점은 float 속성을 사용할 때 요소의 위치가 고정되면 안 되기 때문에 position 속성의 absolute를 사용하면 안 됩니다.</p>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ float - clear

float_clear.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Float Clear</title>
  <head>
    <style>
        .div1 { float: left; width: 100px; height: 50px;
            margin: 10px; border: 3px solid #73AD21; }
        .div2 { border: 1px solid red; }
        .div3 { float: left; width: 100px; height: 50px;
            margin: 10px; border: 3px solid #73AD21; }
        .div4 { border: 1px solid red; clear: left; }
    </style>
</head>
<body>
    <h2>float 속성 사용</h2>
    <div class="div1">div1</div>
    <div class="div2">div2 - float 속성을 사용하여 대상 요소를 웹 문서에 배치하면 그 다음 요소에도 똑같은 속성이 적
        용됩니다. 하지만 float 속성이 사용되는 것을 원하지 않을 때도 있습니다. 이때 clear 속성을 사용합니다. 다양한 레이
        아웃 설계할 때에는 float 속성과 clear 속성을 적절히 잘 사용해야 합니다.</div>
    <h2>clear 속성 사용</h2>
    <div class="div3">div3</div>
    <div class="div4">div4 - clear 속성은 float 속성이 적용되는 것을 원하지 않는 요소에 사용하여 float 속성을 초기화
