

입문자를 위한 CSS 기초 강의

#1 CSS 의 역할 알아보기.

CSS 는 Cascading Style Sheets 를 의미한다.

- Cascading:계단식
- Style:멋을 내다
- Sheets:(종이)한 장

->계단식으로 스타일을 정의하는 문서!

CSS 는 HTML 문서에 스타일을 더해준다!

HTML 문서는 태그가 태그를 포함하는 계단식 구조를 지니고 있다.

내용정리

- CSS 는 웹문서를 꾸며주기 위해 사용하는 언어이다.
  - CSS 코드는 CSS 파일에 작성한다. 확장자는 \*.css 이다.
  - CSS 코드는 HTML 문서에 스타일을 추가(정의)하는 언어이다.
- 따라서 HTML 문서가 존재하지 않을 때는 단순 텍스트에 불과하다.

#2 CSS 기본 문법 및 사용 방법

CSS 기본 문법

```
-----  
선택자{  
    속성명: 속성값;  
}
```

선택자 : 어떤 요소에 스타일을 적용 할지에 대한 정보

{중괄호}:선택한 요소에 적용할 스타일을 정의하는 영역

속성명:어떤 스타일을○ 정의하고 싶은지에 대한 정보(색상, 크기 등)

속성값:어떻게 정의하고 싶은지에 대한 정보.

HTML 에 CSS 를 더하려면

HTML 문서에 CSS 문서(코드)를 적용하고자 할 때는  
다음과 같은 방식들을 사용할 수 있다.

- 인라인 스타일:태그에 직접 기술하기
- 스타일 태그:스타일시트-크 위한 태그를 추가하여 기술하기
- 문서 간의 연결:스타일시트 문서를 따로 작성하여 HTML 문서와 연결하기

인라인 스타일

태그에 style 속성을 추가하여 요소에 직접적으로 스타일을 정의하는 방식.  
따라서 선택자는 필요 없다.  
웹 콘텐츠와 스타일시트를 분리하기 위해서는 사용하지 않는 편이 좋다.

```
<p style="color: blue;">  
  글자를 파랗게 만들어줘  
</p>
```

스타일 태그

HTML 문서에<style></style>태그를 추가하여 그 안에 CSS 코드를 작성할 수 있다.

```
<style>  
  /* style 태그 안에는 CSS 코드를 작성해야 한다 */  
  p{ color: red; }  
</style>
```

문서 간의 연결

확자자가 \*.css 인 스타일시트 파일을 생성해 그 안에 CSS 코드를 작성하고,  
HTML 문서에 이를 연결해줄 수 있다. 이때는 <link>태그를 사용한다.

```
<link href="/style.css" rel="stylesheet">  
• href:연결하고자 하는 외부 소스의 url 을 기술하는 속성  
• rel:현재 문서(HTML)와 외부 소스의 연관 관계를 기술하는 속성
```

<link>태그는 HTML 문서의 <head></head> 내부에서 사용해야 합니다!

인라인 스타일

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

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

```
    <title>HTML 문서</title>
```

```
  </head>
```

```
  <body>
```

```
    <p style="color:red">
```

HTML 문서를 준비하고, 에디터로 여신 다음, 내용을 작성하고  
브라우저로 확인해주세요.

```
    </p>
```

```
  </body>
```

```
</html>
```

-----

스타일 태그

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

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

```
    <title>HTML 문서</title>
```

```
    <style>
```

```
      p{
```

```
        color:red;
```

```
        font-size: 32px;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
<p>
    HTML 문서를 준비하고, 에디터로 여신 다음, 내용을 작성하고
브라우저로 확인해주세요.
```

```
</p>
</body>
</html>
```

-----

문서 간의 연결  
index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <link href="./style.css" rel="stylesheet">
  </head>
  <body>
    <p>
      HTML 문서를 준비하고, 에디터로 여신 다음, 내용을 작성하고
      브라우저로 확인해주세요.
    </p>
  </body>
</html>
```

=====

style.css

```
p{
  color: gold;
}
```

내용 정리

- CSS 코드는 선택자, 중괄호 영역, 속성명, 속성값 등으로 구성된다.
- CSS 코드에도 주석을 사용할 수 있다. /\*주석\*/
- CSS 코드를 HTML 문서에 적용하기 위해서는 인라인 스타일, 스타일 태그, 외부 문서 연결 등의 방식을 사용해야 한다.

-----

### #3 CSS 선택자 1 편

```
선택자{  
    속성명: 속성값;  
}
```

#### 선택자의 종류

- 기본 선택자
- 그룹 선택자
- 특성 선택자
- 결합 선택자
- 의사 클래스
- 의사 요소

#### 기본 선택자 1.전체 선택자

모든 요소를 선택한다.

\*(에스터리스크)는 '문서 내의 모든 요소'를 의미하는 기호이다.

```
*{  
    color:blue;  
}
```

#### 기본 선택자 2.태그 선택자

```
p{
  color:blue;
}
```

문서내 모든 p 태그 요소의 글자 색을 파란 색으로 지정한다!

기본 선택자 3.클래스 선택자

주어진 class 속성값을 가진 요소를 선택한다.

주어진 class 속성값을 가진 요소가 다수일 경우, 해당 요소를 모두 선택한다.

```
.text{
  color:blue;
}
```

문서 내 class 가 "text"인 모든 요소의 글자 색을 파란 색으로 지정한다.

기본 선택자 4.아이디 선택자

주어진 id 속성값을 가진 요소를 선택한다.

id 는 고유한(unique)식별자 역할을 하는 전역 속성이다.

