



웹 프로그래밍



학습 내용

1. 웹 표준 소개
2. HTML
3. CSS
4. JavaScript
5. jQuery

학습목표

- 선택자를 이해하고 다양한 선택자 표현을 익힌다.
- CSS 속성과 속성값의 의미를 알고 HTML에 적용한다.
- 내부, 외부 스타일 시트 적용 방법은 안다.

CSS 스타일 정의와 선택자

CSS란?

- Cascading Style Sheet는 HTML 등의 마크업 언어로 작성된 문서가 실제로 웹사이트에 표현되는 방법을 정해주는 언어



CSS3 기본 용어

- 선택자 : CSS3에서 특정 HTML 태그를 선택할 때 사용
- 속성 : HTML 요소에 적용할 수 있는 스타일 종류이며 값을 지정할 수 있음.
- CSS 적용 기본 문법

선택자(Selector) 속성(Property) 값(Value) 단위(Unit)

`div { width: 960px; }`

중괄호시작 (Opening Brace) 콜론(Colon) 세미콜론(Semi Colon) 중괄호끝 (Closing Brace)

A diagram illustrating the basic CSS syntax using the example `div { width: 960px; }`. Red arrows point from labels to specific parts of the code: '선택자(Selector)' points to 'div', '속성(Property)' points to 'width', '값(Value)' points to '960', '단위(Unit)' points to 'px', '중괄호시작 (Opening Brace)' points to '{', '콜론(Colon)' points to ':', '세미콜론(Semi Colon)' points to ';', and '중괄호끝 (Closing Brace)' points to '}'.

CSS 스타일 적용 방법

- 내부 스타일

- HTML 페이지 내부에서 style 태그를 사용해 스타일시트를 직접 입력
- 스타일시트가 짧은 경우

- 외부 스타일

- 스타일시트를 별도로 생성하고 link 태그의 href 속성을 사용해 불러옴
- 협업 업무나 프로젝트의 규모가 클 경우 사용

- 인라인 스타일

- 태그 안에 직접 지정
- `<p style='background-color:#00f; color:ff0'>`

HTML 페이지 구조 작성방법

- 내부 스타일 시트 작성

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML Intro</title>
  <style>
    h1{
      color : white;
      background: black;
    }
  </style>
</head>
<body>
  <h1>여러분을 환영합니다!!</h1>
</body>
</html>
```


HTML 페이지 구조 작성방법

- 외부 스타일 시트 작성 및 연결

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML Intro</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>여러분을 환영합니다!!</h1>
</body>
</html>
```

스타일 시트 파일

style.css

```
H1 {
  color: white;
  background: black;
}
```

HTML 페이지 구조 작성방법

- 내부 스타일

- HTML 페이지 내부에서 style 태그를 사용해 스타일시트를 직접 입력
- 스타일시트가 짧은 경우

- 외부 스타일

- 스타일시트를 별도로 생성하고 link 태그의 href 속성을 사용해 불러옴
- 협업 업무나 프로젝트의 규모가 클 경우 사용

- 인라인 스타일

- 태그 안에 직접 지정
- `<p style='background-color:#00f; color:ff0'>`

HTML 페이지 구조 작성방법

- 내부 스타일 시트 작성

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML Intro</title>
  <style>
    h1{
      color : white;
      background: black;
    }
  </style>
</head>
<body>
  <h1>여러분을 환영합니다!!</h1>
</body>
</html>
```

HTML 페이지 구조 작성방법

- 내부 스타일 시트 작성

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>HTML Intro</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>여러분을 환영합니다!!</h1>
</body>
</html>
```

스타일 시트 파일

style.css

```
H1 {
  color: white;
  background: black;
}
```

CSS 선택자

CSS 다양한 선택자

- 기본 선택자
- 복합 선택자
- 가상 클래스 선택자
- 가상 요소 선택자
- 속성

CSS – 기본 선택자

- Universal Selector

*

기본

전체 선택자 (Universal Selector)

모든 요소를 선택.