        시킵니다. float: left;를 사용했다면 clear: left;로, float: right;를 사용했다면 clear: right;로 초기화합니다.
        float 속성 값이 left 인지 right 인지 상관없이 무조건 초기화하고 싶다면 clear: both;를 사용합니다. 보통 clear:
        both;를 많이 사용합니다.</div>
</body>
</html>
```

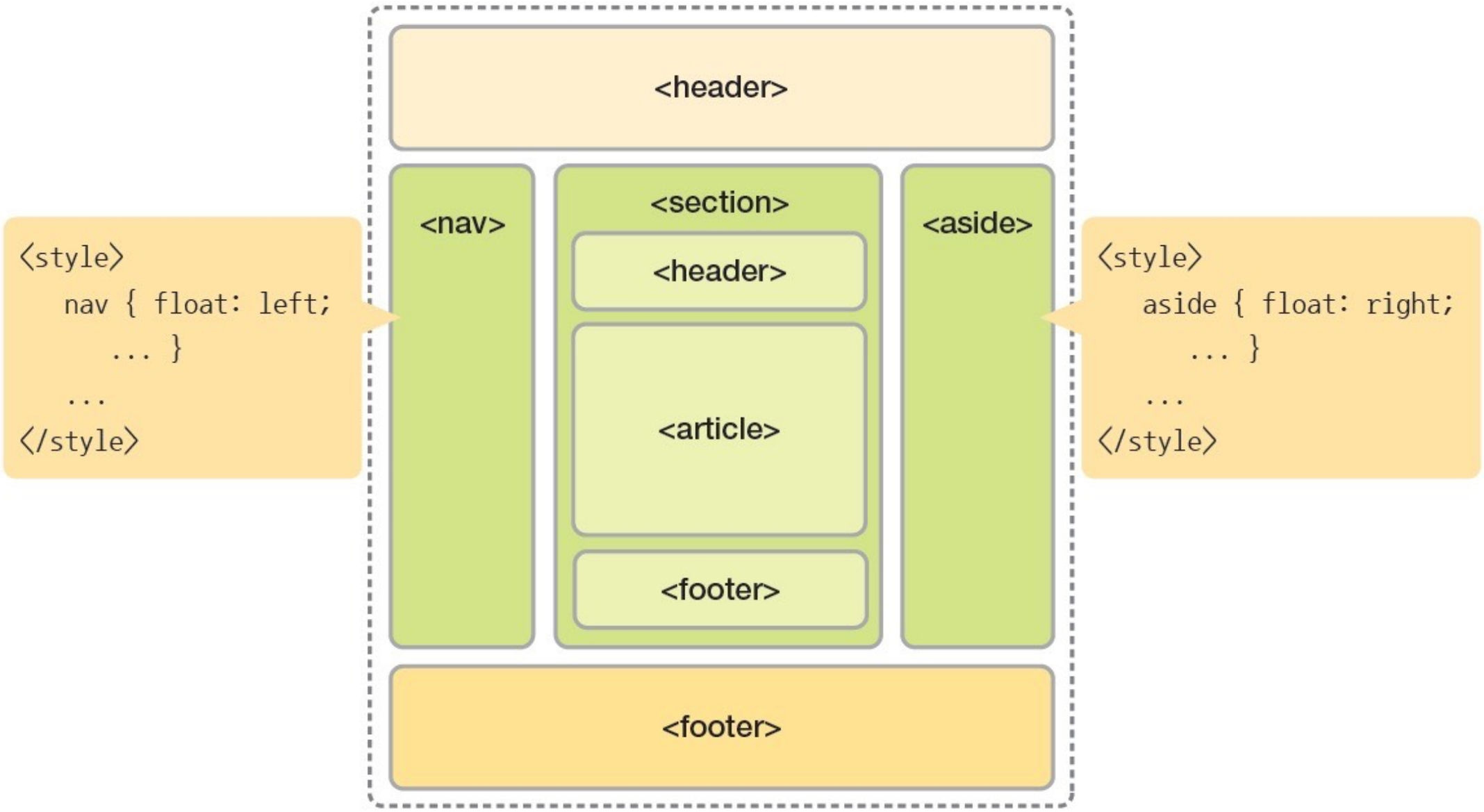
훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ float
- overflow

float_overflow.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8”>
  <title>Float Overflow</title>
  <head>
    <style>
        div {
            border: 3px solid #73AD21;
        }
        .img1 {
            float: right;
        }
        .fix {
            overflow: auto;
        }
        .img2 {
            float: right;
        }
    </style>
</head>
<body>
    <p>이미지가 박스 영역을 벗어남</p>
    <div><img class="img1" src="pic2.jpg" alt="image" width="140" height="140">
    이미지가 오른쪽 정렬로 되어 있는데, 박스 영역을 벗어났습니다.</div>
    <p style="clear:right">overflow: auto; 속성을 사용하여 해결</p>
    <div class="fix"><img class="img2" src="pic2.jpg" alt="img" width="140" height="140">
    이미지가 박스 영역을 벗어날 경우에는 overflow 속성을 auto로 설정하여 해결합니다.</div>
</body>
</html>
```