```
#topic{
  color:blue;
}
```

문서 내 id 가 "topic"인 요소의 글자 색을 파란 색으로 지정한다!

그룹 선택자

다양한 유형의 요소를 한꺼번에 선택하고자 할 때 사용한다.

선표(,)를 이용해 선택자를 그룹화한다

```
h1,p,div{
  color:blue;
}
```

}

문서 내 모든 h1,p,div 태그 요소의 글자 색을 파란 색으로 지정한다!

선택자가 겹치는 경우

선택자가 겹치는 경우, 기본적으로 나중에 작성된 스타일이 적용된다.

선택자가 다르지만 요소가 겹치는 경우, 선택자 우선순위에 의해 적용될 스타일이 결정된다.

선택자 우선순위?

아이디 선택자 > 클래스 선택자 > 태그 선택자

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

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

```
    <title>HTML 문서</title>
```

```
    <style>
```

```
      .text{
```

```
        color:green;
```

```
      }
```

```
      #gagaga{
```

```
        color:orange;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <h1 class="text">제목을 나타낸다</h1>
```

```
    <p id="gagaga">제목을 표시한다 가가가</p>
```

```
    <p class="text">제목을 표시한다 나나나</p>
```

```
    <p>제목을 표시한다 다다다</p>
```

```
  </body>
```

```
</html>
```

-----

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      h1,p{
        color:purple;
      }
    </style>
  </head>
  <body>
    <h1 class="text">제목을 나타낸다</h1>
    <p id="gagaga">제목을 표시한다 가가가</p>
    <p class="text">제목을 표시한다 나나나</p>
    <p>제목을 표시한다 다다다</p>
  </body>
</html>
```

-----

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      .text{ color:red;}
      h1{ color:blue;}
    </style>
```



```
</head>
<body>
  <h1 class="text">제목을 나타낸다</h1>
  <p id="gagaga">제목을 표시한다 가가가</p>
  <p class="text">제목을 표시한다 나나나</p>
  <p>제목을 표시한다 다다다</p>
</body>
</html>
```

## 내용정리

- 선택자는 어떤 요소에 대한 스타일을 정의할 것인지에 대한 정보이다.
- 기본 선택자로는 전체 선택자, 태그 선택자, 클래스 선택자, 아이디
- 그룹 선택자는 여러 개의 선택자를 그룹화할 수 있는 선택자이다.
- 문서 내에 동일한 선택자가 존재하는 경우, 뒤에 작성 스타일이 적용된다.
- 서로 다른 선택자가 동일한 요소를 선택한 경우, 우선순위를 고려해야 한다.

-----

2022-11-25

## #4 텍스트 꾸미기 속성

함께 공부해 볼 텍스트 관련 속성

- font-family:글꼴을 정의한다.
- font-size:글자 크기를 정의한다.
- text-align:정렬 방식을 정의한다.
- color:글자 색상을 정의한다.

font-family

요소를 구성하는 텍스트의 글꼴을 정의한다. 글꼴명을 속성값으로 지정한다.  
여러개의 글꼴을 연달아 기입하여 우선 순위를 지정할 수 있다.

사용예

```
*{  
  font-family:Times, monospace, serif;  
}
```

Times 를 우선 지정하되, 지원되지 않을 경우 monospace 를 지정한다!

font-size

수치와 단위를 지정해 글자의 크기를 정의할 수 있다.

단위: px 모니터 상의 화소 하나 크기에 대응하는 절대적인 크기

rem <html>태그의 font-size 에 대응하는 상대적인 크기

em 부모태그<상위태그>의 font-size 에 대응하는 상대적인 크기

사용예

```
span{ font-size: 16px; }  
span{ font-size: 2rem; }  
span{ font-size: 1.5em; }
```

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>HTML 문서</title>

<style>

html{ font-size: 12px; }

h1{ font-size: 2rem; }

p{ font-size: 16px; }

span{ font-size: 5em; }

</style>

</head>

<body>

<h1>아무 내용</h1>

<p>하하 반가워요!<span>호호</span></p>

</body>

</html>

-----

## text-align

블록 내에서 텍스트의 정렬 방식을 정의한다. 미리 정의된 키워드 값을 지정한다.

속성값 left/right 왼쪽 또는 오른쪽 정렬한다.

center 가운데 정렬한다.

justify 양끝 정렬한다(마지막 줄 제외)

## 사용예

```
p{ text-align: right; }
```

## color

텍스트의 색상을 정의한다. 다양한 방법으로 색상을 지정할 수 있다.

키워드: 미리 정의된 색상별 키워드를 사용한다.(ex..red,blue)

RGB 색상 코드: # + 여섯자리 16 진수 값 형태로 지정한다.

RGB 함수: Red,Green,Blue 의 수준을 각각 정의해 지정한다.

## 사용 예

```
span{ color: red; }
```

```
span{ color: #FF0000; }
```

```
span{ color: rgb(100%, 0%, 0%); }
```

RGB 색상은 빛의 삼원색(Red,Green,Blue)을

혼합하여 색을 정의하는 방식이다. 모든 수치가 가장 높을 때 흰색이 된다.

```
/* R:00, G:00, B:00 */
```

```
span{ color: #000000; }
```

```
/* R:70%, G:10%, B:20% */
```

```
span{ color:rgb(70%,10%,20%); }
```

-----

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      p{ text-align: center;
        color: blue;
      }
    </style>
  </head>
  <body>
    <p>결코 시간이 멈추어 질순 없다 yo!
      무엇을 망설이나 되는 것은 단지 하나 뿐인데
      바로 지금이 그대에게 유일한 순간이며
      바로 여기가 단지 그대에게 유일한 장소이다.

      환상 속에 그대가 있다
      모든 것이 이제 다 무너지고 있어도
      환상 속에 아직 그대가 있다
      지금 자신의 모습은 진짜가 아니라고 말한다
    </p>
  </body>
</html>
```

#### 내용정리

- font-family 는 글꼴을 지정하는 속성이며, 여러 글꼴을 연달아 기입하여 우선순위를 지정할 수 있다.
- font-size 는 글자 크기는 지정하는 속성이며, px,rem,em 등의 단위를 사용할 수 있다.
- text-align 은 블록 내에서 텍스트의 정렬 방식을 정의한다.
- color 는 다양한 방법으로 텍스트의 색상을 정의한다.

-----

## #5 display 속성 & border 속성

블록 레벨 요소 vs 인라인 요소

블록 레벨 요소:

자기가 속한 영역의 너비를 모두 차지하여 블록을 형성한다.

(ex. div, p, h1 등)

인라인 요소:

자기에게 필요한 만큼의 공간만 차지한다.

(ex. span, a 등)

display 속성

display 속성은 요소를 블록과 인라인 요소 중 어느 쪽으로 처리할지 정의한다.