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li>오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span>오렌지</span>
</div>
```

선택

*

```
{
  color: ■ red;
}
```

CSS의 다양한 선택자

- 태그 선택자

ABC

기본

태그 선택자 (Type Selector)

태그 이름이 ABC인 요소 선택.

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li>오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span>오렌지</span>
</div>
```

선택

```
li {
  color: ■ red;
}
```


CSS의 다양한 선택자

- 클래스 선택자

.ABC

기본

클래스 선택자 (Class Selector)

HTML class 속성의 값이 ABC인 요소 선택.

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li class="orange">오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span class="orange">오렌지</span>
</div>
```

```
.orange {
  color: red;
}
```

CSS의 다양한 선택자

- ID 선택자

#ABC

기본


아이디 선택자 (ID Selector)

HTML id 속성의 값이 ABC인 요소 선택.

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li id="orange" class="orange">오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span class="orange">오렌지</span>
</div>
```

선택

```
#orange {
  color: red;
}
```

A curved arrow labeled '선택' (select) points from the CSS selector '#orange' to the HTML element '<li id="orange" class="orange">오렌지'. The HTML element is highlighted with a light gray background.

CSS의 다양한 선택자

- 일치 선택자

ABCXYZ

복합

일치 선택자 (Basic Combinator)

선택자 ABC와 XYZ를 동시에 만족하는 요소 선택.

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li class="orange">오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span class="orange">오렌지</span>
</div>
```

```
span.orange {
  color: ■ red;
}
```

선택

CSS의 다양한 선택자

- 자식 선택자

ABC > XYZ

복합

자식 선택자 (Child Combinator)

선택자 ABC의 자식 요소 XYZ 선택.

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li class="orange">오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span class="orange">오렌지</span>
</div>
```

```
ul > .orange {
  color: red;
}
```

선택

CSS의 다양한 선택자

- 후손선택자

ABC XYZ

복합

하위(후손) 선택자 (Descendant Combinator)

선택자 ABC의 하위 요소 XYZ 선택.
'띄어쓰기'가 선택자의 기호!

```
<div>
  <ul>
    <li>사과</li>
    <li>딸기</li>
    <li class="orange">오렌지</li>
  </ul>
  <div>당근</div>
  <p>토마토</p>
  <span class="orange">오렌지</span>
</div>
<span class="orange">오렌지</span>
```

선택

```
div .orange {
  color: ■ red;
}
```

CSS의 다양한 선택자

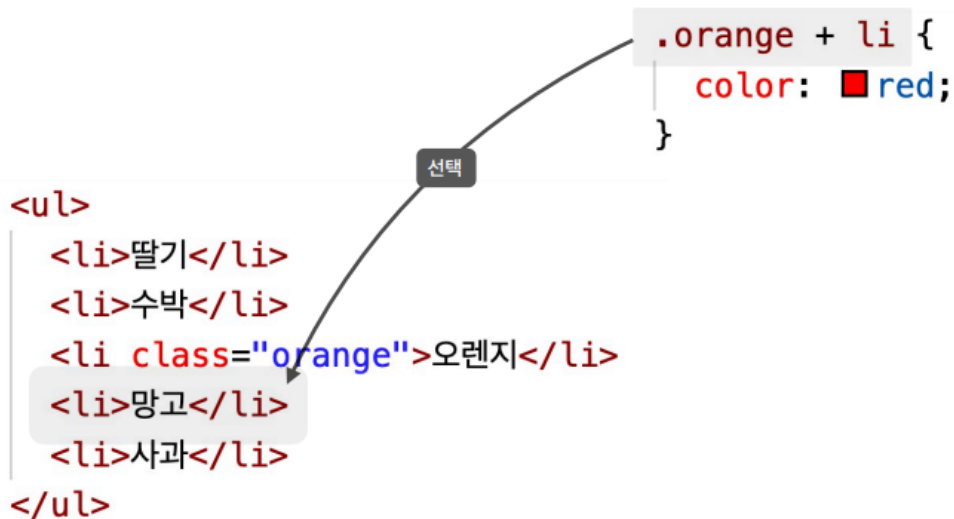
- 인접 형제 선택자

ABC + XYZ

복합

인접 형제 선택자 (Adjacent Sibling Combinator)

선택자 ABC의 다음 형제 요소 XYZ 하나를 선택.