❖ 시맨틱 문서 구조에서 float 속성의 사용



❖ float - semantic

훈련교사 : 전 은 석
email : euns_jun@naver.com

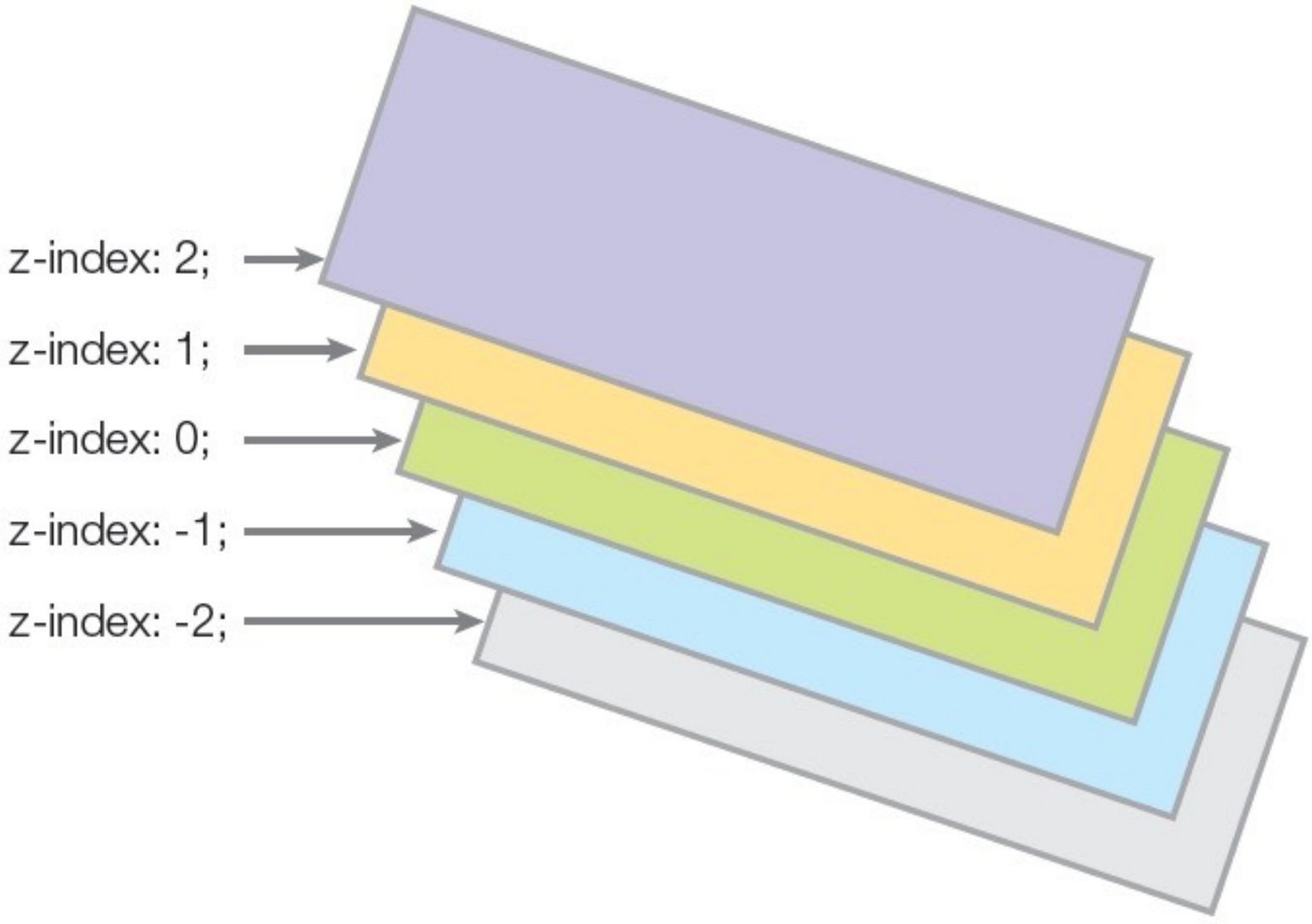
float_semantic.html

```
<!doctype html>
<html>
<head>
  <meta charset=“UTF-8”>
  <title>Float Semantic</title>
  <head>
    <style>
      div { border: 3px solid blue; }
      .fix { overflow: auto; }
      nav { float: left; width: 200px; border: 3px solid #73AD21; }
      section { margin-left: 206px; border: 3px solid red; }
    </style>
</head>
<body>
   <div class="fix">
      <nav>
         <span>목차</span>
         <ul>
            <li><a target="_blank" href="http://www.google.com">Google</a></li>
            <li><a target="_blank" href="http://www.apple.com">Apple</a></li>
            <li><a target="_blank" href="http://www.w3.org">W3C</a></li>
            <li><a target="_blank" href="http://www.oracle.com">Oracle</a></li>
            <li><a target="_blank" href="http://www.adobe.com">Adobe</a></li>
            <li><a target="_blank" href="http://www.amazon.com">Amazon</a></li>
            <li><a target="_blank" href="http://www.Mysql.com">Mysql</a></li>
         </ul>
      </nav>
      <section>
         <span>section 1</span>
         <p>float 속성은 시맨틱 문서 구조에 유용하게 사용할 수 있습니다. 예를 들면 nav나 aside에 float 속성을 추가하면 떠다니
            는 내비게이션 또는 사이드 바를 만들 수 있습니다.</p>
      </section>
      <section>
         <span>section 2</span>
         <p>시맨틱 문서 구조에 float 속성을 적용할 때는 footer 부분에 적용되지 않도록 해야 합니다.</p>
      </section>
   </div>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ z-index

- 한 요소 위에 다른 요소를 쌓을 때 사용
- z-index 속성값이 작을수록 아래에 쌓임



훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ z-index

z_index.html