/\* 블록 레벨 요소인 div 요소를

인라인으로 처리하고 싶다면

\*/

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

/\* 인라인 요소인 a 요소를

블록 레벨로 처리하고 싶다면

\*/

```
a{ display: block; }
```

display 속성 값

속성값 display에는 미리 정의되어 있는 키워드를 속성값으로 지정한다.

inline 인라인으로 처리한다.

block 블록 레벨로 처리한다.

inline-block 인라인으로 배치하되, 블록 레벨 요소의 속성을 추가할 수 있도록 처리한다.

none 디스플레이(표시)하지 않는다.

border 속성

border 속성을 사용하면 요소가 차지하고 있는 영역에 테두리를 그릴 수 있다.

border 속성에는 속성값으로 테두리의 두께, 모양, 크기 등을 함께 지정할 수 있는데, 이러한 속성을 '단축속성'이라 한다.

```
span{ border: 2px solid green; }
```

두께가 2px 인 직선 모양(solid)의 초록(green)테두리를 만들어줘!

border 속성의 하위 속성

border 속성에 지정하는 값들을 따로따로 지정할 수도 있다.

각 속성은 다음과 같다.

속성명 border-color : color 정의 방식과 동의

border-width : thin,medium,thick 등의 키워드 또는 px,em,rem 등의 단위

border-style : none(기본값),solid(직선),dotted(점선),dashed(긴 점선) 등

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>HTML 문서</title>
```

```
    <style>
```

```
      div{ border:3px solid red;
```

```
        display:inline;}
```

```
      span{
```

```
        border-width: 5px;
```

```
        border-style: dashed;
```

```
        border-color: rgb(100%,50%,0%);
    }
</style>
</head>
<body>
    <div>나는야 블록 레벨</div>
    <span>나는야 인라인</span>
</body>
</html>
```

-----

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      div{ border:3px solid red;
        display:inline;}
      span{
        border-width: 5px;
        border-style: dashed;
        border-color: rgb(100%,50%,0%);
        display:block;
      }
    </style>
  </head>
  <body>
    <div>나는야 블록 레벨</div>
    <span>나는야 인라인</span>
  </body>
</html>
```

내용 정리

- display 속성은 요소의 블록 레벨, 인라인 여부를 변경 처리할 수 있다.
- display 속성은 요소를 배치할 레이아웃을 결정할 때도 사용될 수 있다.
- border 속성은 요소가 차지하는 영역에 테두리를 그릴 수 있다.
- border 는 여러가지 속성을 함께 정의할 수 있는 '단축속성'이다.
- border 의 하위 속성은 border-style,border-width,border-color 이다.

-----

## #6 박스모델 1 편, 박스모델 소개

### 박스모델 (Box-Model)

브라우저가 요소를 렌더링 할 때, 각각의 요소는 기본적으로 사각형 형태로 영역을 차지하게 된다. 이 영역을 '박스'라 표현하며, css 는 박스의 크기, 위치, 속성 (색, 배경, 테두리 모양 등)을 결정할 수 있다.

하나의 박스는 다음 네 개의 영역으로 구성된다.

- 콘텐츠 영역
- 안쪽 여백
- 경계선(테두리)
- 바깥쪽 여백

박스 각 영역의 크기는 어떻게?

박스 각 영역의 크기를 정의할 수 있는 속성은 다음과 같다.

- 콘텐츠 영역:width,height
- 안쪽 여백:padding
- 바깥쪽 여백:margin
- 테두리:border-width

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

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



```

<title>HTML 문서</title>
<style>
    div{ border:3px solid red;
        padding:10px;
        margin:20px;
    }
</style>
</head>
<body>
    <div>요소의 콘텐츠</div>
</body>
</html>

```

#### 내용 정리

- 브라우저는 요소를 렌더링할 때 박스 모델에 따라 사각형으로 표현한다.
- 하나의 박스는 네 개의 영역으로 구성된다.
- 콘텐츠 영역, 안쪽 여백, 경계선, 바깥쪽 여백.
- 각 영역의 크기를 정의하는 속성이 존재한다.

#### #7 박스모델 2 편, margin padding 다루기

##### 다양한 경우의 수

여백은 상하좌우 네 개의 면에 존재하는 영역이다.

작성자는 각 면에 개별적으로 두께를 정의할 수 있다.

이를 위해 다음 두 가지 방법을 사용한다.

- 하위 속성 정의하기
- 여러 값을 한 번에 정의하기

##### 하위 속성 정의하기

상하좌우 여백을 정의할 수 있는 개별 속성들이 있다?!

padding-top, padding-left, padding-right, padding-bottom

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      div{
        width:100px; height:100px;
        padding:10px;
        border:5px solid red;
        margin: 20px;
        padding-top:10px;
        padding-right:20px;
        padding-bottom:30px;
        padding-left:40px;
        margin-top:40px;
        margin-right:30px;
        margin-bottom:20px;
        margin-left:10px;
      }
    </style>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