CSS의 다양한 선택자

ABC ~ XYZ

복합

일반 형제 선택자 (General Sibling Combinator)

선택자 ABC의 다음 형제 요소 XYZ 모두를 선택.

```
<ul>
  <li>딸기</li>
  <li>수박</li>
  <li class="orange">오렌지</li>
  <li>망고</li>
  <li>사과</li>
</ul>
```

선택

```
.orange ~ li {
  color: ■ red;
}
```

CSS의 다양한 선택자

ABC: hover

가상 클래스 선택자 (Pseudo-Classes)

HOVER

선택자 ABC 요소에 마우스 커서가 올라가 있는 동안 선택.

화면에 출력!

NAVER

NAVER

선택

```
a:hover {  
  color: ■ red;  
}
```

```
<a href="https://www.naver.com">NAVER</a>
```


CSS의 다양한 선택자

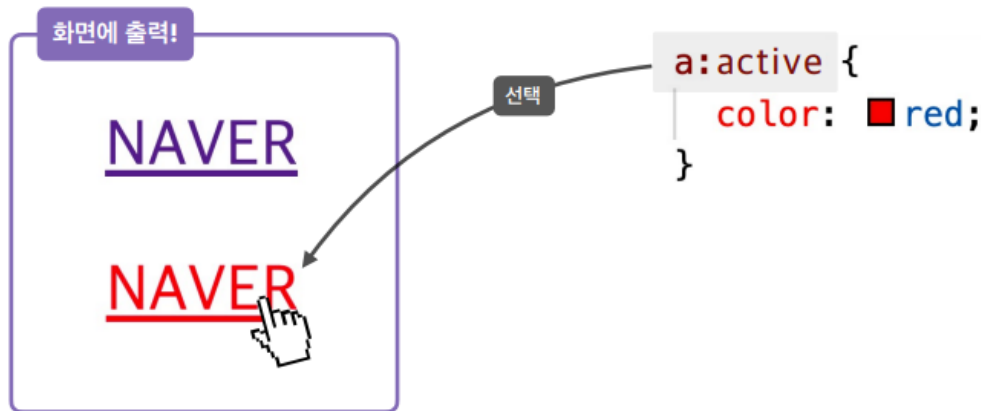
- 가상 클래스 선택자

ABC:active

가상 클래스 선택자 (Pseudo-Classes)

ACTIVE

선택자 ABC 요소에 마우스를 클릭하고 있는 동안 선택.



```
<a href="https://www.naver.com">NAVER</a>
```

CSS의 다양한 선택자

- 가상 클래스 선택자

ABC:focus

가상 클래스 선택자 (Pseudo-Classes)

FOCUS

선택자 ABC 요소가 포커스되면 선택.



```
input:focus {  
  background-color: orange;  
}
```

```
<input type="text" />
```