```
<!doctype html>
<html>
<head>
 <meta charset="UTF-8">
 <title>Z-Index</title>
 <head>
 <style>
      #box1 { position: absolute; top: 0px; left: 0px;
         width: 100px;  height: 100px;
         background: blue; z-index: 3;
      }
      #box2 { position: absolute; top: 30px; left: 30px;
         width: 100px; height: 100px;
         background: yellow;z-index: 2;
      }
      #box3 { position: absolute; top: 60px; left: 60px;
         width: 100px; height: 100px;
         background: green; z-index: 1;
      }
   </style>
</head>
<body>
   <div id="box1">box #1</div>
   <div id="box2">box #2</div>
   <div id="box3">box #3</div>
</body>
```

❖ Overflow

- ◆ 영역의 크기보다 내용물이 더 많을 때, 영역을 벗어나 넘친 부분을 처리하는 방식을 지정

속 성	설 명
visible	넘치거나 말거나 그대로 출력 - 내용물이 넘치면 밖으로 빠져 나간다. : 기본값
hidden	넘친 부분을 잘라낸다. 가려진 부분 못본다.
scroll	넘친 부분은 숨기지만 스크롤 바를 표시하여 이동할 수 있다.
auto	넘칠 경우에만 스크롤 바를 보여준다.
no-display	전부 표시할 수 없다면 아예 전체 태그를 숨긴다. 아직 지원하는 브라우저가 없다.
no-content	전부 표시할 수 없다면 내용물을 숨긴다. 아직 지원하는 브라우저가 없다.

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 배경 속성

- 배경 속성을 이용해 해당 태그의 배경을 조절

속성 이름	설 명
background-image	배경 이미지 지정
background-size	배경 이미지 크기 지정
background-repeat	배경 이미지 반복 형태 지정
background-attahment	배경 이미지 부착 형태 지정
background-position	배경 이미지 위치 지정
background	한 번에 모든 배경 속성 입력

속성값	설 명
repeat	이미지를 수평 수직으로 반복한다. 기본값
no-repeat	한번만 표시
repeat-x	수평방향으로 반복
repeat-y	수직방향으로 반복

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Background

background.html

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Background</title>
  <link rel="stylesheet" type="text/css" href="./CSS/style01.css" />
</head>
<body>
   <h1>Lorem ipsum dolor sit amet</h1>
   <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec facilisis enim vitae est elementum euismod. Duis lobortis euismod nunc a dapibus. Proin nulla enim, adipiscing ac pharetra quis, vehicula a mauris. Maecenas nec tellus metus. Fusce sollicitudin facilisis orci, vel fringilla nisi molestie lobortis. Suspendisse dapibus dapibus semper. Duis ac purus nulla. Quisque elementum nibh non ipsum feugiat placerat. Cras tristique urna non urna hendrerit lacinia. Pellentesque porta velit ut felis faucibus in mollis purus malesuada. Duis porta, nulla sed suscipit elementum, arcu mi consectetur enim, id auctor risus enim a metus. Phasellus lorem quam, suscipit a congue vel, viverra eu nisi. Mauris ac felis lorem.</p>
   <p>Suspendisse convallis orci a justo gravida et fringilla nulla commodo. Morbi laoreet ante quis massa ullamcorper quis viverra leo semper. Donec velit mauris, sodales ut interdum eget, imperdiet vel tellus. Aenean tincidunt feugiat ipsum, sit amet placerat nunc venenatis eget. Sed iaculis ultricies enim. Praesent sed libero diam. Quisque eu ante ultrices enim vulputate laoreet. Ut vitae elit nisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Maecenas gravida erat ut est rhoncus pulvinar.</p>
   <p>Vestibulum mollis, ipsum eu ultrices commodo, metus mi tristique leo, at lacinia magna mi non justo. Suspendisse consequat, justo dictum scelerisque rutrum, mi massa placerat enim, in dictum dolor tellus vel justo. Suspendisse molestie dapibus augue quis cursus. Fusce volutpat, sapien eget fermentum congue, turpis est convallis urna, ac cursus diam ligula eu orci. Donec libero lacus, venenatis ac tempus sit amet, pretium in purus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Praesent dapibus aliquam nisi a venenatis. Mauris porttitor purus ut felis dapibus et adipiscing ante bibendum. Aenean rutrum leo lorem.</p>
   <p>Fusce ante augue, rutrum sit amet malesuada nec, accumsan et metus. Duis eros est, fermentum sit amet posuere a, venenatis in est. Nulla scelerisque turpis a magna tincidunt placerat. Nullam vehicula arcu eu massa cursus scelerisque. Sed vitae volutpat elit. Sed tincidunt est at ipsum porttitor ut luctus metus accumsan. Cras sollicitudin eros quis quam adipiscing quis tempus tortor lacinia. Vestibulum sit amet odio tortor, ut molestie dolor. Aliquam erat volutpat. Integer malesuada sem non velit molestie sed adipiscing risus feugiat. Ut ullamcorper, velit sed facilisis lacinia, ante odio laoreet mauris, nec tempor lectus ipsum non dui. Suspendisse porta convallis sem, in consectetur risus porta a. Cras dignissim velit eu urna aliquam varius. Nullam tempus scelerisque metus non ultrices.</p>
</body>
</html>
```