-----

여러 값을 한 번에 정의하기

padding 과 margin 은 네 면의 여백에 대한 단축속성이다.

```
span{
  display: inline-block;
  width:100px; height:100px;
  margin:10px 20px 30px 40px;
}
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      div{
        width:100px; height:100px;
        border:5px solid red;
        margin: 10px 30px 20px 40;
      }
    </style>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

#### 내용정리

- 여백은 상하좌우 네 면에 존재하는 영역이다.
- 상하좌우 개별 속성을 사용하면 여백 두께를 개별 정의할 수 있다.
- padding 과 margin 속성에 여러 면의 여백을 함께 정의할 수 있다.
- 어떤 방법을 사용할지는 사용자(개발자)마음이다.

-----

## #8 박스모델 3 편, box-sizing

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      div{
        width:100px; height:100px;
        border:5px solid tomato;
        padding:10px 20px;
      }
    </style>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

-----

### box-sizing

box-sizing 속성은 요소의 너비(width)와 높이(height)를 계산하는 방법을 지정한다.

속성값 content-box : 기본값. 너비와 높이가 콘텐츠 영역만을 포함한다.  
border-box : 너비와 높이가 안쪽 여백과 테두리까지 포함한다.

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      div{
        box-sizing:content-box;
        /* box-sizing:border-box */
        width:100px; height:100px;
        border:5px solid purple;
        padding:30px;
      }
    </style>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

#### 내용 정리

- 박스 모델에 너비와 높이를 지정하면, 콘텐츠 영역의 크기가 결정된다.
- box-sizing 속성은 너비와 높이가 포함할 영역을 변경함으로써 너비와 높이의 계산 방법을 결정할 수 있다.
- box-sizing 의 기본값은 content-box 이다.
- box-sizing 을 border-box 로 지정하면 테두리까지 너비와 높이에 포함된다.

-----

#### #9 박스모델 4 편, background

##### background

배경(background)은 콘텐츠의 배경을 정의한다.

단축 속성으로써 색상, 이미지, 반복 등 다양한 하위 속성을 정의할 수 있다.

하위속성 background-color : 배경 색을 정의한다.  
background-image : 배경 이미지를 정의한다.  
background-position : 배경 이미지의 초기 위치를 정의한다.  
background-size : 배경 이미지의 크기를 정의한다.  
background-repeat : 배경 이미지의 반복 방법을 정의한다.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      div{
        box-sizing:border-box;
        width:500px; height:500px;
        border:1px solid red;
        background-color:rgb(30%, 20%, 65%);
        background-image: url(/images/tree.jpg);
        background-repeat: no-repeat; /* 반복 되지 않는 이미지 */
      }
    </style>
  </head>
  <body>
    <div>헬로우</div>
  </body>
</html>
```

-----

```
<!DOCTYPE html>
<html lang="en">
  <head>
```

```

<meta charset="utf-8">
<Title>HTML 문서</Title>
<Style>
    div{
        box-sizing:border-box;
        width:500px; height:500px;
        border: 1px solid red;
        /*
        background-color:rgb(20%, 40%, 80%);
        background-image:url(/images/my-cat.jpg);
        background-repeat:no-repeat;
        background-position:center;
        background-size:contain;
        background: blue;
        */
        background:url(/images/my-cat.jpg) no-repeat;
    }
</Style>
</head>
<body>
    <h1>HTML 문서</h1>
    <div>헬로우</div>
</body>
</html>

```

## 내용 정리

- background 는 콘텐츠의 배경을 정의한다.
- 단축 속성으로써, 다양한 하위 속성을 포함한다.
- 하위 속성을 한 번에 정의할 경우, 정의할 속성과 정의를 생략할 속성을 다양하게 선택할 수 있어 여러 경우의 수가 존재한다.
- 학습 팁! 정의 시 여러 경우의 수가 존재하는 속성을 학습할 때는 외우려고 하기 보다는 경험을 통해 체득하는 방향으로 공부하자.

-----

#10 float 그리고 clear

float

float 속성은 요소가 문서의 일반적인 흐름에서 제외되어 자신을 포함하고 있는 컨테이너의 왼쪽이나 오른쪽에 배치되게 한다.

속성값 none : 기본값, 원래 상태

left : 자신을 포함하고 있는 박스의 왼쪽에 떠 있어야 함.

right : 자신을 포함하고 있는 박스의 오른쪽에 떠 있어야 함.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

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

```
    <Title>HTML 문서</Title>
```

```
    <Style>
```

```
      #a{ width:100px; height:50px;
          background-color:orange; }
```

```
      #b{ width:100px; height:100px;
          background-color:royalblue; }
```

```
    </Style>
```

```
  </head>
```

```
  <body>
```

```
    <div id="a"></div>
```

```
    <div id="b"></div>
```

```
    <p>안녕하세요 저는 세 번째 태그예요</p>
```

```
  </body>
```

```
</html>
```

-----



```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <Title>HTML 문서</Title>
    <Style>
      #a{ width:100px; height:50px;
        background-color:orange;
        float: right; }
      #b{ width:100px; height:100px;
        background-color:royalblue;
        float:left; }
    </Style>
  </head>
  <body>
    <div id="a"></div>
    <div id="b"></div>
    <p>안녕하세요 저는 세 번째 태그예요</p>
    <p>안녕하세요 저는 세 번째 태그예요</p>
    <p>안녕하세요 저는 세 번째 태그예요</p>
    <p>안녕하세요 저는 세 번째 태그예요</p>
    <p>안녕하세요 저는 세 번째 태그예요</p>
    <p>안녕하세요 저는 세 번째 태그예요</p>
  </body>
</html>

```

-----

clear

clear 속성은 float 요소 이후에 표시되는 요소가 float 을 해제(clear)하여 float 요소의 아래로 내려가게 할 수 있다.

속성값 none : 기본값, 아래로 이동되지 않음을 나타내는 키워드  
left : float 이 left 인 요소의 아래로 내려가겠다  
right : float 이 right 인 요소의 아래로 내려가겠다  
both : float 이 left 및 right 인 요소의 아래로 내려가겠다

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <Title>HTML 문서</Title>
    <Style>
      #a{ width:100px; height:50px;
        background-color:orange;
        float: right; }
      #b{ width:100px; height:100px;
        background-color:royalblue;
        float:left; }

      /*
      p{ clear: left; }
      p{ clear: right; }
      */
      p{ clear: both; }
    </Style>
  </head>
  <body>
    <div id="a"></div>
    <div id="b"></div>
    <p>안녕하세요 저는 세 번째 태그예요</p>
    <p>안녕하세요 저는 세 번째 태그예요</p>
    <p>안녕하세요 저는 세 번째 태그예요</p>
    <p>안녕하세요 저는 세 번째 태그예요</p>
    <p>안녕하세요 저는 세 번째 태그예요</p>
```