CSS의 다양한 선택자

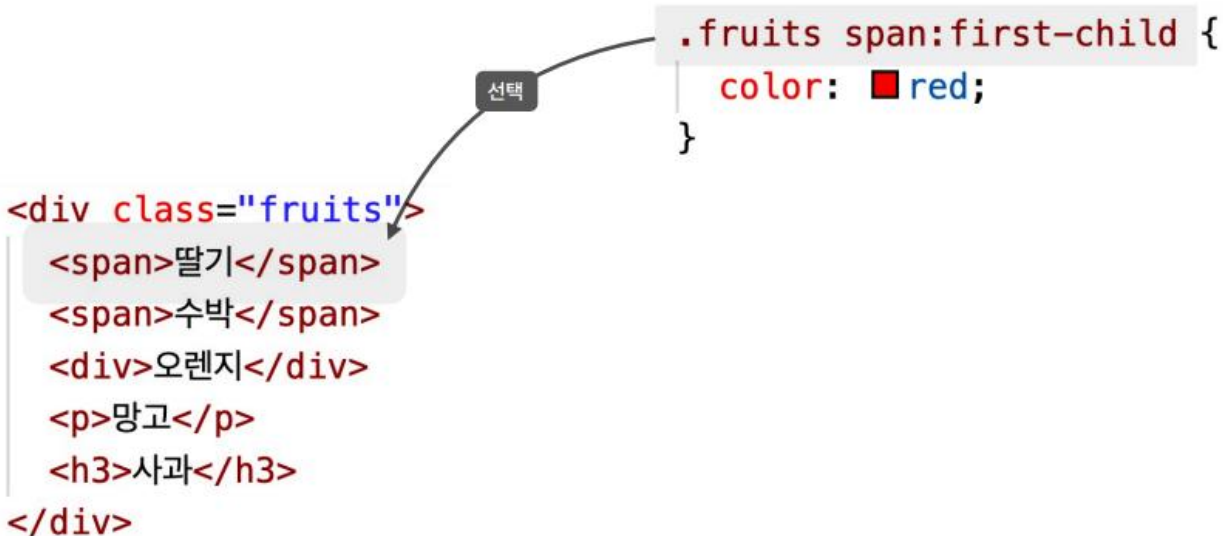
- 가상 클래스 선택자

ABC:first-child

가상 클래스 선택자 (Pseudo-Classes)

FIRST CHILD

선택자 ABC가 형제 요소 중 첫째라면 선택.



CSS의 다양한 선택자

- 가상 클래스 선택자

ABC:last-child

가상 클래스 선택자 (Pseudo-Classes)

LAST CHILD

선택자 ABC가 형제 요소 중 막내라면 선택.

```
<div class="fruits">
  <span>딸기</span>
  <span>수박</span>
  <div>오렌지</div>
  <p>망고</p>
  <h3>사과</h3>
</div>
```

선택

```
.fruits h3:last-child {
  color: red;
}
```

CSS의 다양한 선택자

- 가상 클래스 선택자

ABC:nth-child(n)

가상 클래스 선택자 (Pseudo-Classes)

NTH CHILD

선택자 ABC가 형제 요소 중 (n)째라면 선택.

```
<div class="fruits">
  <span>딸기</span>
  <span>수박</span>
  <div>오렌지</div>
  <p>망고</p>
  <h3>사과</h3>
</div>
```

선택

```
.fruits *:nth-child(2) {
  color: ■ red;
}
```

CSS의 다양한 선택자

- 가상 클래스 선택자

ABC:nth-child(n)

가상 클래스 선택자 (Pseudo-Classes)

NTH CHILD

선택자 ABC가 형제 요소 중 (n)째라면 선택.

```
<div class="fruits">
  <span>딸기</span>
  <span>수박</span>
  <div>오렌지</div>
  <p>망고</p>
  <h3>사과</h3>
</div>
```

선택

```
.fruits *:nth-child(2n) {
  color: red;
}
```

n은 0부터 시작!
(Zero-Based Numbering)

CSS의 다양한 선택자

- 가상클래스 선택자

ABC:nth-child(n)

가상 클래스 선택자 (Pseudo-Classes)

NTH CHILD

선택자 ABC가 형제 요소 중 (n)째라면 선택.

```
<div class="fruits">
  <span>딸기</span>
  <span>수박</span>
  <div>오렌지</div>
  <p>망고</p>
  <h3>사과</h3>
</div>
```

선택

```
.fruits *:nth-child(2n+1) {
  color: red;
}
```

n은 0부터 시작!
(Zero-Based Numbering)

CSS의 다양한 선택자

- 가상클래스 선택자

ABC:nth-child(n)

가상 클래스 선택자 (Pseudo-Classes)

NTH CHILD

선택자 ABC가 형제 요소 중 (n)째라면 선택.

```
<div class="fruits">
  <span>딸기</span>
  <span>수박</span>
  <div>오렌지</div>
  <p>망고</p>
  <h3>사과</h3>
</div>
```

선택

```
.fruits *:nth-child(n+2) {
  color: red;
}
```

n은 0부터 시작!
(Zero-Based Numbering)

CSS의 다양한 선택자

- 가상요소 선택자

ABC::before

인라인(글자) 요소

화면에 출력!

앞! Content!