❖ Background - style01.css

style01.css

```
body {
    background-color: #E7E7E8;
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
    background-size: 100%;
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: 0px 50%;
}
```

❖ 글자 속성

속성 이름	설 명
font-size	글자의 크기 지정
font-family	컴퓨터에 설치된 글꼴 지정
font-style	글자의 기울기 지정
font-weight	글자의 두께 지정
text-align	글자의 정렬 방법 지정
line-height	글자의 높이 지정 / 높이지정보다 수직정렬에 많이 사용
text-decoration	글자를 꾸미는 속성 / a 태그의 href 속성의 밑줄 . . .

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Font

font.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Font Property</title>
    <style>
        .font_big { font-size: 2em; }
        .font_italic { font-style: italic; }
        .font_bold { font-weight: bold; }
        .font_center { text-align: center; }

        .button {
            width: 150px;
            height: 70px;
            background-color: #FF6A00;
            border: 10px solid #FFFFFF;
            border-radius: 30px;
            box-shadow: 5px 5px 5px #A9A9A9;
        }
        .button > a {
            display: block;
            line-height: 70px;
        }
    </style>
</head>
<body>
    <div class="button">
        <a href="#" class="font_big font_italic font_bold font_center">Click</a>
    </div>
</body>
</html>
```

❖ Vendor Prefix