```
<p>안녕하세요 저는 세 번째 태그예요</p>
</body>
</html>
```

## 내용 정리

- float 속성은 요소가 자신을 포함하고 있는 컨테이너의 왼쪽이나 오른쪽에 배치되게 한다.
- 이때 요소는 문서의 흐름에선 제외되지만, 필요한 만큼의 공간은 차지한다.
- clear 속성은 float 요소 이후에 표시되는 요소가 float 요소의 아래로 내려가게 할 수 있다.

-----

## #11 position 1 편, relative absolute

### position

position 은 문서 상에 요소를 배치하는 방법을 정의한다.  
position 이 요소의 배치 방법을 결정하면,  
top, bottom, right, left 가 최종 위치를 결정하는 방식이다.

position: 난 이렇게 배치할거야;  
top: 윗면에서부터 얼마나 떨어뜨릴거야;  
right: 오른쪽면에서부터 얼마나 떨어뜨릴거야;  
bottom: 아랫면에서부터 얼마나 떨어뜨릴거야;  
left: 왼쪽면에서부터 얼마나 떨어뜨릴거야;

### position 속성값

position 속성에는 다음 속성값들을 지정할 수 있다.

속성값 static : 기본값. 요소를 일반적인 문서 흐름에 따라 배치한다.  
relative : 일반적인 문서 흐름에 따라 배치하되, 상하좌우 위치 값에 따라  
오프셋을 적용한다.  
absolute : 일반적인 문서 흐름에서 제거하고,  
가장 가까운 position 지정 요소에 대해 상대적으로 오프셋을  
적용한다.  
fixed : 일반적인 문서 흐름에서 제거하고, 지정된 위치에 고정된다.  
sticky : 일반적인 문서 흐름에서 제거하고,  
스크롤 동작이 존재하는 가장 가까운 요소에 대해 오프셋을  
적용한다.

static(고정된, 정지상태의) relative(비교상의, 상대적인, 관련된)  
absolute(완전한,완벽한,절대적인것) fixed(변함없는, 고정된, 확고한)  
sticky(달라붙는, 붙일수있는)

position: relative;

요소를 일반적인 문서 흐름에 따라 배치하되, 상하좌우 위치 값에 따라 오프셋  
을 적용한다. 오프셋(보충)이란, 위치를 얼마간 이동시키는 것을 의미한다!

```
div{  
  width:100px; height:100px;  
  background-color: red;  
  
  position: relative;  
  top: 100px; left: 100px;  
}
```

\* position:lelative; \*

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="utf-8">
```

```

<title>HTML 문서</title>
<style>
    div{
        width:100px; height:100px;
        background-color:red;

        position:relative;
        top:500px; left:50px;
    }
</style>
</head>
<body>
    <div></div>
</body>
</html>

```

-----

\* position: absolute; \*

position: absolute;

요소를 일반적인 문서 흐름에서 제거하고, 상위 요소 중 가장 가까운 position 지정 요소에 대해 상대적으로 오프셋을 적용한다.

position 지정 요소란, position 속성에 속성값이 정의되어 있는 요소!

```

div{
    width:100px; height:100px;
    background-color:red;

    position:absolute;
    top:100px; left:100px;
}

```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      div{
        width:200px; height:200px;
        background-color:yellow;
        position:relative;
        border:1px solid black;
      }
      .abs{
        width:100px; height:100px;
        background-color:red;
        position:absolute;
        top:0px; left:100px;
      }
    </style>
  </head>
  <body>
    <div></div>
    <div>
      <div class="abs"></div>
      <p>PPPPP</p>
    </div>
  </body>
</html>
```

-----

#12 position 2 편, fixed sticky

position 속성값

position 속성은 다음 속성값들로 정의할 수 있다.

속성값    static    기본값. 요소를 일반적인 문서 흐름에 따라 배치한다.

          relative    일반적인 문서 흐름에 따라 배치하되, 상하좌우 위치 값에 따라 오프셋을 적용한다

          absolute    일반적인 문서 흐름에서 제거하고,  
                      가장 가까운 position 지정 요소에 대해 상대적으로 오프셋을 적용한다.

          fixed        일반적인 문서 흐름에서 제거하고, 지정한 위치에 고정된다.

          sticky        일반적인 문서 흐름에서 제거하고, 지정한 위치에 고정된다.  
                      스크롤 동작이 존재하는 가장 가까운 요소에 대해 오프셋을 적용한다.

position:fixed;

요소를 일반적인 문서 흐름에서 제거하고, 지정된 위치에 고정시킨다.

```
.pos{
  width:100px; height:200px;
  background: peru;
  position: fixed;
  top: 50px; left: 50px;
}
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

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

```
    <title>HTML 문서</title>
```

```
    <style>
```

```
      .pos{
```

```
        width:200px; height:200px;
```

```
        background-color:tomato;
```

```

        position:fixed;
        top:200px; right:100px;
    }
</style>
</head>
<body>
    <div>
        첫번째</br>첫번째</br>첫번째</br>첫번째</br>
        첫번째</br>첫번째</br>첫번째</br>첫번째</br>
        첫번째</br>첫번째</br>첫번째</br>첫번째</br>
        첫번째</br>첫번째</br>첫번째</br>첫번째</br>
        첫번째</br>첫번째</br>첫번째</br>첫번째</br>
        첫번째</br>첫번째</br>첫번째</br>첫번째</br>
        첫번째</br>첫번째</br>첫번째</br>첫번째</br>
    </div>
    <div class="pos"></div>
    <div>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
    </div>
</body>
</html>

```

-----

position:sticky;

요소를 일반적인 문서 흐름에 따라 배치하고, 스크롤(scroll)되는 가장 가까운 상위 요소에 대해 오프셋을 적용한다.



```
.pos{
  width:200px; height:200px;
  background: peru;
  position: sticky;
  bottom: 0px;
}
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      .pos{
        width:200px; height:200px;
        background-color:tomato;
        position:sticky;
        bottom:0px;

        /*
        top:0px;
        */

      }
    </style>
  </head>
  <body>
    <div>
      첫번째<br>첫번째<br>첫번째<br>첫번째<br>
      첫번째<br>첫번째<br>첫번째<br>첫번째<br>
      첫번째<br>첫번째<br>첫번째<br>첫번째<br>
      첫번째<br>첫번째<br>첫번째<br>첫번째<br>
      첫번째<br>첫번째<br>첫번째<br>첫번째<br>
    </div>
  </body>
</html>
```

```

        첫번째</br>첫번째</br>첫번째</br>첫번째</br>
        첫번째</br>첫번째</br>첫번째</br>첫번째</br>
    </div>
    <div class="pos"></div>
    <div>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
        두번째<br/>두번째<br/>두번째<br/>두번째<br/>
    </div>
</body>
</html>

```

## 내용정리

- position 은 문서 상에 요소를 배치하는 방법을 정의한다. 기본값은 static
- top, bottom, right, left 를 선택적으로 사용해 위치를 지정한다.
- relative 는 자기의 원래 위치를 기준으로 오프셋을 적용한다.
- absolute 는 가장 가까운 포지션 요소를 기준으로 오프셋을 적용한다.
- fixed 는 고정되어 움직이지 않는다. 스크롤 이동에도 꿈쩍 않는다.
- sticky 는 스크롤 되는 상위 요소를 기준으로 오프셋을 적용한다.