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

Content!

```
</div>
```

가상 요소 선택자 (Pseudo-Elements)

BEFORE

선택자 ABC 요소의 내부 앞에 내용(Content)을 삽입.

```
.box::before {  
  content: "앞!";  
}
```

::before

CSS의 다양한 선택자

- 가상요소 선택자

ABC::after

인라인(글자) 요소

화면에 출력!

Content! 뒤!

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

Content!

```
</div>
```

가상 요소 선택자 (Pseudo-Elements)

AFTER

선택자 ABC 요소의 내부 뒤에 내용(Content)을 삽입.

```
.box::after {  
  content: "뒤!";  
}
```

::after

CSS의 다양한 선택자

- 속성 선택자

[ABC]

속성 선택자 (Attribute)

ATTR

속성 ABC를 포함한 요소 선택

```
[disabled] {  
  color: red;  
}
```

선택

```
<input type="text" value="HEROPY">  
<input type="password" value="1234">  
<input type="text" value="ABCD" disabled>
```

CSS의 다양한 선택자

- 속성 선택자

[ABC]

속성 선택자 (Attribute)

ATTR

속성 ABC를 포함한 요소 선택

선택

```
[type] {  
  color: red;  
}
```

```
<input type="text" value="HEROPY">  
<input type="password" value="1234">  
<input type="text" value="ABCD" disabled>
```

CSS의 다양한 선택자

- 속성 선택자

[ABC]

속성 선택자 (Attribute)

ATTR

속성 ABC를 포함한 요소 선택

선택

```
[type] {  
  color: red;  
}
```

```
<input type="text" value="HEROPY">  
<input type="password" value="1234">  
<input type="text" value="ABCD" disabled>
```

CSS의 다양한 선택자

- 속성 선택자

[ABC="XYZ"]

속성 선택자 (Attribute)

ATTR=VALUE

속성 ABC를 포함하고 값이 XYZ인 요소 선택.

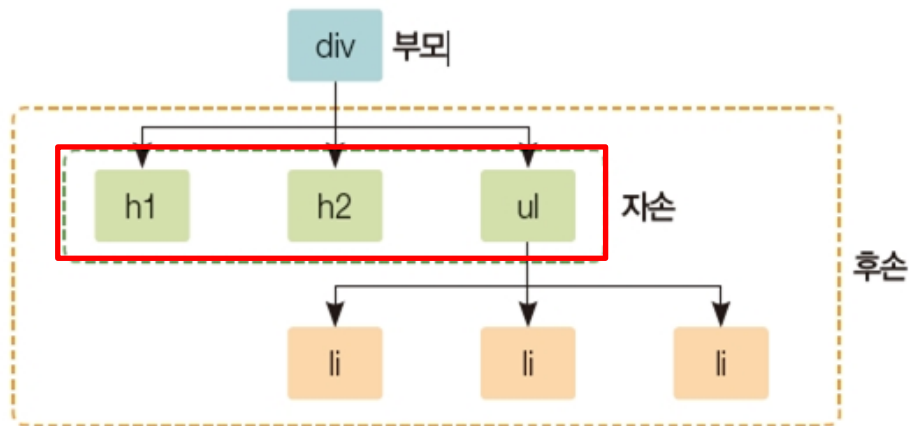
선택

```
[type="password"] {  
  color: red;  
}
```

```
<input type="text" value="HEROPY">  
<input type="password" value="1234">  
<input type="text" value="ABCD" disabled>
```