- 누가 어떤 목적으로 도입한 기능인지 명확히 밝히는 규칙

업체 이름	접 두 어
마이크로소프트	-ms-
구글	-webkit-
파이어폭스	-moz-
사파리	-webkit-
오페라	-o-

❖ Gradient

- 두가지 색을 이용해서 변화를 주면서 채색하는 방법
- CSS에서는 Gradient를 이미지로 취급하므로 이미지를 사용하는 곳에서 사용한다.

훈련교사 : 전 은 석
email : euns_jun@naver.com

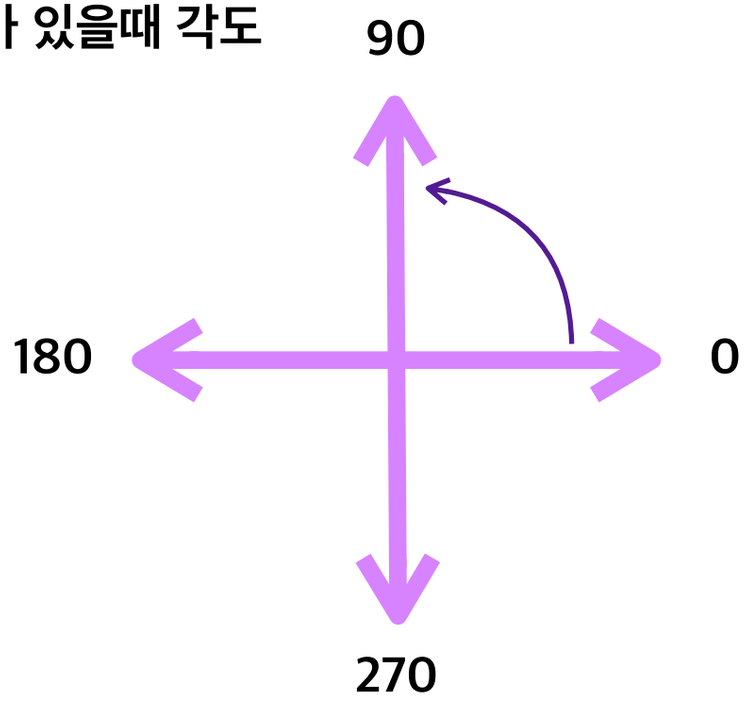
❖ Linear-Gradient

- 직선 방향 형태로 색상의 변화를 준다.

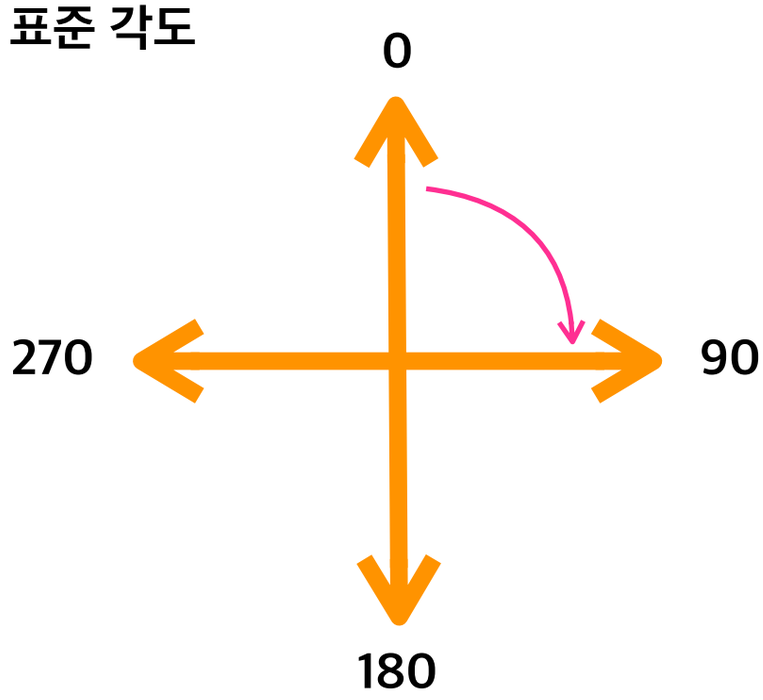
```
linear-gradient(각도, 시작색, 끝색)
```

```
linear-gradient(각도 I to 방향, 색상 %, 색상 %, 색상 %)
```

접두어가 있을때 각도



CSS 표준 각도



❖ Linear -Gradient

linear_gradient.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>linear-gradient2</title>
    <style>
        td {
            width:200px;
            height:200px;
            font-size:20pt;
        }
    </style>
</head>
<body>
    <table>
        <tr>
<td style="background:linear-gradient(180deg, red, yellow)">하향</td>
<td style="background:linear-gradient(to right, red, yellow)">우향</td>
<td style="background:linear-gradient(to right bottom, red, yellow)">우하향</td>
        <tr>
        </tr>
<td style="background:linear-gradient(90deg, red, yellow, green)">빨노초</td>
<td style="background:linear-gradient(90deg, red, yellow 30%, green)">초록 치우침</td>
<td style="background:linear-gradient(90deg, red 20%, yellow, green 80%)">가운데 몰림</td>
        </tr>
    </table>
</body>
</html>
```