-----  
2022-11-28

## #13 flexbox 1 편, 정의 및 사용 방법

### flexbox

flexbox 란 박스 내 요소 간의 공간 배분과 정렬 기능을 제공하기 위한 1 차원 레이아웃 모델이다.

flexbox 를 1 차원 모델이라 부르는 이유는, 레이아웃을 다룰 때 한 번에 하나의 차원(행이나 열)만을 다룬다는 특성 때문이다.

layout(배치) row(행,세로) column(열,가로)

flexbox 만들기

flexbox 를 flex 컨테이너 라고도 한다(요소들을 포함하기 때문).

flex 컨테이너를 만들기 위해서는 컨테이너에 display:flex; 를 적용해야 한다.

적용전)세로로 표현된 결과

```
<div>
  <div class="item">하나</div>
  <div class="item">둘</div>
  <div class="item">셋</div>
</div>
```

적용후)가로로 표현된 결과

```
<div style="display: flex;">
  <div class="item">하나</div>
  <div class="item">둘</div>
  <div class="item">셋</div>
</div>
```

먼저 알아야 할 것

flexbox 에는 '주축'(main-axis)'과 '교차축(cross-axis)'이 있다.

flex-box(네모상자를 십자형태로 "좌에서 우는 주축" "위에서 아래는 교차축"이라 하고 기준에 따라서 바뀜)

flex-direction

flex-direction 속성은 flexbox 내 요소를 배치할 때 사용할 주축 및 방향(정방향, 역방향)을 지정한다.

속성값	row	기본값, 주축은 행이고 방향은 콘텐츠의 방향과 동일.
	row-reverse	주축은 행이고 방향은 콘텐츠의 방향과 반대.
	column	주축은 열이고 방향은 콘텐츠의 방향과 동일.
	column-reverse	주축은 열이고 방향은 콘텐츠의 방향과 반대.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      .container{
        display:flex;
        /* flex-direction: row; 기본값 가로 */
        /* flex-direction: column; 세로 배치 */
        /* flex-direction: column-reverse; 세로 역순*/
        /* flex-direction: row-reverse; 가로 역순 */
        flex-direction: column-reverse;
      }
      .item{
        width:80px; height:80px;
        background-color:orange;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="item">1</div>
      <div class="item">2</div>
      <div class="item">3</div>
    </div>
  </body>
</html>
```

## 내용 정리

- flexbox 란 박스 내 요소 간의 공간 배분과 정렬 기능을 제공하기 위한 1 차원(행 또는 열)레이아웃 모델이다.
- 행 또는 열이 주축이 되어 요소의 정렬 방향을 결정한다.
- flex-direction 속성은 요소를 배치할 때 사용할 주축 및 방향을 지정한다.

-----

## #14 flexbox 2 편, 몇 가지 관련 속성

### 중요한 건 다시 한번

flexbox 에는 '주축(main-axis)'과 '교차축(cross-axis)'이 있다.

### flexbox 다루기

flexbox 를 다루기 위해 다음과 같은 속성들을 사용할 수 있다.

- 주축 배치 방법:justify-content
- 교차축 배치 방법:align-items
- 교차축 개별요소 배치 방법:align-self
- 줄 바꿈 여부:flex-wrap

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

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

```
    <title>HTML 문서</title>
```

```
    <style>
```

```
      .container{
```

```
        display:flex;
```

```
        width:300px; height:300px;
```

```

border:2px solid black;
justify-content:space-between;
align-items:center;
/* align-items: center; 교차축 가운데 정렬 */
/* justify-content:center; 가운데 정렬*/
/* justify-content:flex-end; 오른쪽 정렬*/
/* justify-content:flex-start; 왼쪽 정렬*/
/* justify-content:space-around; 균등 좌우 여백 정렬*/
/* justify-content:space-between; 첫번째와 마지막 요소를
양끝에 붙이고 정렬함*/
/* align-items: flex-start; 교차축 제일 앞쪽(맨위) */
/* align-items: center; 교차축 가운데 정렬 */
}
.item{
width:60px; height:60px;
background-color:teal;
}
.self{
/*align-self:flex-start; 개별요소 하나가 교차축 제일 앞 (가운데 맨
위로) */
align-self:flex-end; /* 개별요소 하나가 교차축 제일 뒤 (가운데 맨
밑 으로) */
}
</style>
</head>
<body>
<div class="container">
<div class="item">1</div>
<div class="item self">2</div>
<div class="item">3</div>
</div>
</body>
</html>

```

-----

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      .container{
        display:flex;
        width:300px; height:300px;
        border:2px solid black;
        flex-wrap: nowrap;
        /* flex-wrap: nowrap; 기본 상태와 동일*/
        /* flex-wrap: wrap; *주축 하위요소 도합 크기가 플렉스 컨테이너
        축길이 보다 커졌을때 두행 이상으로 처리 하겠다는 의미 /
        /* flex-wrap: wrap-reverse; 랩(싸다, 포장하다)을 하는데
        반대 반향으로 하겠다는 의미 */
      }
      .item{
        width:120px; height:60px;
        background-color:teal;
      }
      .self{
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="item">1</div>
      <div class="item self">2</div>
      <div class="item">3</div>
    </div>
  </body>
</html>
```

## 내용정리

- justify-content 는 flexbox 주축에서의 요소 배치 방법을 정의한다.
- align-items 는 flexbox 교차축에서의 요소 배치 방법을 정의한다.
- align-self 는 flexbox 내 개별 요소의 교차축 배치 방법을 정의한다.
- flex-wrap 은 개별 요소들의 도합 크기가 컨테이너 주축 길이보다 커졌을 때의 처리 방법 및 방향을 정의한다.