Html 문서 요소의 관계 이해

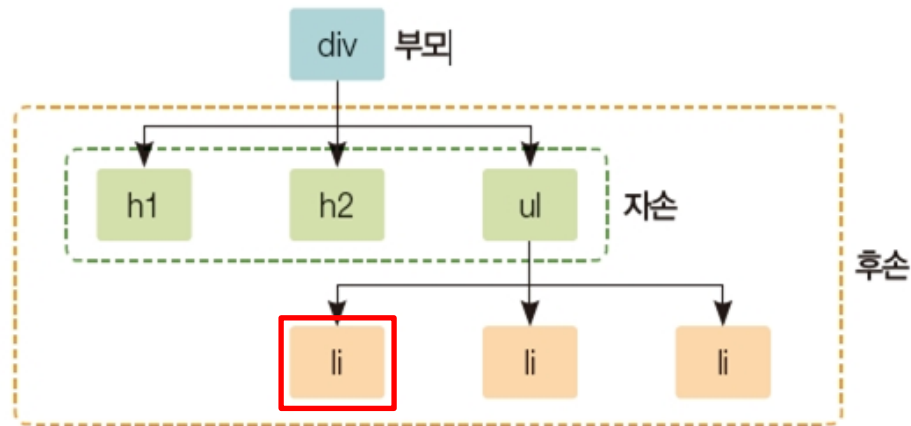
`<div>` 부모요소
`<div>` 자식(자손)요소
자식(자손)요소 `<h1>웹표준</h1>`
``
`html5`
`CSS3`
`Javascript`
``
`<h2>웹프레임워크</h2>`
`</div>`
`</div>`



자손 선택자와 후손 선택자

Html 문서 요소의 관계 이해

`<div>` ← 상위(조상) 요소
 `<div>` ←
 `<h1>웹표준</h1>`
 ``
 `html5`
 `CSS3`
 `Javascript`
 ``
 `<h2>웹프레임워크</h2>`
 `</div>`
`</div>`



자손 선택자와 후손 선택자

Html 문서 요소의 관계 이해

`<div>` 하위(후손) 요소

`<div>`

`<h1>웹표준</h1>`

``

`html5`

`CSS3`

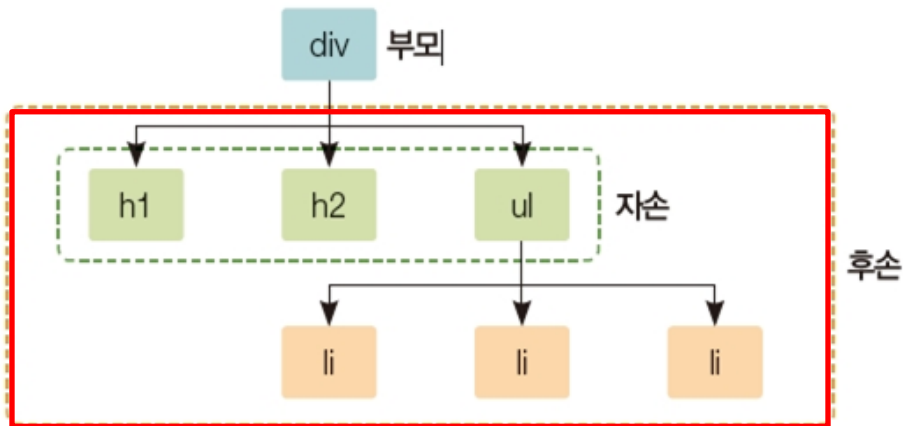
`Javascript`

``

`<h2>웹프레임워크</h2>`

`</div>`

`</div>`



자손 선택자와 후손 선택자

Html 문서 요소의 관계 이해

<div>

<div>

<h1>웹표준</h1>

html5

이전 형제 요소

CSS3

형제 요소

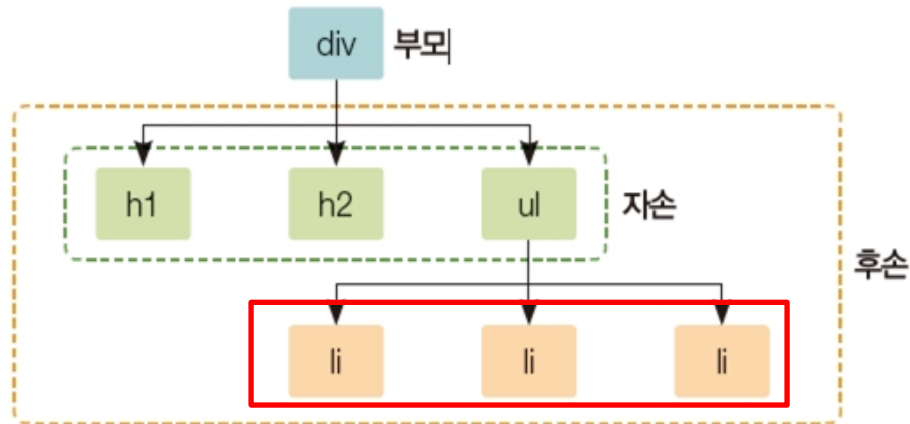
Javascript

다음 형제 요소

</div>

<h2>웹프레임워크</h2>

</div>



자손 선택자와 후손 선택자

기본 선택자

- 여러 선택자에 속성 적용하기

```
<style>
```

```
body, p, h1, h2, h3, h4, h5, h6 { margin: 0; padding: 0}
```

```
</style>
```