❖ Radial-Gradient

- 방사형 방향 형태로 색상의 변화를 준다.

```
radial-gradient([circle I ellipse] [크기 I 예약어] at 위치, 색상 %, 색상 %)
```

크기 예약어	설명
closest-side	가장 가까운 면
closest-coner	가장 가까운 모서리
farthest-side	가장 먼 면
farthest-corner	가장 먼 모서리

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Radial-Gradient

radial_gradient.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>radial-gradient</title>
    <style>
            td { width:200px; height:120px; }
    </style>
</head>
<body>
    <table>
            <tr>
                    <td style="background:radial-gradient(circle, red, yellow);"></td>
                    <td style="background:radial-gradient(circle 60px, red, yellow);"></td>
                    <td style="background:radial-gradient(circle 60px at 120px 80px, red, yellow);"></td>
                    <td style="background:radial-gradient(circle 60px at 120px 80px, red, yellow 60%, green);"></td>
        <tr>
        </tr>
                    <td style="background:radial-gradient(circle farthest-corner at 120px 80px, red, yellow);"></td>
                    <td style="background:radial-gradient(circle farthest-side at 120px 80px, red, yellow);"></td>
                    <td style="background:radial-gradient(circle closest-corner at 120px 80px, red, yellow);"></td>
                    <td style="background:radial-gradient(circle closest-side at 120px 80px, red, yellow);">     </td>
        <tr>
        </tr>
                    <td style="background:radial-gradient(ellipse, red, yellow);"></td>
                    <td style="background:radial-gradient(ellipse 100px 50px, red, yellow);"></td>
                    <td style="background:radial-gradient(ellipse 100px 50px at 120px 80px, red, yellow);"></td>
                    <td style="background:radial-gradient(ellipse closest-side at 120px 80px, red, yellow);"></td>
        </tr>
    </table>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ table-layout

- ◆ 셀 안 내용의 양에 따라 셀 너비를 조절

속성값	설 명
table-layout: auto;	내용 분량에 따라 셀 너비가 자동으로 조절(기본값)
table-layout: fixed;	내용 분량과 관계없이 셀 너비 고정
table-layout: initial;	변경된 테이블 레이아웃을 기본값 상태(auto)로 설정
table-layout: inherit;	부모 요소의 값을 상속 받아 셀 너비를 결정

❖ 표 레이아웃 설정

tableLayout.html

```
<head>
    <meta charset="UTF-8">
    <title>Table Layout</title>
  <style type="text/css">
      td, th { border: 1px solid black;}
      #tb1 { border: 2px solid red; table-layout: auto; }
      #tb2 { border: 3px dotted teal; background-color: yellow; table-layout: fixed; }
  </style>
</head>
<body>
  <h2>table layout auto 예제</h2>
  <table id="tb1">
      <tr>
          <th>table layout auto</th>
          <td>내용 분량에 따라서 자동으로 조절</td>
      </tr>
  </table>
  <h2>table layout fixed 예제</h2>
  <table id="tb2" width="250px">
      <tr>
          <th>table layout fixed</th>
          <td>내용 분량과 상관 없이 고정</td>
      </tr>
  </table>
</body>
```

❖ 셀 테두리 설정

tableCollapse.html

```
<head>
    <meta charset="UTF-8">
    <title>Table Collapse</title>
  <style type="text/css">
    td, th { border: 2px solid black; }
    #tb1 { border: 3px solid red; background-color: yellow;
          border-collapse: separate; table-layout: auto; }
    #tb2 { border: 3px solid red; background-color: yellow;
          border-collapse: collapse; table-layout: auto; }
  </style>
</head>
<body>
  <table style id="tb1" width="350px">
    <tr>
      <th>table border-collapse</th>
      <td>separate 적용</td>
    </tr>
  </table>
  <p></p>
  <table style id="tb2" width="350px">
    <tr>
      <th>table border-collapse</th>
      <td>collapse 적용</td>
    </tr>
  </table>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 바깥 테두리와 셀 테두리 사이 간격 조정

tableSpacing.html