-----

## #15 선택자 2 편, 특성 선택자와 결합자

```
선택자{  
  속성명: 속성값;  
}
```

선택자:어떤 요소에 스타일을 적용 할 것인지에 대한 정보.

### 선택자의 종류

- 기본 선택자
- 그룹 선택자
- 특성 선택자
- 결합 선택자
- 의사 클래스
- 의사 요소

이번 '선택자 2 편'에서는 특성 선택자,결합 선택자에 대해 알아보겠습니다!

### 특성 선택자 1.컨셉

특성 선택자(속성 선택자)는 주어진 속성의 존재 여부나 그 값에 따라 요소를 선택한다.



```
[class]{
  background-color: tomato;
}
```

->클래스 속성을 가지고 있는 요소를 선택하기(위)

-----

```
[class="item"]{
  background-color: tomato;
}
```

->클래스가 "item"인 요소를 선택하기(아래)

-----

특성 선택자 2.값 확인

기호를 추가하여 요소를 선택하는 방식을 다양화할 수 있다.

```
[class *= "it"]{ color: white; }
[class ^= "it"]{ color: white; }
[class $= "it"]{ color: white; }
```

->클래스 값에 "it"가 포함되는 요소를 선택하기(상)

->클래스 값이 "it"로 시작하는 요소를 선택하기(중)

->클래스 값이 "it"로 끝나는 요소를 선택하기(하)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

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

```
    <title>HTML 문서</title>
```

```
<style>
    [class$="ext"]{
        color:blue;
    }
    [id^="que"]{
        color:red;
    }
</style>
</head>
<body>
    <h1>선택자 연습하기</h1>
    <p>1 번 문단입니다 그냥 태그</p>
    <p class="text">2 번 문단입니다. 클래스 있음</p>
    <p id="unique">3 번 문단입니다. 아이디 있음</p>
    <p class="text">4 번 문단입니다. 클래스 있음</p>
    <p>5 번 문단입니다. <span>스판 있는 </span>태그</p>

</body>
</html>
```

-----

## 결합 선택자 1.컨셉

결합 선택자(결합자)는 두 개 이상의 선택자를 결합시켜 결합된 조건을 만족하는 요소를 선택한다.

다음 두 가지로 구분할 수 있다.

- 자손 결합자
- 형제 결합자

## 결합 선택자 2.자손 결합자

두 개의 선택자 중 첫 번째 선택자 요소의 자손을 선택할 수 있다.

```
div p{ color: white; }
```

```
div > p{ color: white; }
```

->div 요소 안에 위치하는 모든 p 요소를 선택하기(상)

-> div 요소의 바로 아래에 위치하는 모든 p 요소를 선택하기(하)

```
h1 ~ p{ color: red; }
```

```
h1 + p{ color: red; }
```

->h1 요소의 뒤에 오는 형제 중 모든 p 요소를 선택하기(상)

->h1 요소의 바로 뒤에 오는 형제 p 요소를 선택하기(하)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

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

```
    <title>HTML 문서</title>
```

```
    <style>
```

```
      p > span{
```

```
        color:goldenrod;
```

```
      }
```

```
      .text + p{
```

```
        color:red;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <h1>선택자 연습하기</h1>
```

```
    <p>1 번 문단입니다 그냥 태그</p>
```

```
    <p class="text">2 번 문단입니다. 클래스 있음</p>
```

```
    <p id="unique">3 번 문단입니다. 아이디 있음</p>
```

```
    <p class="text">4 번 문단입니다. 클래스 있음</p>
```

<p>5 번 문단입니다. <span>스판 있는 </span>태그</p>

</body>

</html>

## 내용 정리

- 특성 선택자는 주어진 특성의 존재 여부나 그 값에 따라 요소를 선택한다.
- 특성 선택자는 값의 포함 여부, 포함 위치 등을 통해서도 선택할 수 있다.
- 결합 선택자는 여러 선택자를 결합하여 요소를 선택한다.
- 자손 또는 형제 중에서 특정 조건을 만족하는 요소를 선택할 수 있다.

-----

## #16 의사클래스(가상클래스)

### 의사클래스

의사클래스(가상클래스)는 선택자에 추가하는 키워드로, 요소가 어떤 특정한 상태가 되었을 때 요소를 선택하겠다는 의미이다.

선택자:의사클래스}

속성명: 속성값;

{

-----

```
h1:hover{
  color: red;
}
```

->h1 요소에 마우스 커서가 올라오면(hover)글자를 빨간색으로 하겠다!

(정상시 상태 검정색에서 마우스 커서가 올려진 상태에서 빨간색으로 바뀜.)

## 실습대상 의사클래스

hover        마우스 포인터가 요소에 올라가 있다.  
active        사용자가 요소를 활성화했다.(예를 들면, 마우스로 누르기와 같은)  
focus        요소가 포커스를 받고 있다.  
disabled     비활성 상태의 요소이다.  
nth-child() 형제 사이에서의 순서에 따라 요소를 선택한다.

=>사용 가능한 모든 의사 클래스는 링크를 참조!

<https://developer.mozilla.org/ko/docs/Web/CSS/Pseudo-classes>

\*빨간색 버튼 위에 마우스 포인터가 올라갔을때 회색으로 바뀜.클릭하면 검정색으로 바뀜.

(hover, active 사용)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

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

```
    <title>HTML 문서</title>
```

```
    <style>
```

```
      /* hover active focus disabled nth-child() */
```

```
      [type="button"]{
```

```
        width:100px; height:30px;
```

```
        background-color:tomato; color: white;
```

```
        border:none; border-radius: 8px;
```

```
      }
```

```
      [type="button"]:hover{
```

```
        background-color:gray;
```

```
      }
```

```
      [type="button"]:active{
```

```
        background-color:black;
```

```

        }
    </style>
</head>
<body>
    <input type="button" value="버튼">
</body>
</html>

```

-----

\*아무거나 쓰기 테두리 박스생성. 클릭하면 빨강색으로 바뀌고 글자입력시 흰색으로  
입력됨.