```
<head>
    <meta charset="UTF-8">
    <title>Table Spacing</title>
    <style type="text/css">
        td, th { border: 1px solid red; }
        #tb1 { border: 2px solid green; border-spacing: 10px; }    /* 상하좌우 */
        #tb2 { border: 3px solid maroon; background-color: aqua;
            border-spacing: 20px 40px; }    /* 첫번째 값: 좌우, 두번째 값: 상하 */
    </style>
</head>
<body>
    <table style id="tb1" width="350px">
        <tr>
            <th>table border-spacing</th>
            <td>10px</td>
        </tr>
    </table>
    <p></p>
    <table style id="tb2" width="350px">
        <tr>
            <th>table border-spacing</th>
            <td>20px 40px</td>
        </tr>
    </table>
</body>
```

❖ 빈 셀을 보이게 하거나 숨기기

tableEmpty.html

```
<head>
   <meta charset="UTF-8">
   <title>Empty Cell</title>
   <style type="text/css">
      td, th { border: 1px solid blue; }
      #tb1 { border-collapse: separate; empty-cells: hide; }
      #tb2 { border-collapse: separate; empty-cells: show; }
   </style>
</head>
<body>
   <table id="tb1" border="1" width="300px">
      <tr>
         <td>국어</td><td>영어</td><td></td>
      </tr>
      <tr>
         <td>수학</td><td></td>
      </tr>
   </table>
   <p></p>
   <table id="tb2" border="1" width="300px">
      <tr>
         <td>국어</td><td>영어</td><td></td>
      </tr>
      <tr>
         <td>수학</td><td></td>
      </tr>
   </table>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 캡션 추가하기

tableCaption.html

```
<head>
    <meta charset="UTF-8">
    <title>Table Caption</title>
    <style type="text/css">
        td, th { border: 2px solid black; }
        #c1 { border: 3px solid blue; caption-side: top; border-collapse: collapse; }
        #c2 { border: 3px solid red; caption-side: bottom; border-collapse: collapse; }
    </style>
</head>
<body>
    <table id="c1" border="1" width="300px">
        <caption>[table 1-1] Korea University</caption>
        <tr>
            <th>University</th><th>Contact</th><th>Country</th>
        </tr>
        <tr>
            <td>서울대학교</td><td>홍길동</td><td>KOREA</td>
        </tr>
    </table>
    <p></p>
    <table id="c2" border="1" width="300px">
        <caption>[table 1-2] USA University</caption>
        <tr>
            <th>University</th><th>Contact</th><th>Country</th>
        </tr>
        <tr>
            <td>Havard</td><td>Jackie</td><td>USA</td>
        </tr>
    </table>
</body>
```

❖ 마우스를 올리면 선명하게 보이게 설정하기

opacity1.html

```
<head>
    <style>
        a:link {
            opacity: 0.5;
        }
        a:hover {
            opacity: 1.0;
        }
        img {
            opacity: 0.2;
        }
        img:hover {
            opacity: 1.0;
        }
    </style>
</head>
<body>
    <h3>마우스를 올리면 선명하게 보입니다.</h3>
    <div>
        <a href="http://www.google.com">구글 웹 사이트</a>
    </div>
    <p></p>
    <div>
        <img src="pic1.jpg">
    </div>
</body>
```

❖ 텍스트 상자 안의 배경 이미지를 반투명하게 처리하기

opacity2.html

```
<head>
   <style>
      div.background {
         background: url(sky.jpg) repeat;
         border: 1px solid black;
      }
      div.box {
         margin: 30px;
         background-color: #ffffff;
         border: 2px solid blue;
         opacity: 0.5;
      }
      div.box p {
         margin: 5%;
         font-weight: bold;
         color: #000000;
         text-align: center;
      }
   </style>
</head>
<body>
   <div class="background">
   <div class="box">
      <p>HTML5 웹 프로그래밍</p>
  </div>
   </div>
</body>
```