(focus)

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>HTML 문서</title>
        <style>
            /* hover active focus disabled nth-child() */
            input:focus{
                color:white;
                background-color:red;
            }
        </style>
    </head>
    <body>
        <input type="text" placeholder="아무거나 쓰기">
    </body>
</html>

```

-----

입력창이 사용하지 못하게 됨.

(disabled 사용)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      /* hover active focus disabled nth-child() */
      input:focus{
        color:white;
        background-color:red;
      }
      input:disabled{
        height: 100px;
      }
    </style>
  </head>
  <body>
    <input type="text" placeholder="아무거나 쓰기" disabled>
  </body>
</html>
```

-----

1 번 에서 5 번 파란색 박스가 화면 맨 위에 정렬되었음. 글자는 흰색.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>HTML 문서</title>
    <style>
      body{
        display: flex;
        justify-content:space-between;
```

```

    }
    .box{
        width:50px; height:50px;
        background-color:blue;color:white;
    }
</style>
</head>
<body>
    <div class="box">1 번</div>
    <div class="box">2 번</div>
    <div class="box">3 번</div>
    <div class="box">4 번</div>
    <div class="box">5 번</div>
</body>
</html>

```

-----

1 번 에서 5 번 파란색 박스가 화면 맨 위에 정렬되었음. 글자는 흰색.  
3 번 박스 배경색이 빨간색 빨강색 으로 되어있음.

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>HTML 문서</title>
        <style>
            body{
                display: flex;
                justify-content:space-between;
            }
            .box{
                width:50px; height:50px;
                background-color:blue;color:white;
            }
        </style>
    </head>
    <body>
        <div class="box">1 번</div>
        <div class="box">2 번</div>
        <div class="box">3 번</div>
        <div class="box">4 번</div>
        <div class="box">5 번</div>
    </body>
</html>

```



```

    }
    .box:nth-child(3){
        background-color:red;
    }
</style>
</head>
<body>
    <div class="box">1 번</div>
    <div class="box">2 번</div>
    <div class="box">3 번</div>
    <div class="box">4 번</div>
    <div class="box">5 번</div>
</body>
</html>

```

\*참고내용\* n 番目:nth-child(n) 짝수:nth-child(even) 또는 :nth-child(2n)  
 홀수:nth-child(odd) 또는 :nth-child(2n+ 1)  
 3 의배수 :nth-child(3n)  
 3 의배수-1 번째에 적용 :nth-child(3n-1)  
 4 번째 이후에 적용 :nth-child(n+ 4)

## 내용 정리

- 의사클래스는 선택자에 추가하는 키워드이다.
- 요소가 어떤 특정한 상태가 되었을 때 요소를 선택하겠다는 의미이다.
- 콘텐츠 자체적인 내용 뿐 아니라 사용자와의 상호작용과 관련된 경우에도 스타일을 적용할 수 있어 유용하다(hover, active 등).

-----

#17 의사 요소 (의사클래스와는 다름)

## 의사요소

의사요소(pseudo-elements)는 선택자에 추가하는 키워드로, 이를 이용하면 선택한 요소의 특정 부분에 대한 스타일을 정의할 수 있다.

-----

```
선택자::의사요소{
  속성명: 속성값;
}
```

-----

```
li::first-letter{
  font-size: 20px;
}
```

->li 요소의 첫 번째 글자만 크기를 20px 로 하겠다(기본값은 16px)

## 시습대상

### 의사요소

after        요소의 앞에 의사 요소를 생성 및 추가한다.

before      요소의 뒤에 의사 요소를 생성 및 추가한다.

first-line   블록 레벨 요소의 첫 번째 선에 스타일을 적용한다.

marker      목록 기호의 스타일을 적용한다.

placeholder   입력 요소의 플레이스홀더(자리표시자) 스타일을 적용한다.

사용 가능한 다른 의사 요소들 링크 참조!

<https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements>

\*첫째줄만 빨간색이며 앞머리 글자는 노란색 표현(first-line, first-letter)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

```

<meta charset="utf-8">
<title>HTML 문서</title>
<style>
    /*
    first-line marker placeholder before after
    */
    p::first-line{
        color:red;
    }
    p::first-letter{
        color:yellow;
    }
</style>
</head>
<body>
    <p>
        퍼스트 라인 그리고 퍼스트 레터
        퍼스트 라인 그리고 퍼스트 레터
        퍼스트 라인 그리고 퍼스트 레터
        퍼스트 라인 그리고 퍼스트 레터
    </p>
</body>
</html>

```

-----

\*(marker)블릿기호

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>HTML 문서</title>
        <style>
            /*
            first-line marker placeholder before after

```

```

    */
    p::first-line{
        color:red;
    }
    p::first-letter{
        color:yellow;
    }
    li::marker{
        color:teal;
        font-size:60px;
    }
</style>
</head>
<body>
    <ul>
        <li>강아지</li>
        <li>고양이</li>
        <li>다람쥐</li>
        <li>햄스터</li>
    </ul>
</body>
</html>

```

\*텍스트 박스 안쪽에 안내문구가 24 픽셀로 표현됨(placeholder)

```

        */
        input::placeholder{
            font-size:24px;
            color:red;
        }
    </style>
</head>
<body>
    <input type="text" placeholder="아무거나 쓰세요!">
</body>
</html>

```

-----

\*원래 쓰여있던 내용을 기준으로 before 로 앞에다 텍스트를 추가후 빨강색으로 표현  
 원래 쓰여있던 내용을 기준으로 after 로 뒤에다 텍스트를 추가후 초록색으로 표현

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>HTML 문서</title>
        <style>
            /*
            first-line marker placeholder before after
            */
            p::before{
                content:"앞에다 추가한 텍스트";
                color: red; font-weight: 900;
            }
            p::after{
                content:"뒤에다 추가한 텍스트";
                color: yellowgreen; font-weight:100;
            }
        </style>

```

```
</head>
<body>
  <p>원래 쓰여 있던 내용</p>
</body>
</html>
```

#### 내용 정리

- 의사요소는 선택자에 추가하는 키워드이다.
- 의사요소는 요소의 특정 부분에만 스타일을 적용하고자 할 때 사용한다.
- HTML 문서가 포함하지 않는 개체(목록 구분점, 플레이스홀더 등)를 선택할 수 있어 무척 유용하다.

-----