- ID 선택자에 속성 적용하기

```
<p id="title">웹 표준 이해</p>
```

```
#title { font-size : 20px}
```

- Class 선택자에 속성 적용하기

```
<p class="title">웹 표준 이해</p>
```

```
.title { font-size : 20px}
```

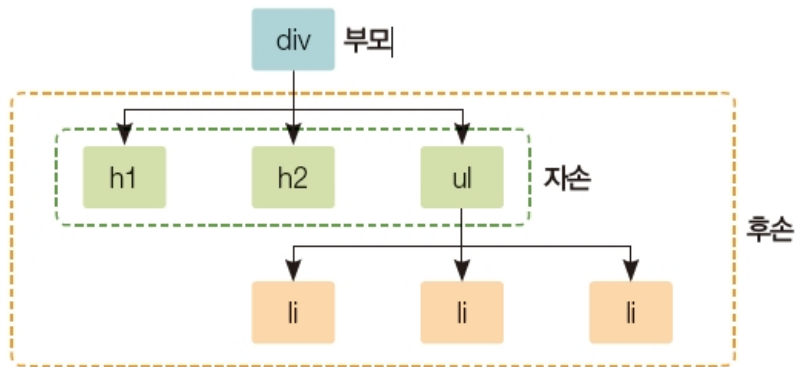
후손 선택자와 자손 선택자

[html]

```
<div id="header">
  <h1 class="title">CSS3 선택자 기본</h1>
  <h2 class="sub_title">기본 선택자</h2>
  <ul class="list">
    <li>전체 선택자</li>
    <li>태그 선택자</li>
    <li>아이디 선택자와 클래스 선택자</li>
  </ul>
</div>
```

[css]

```
<style>
  #header .title { color: red; }
  #header li { color: orange; }
</style>
```



자손 선택자와 후손 선택자

- 후손 선택자 표시
div .list li
div#header li
#header ul li

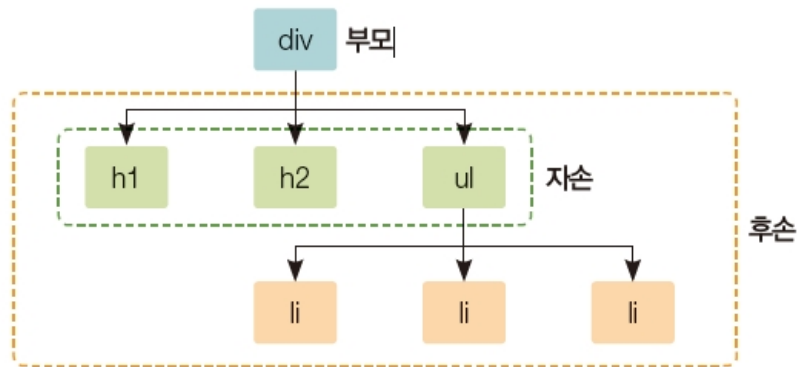
후손 선택자와 자손 선택자

[html]

```
<div id="header">
  <h1 class="title">CSS3 선택자 기본</h1>
  <h2 class="sub_title">기본 선택자</h2>
  <ul class="list">
    <li>전체 선택자</li>
    <li>태그 선택자</li>
    <li>아이디 선택자와 클래스 선택자</li>
  </ul>
</div>
```

[css]

```
<style>
  #header > h1 { color: red; }
  #header > ul.list > li { color: orange; }
</style>
```



자손 선택자와 후손 선택자

- 자손선택자 표시
div#header > .list > li
#header > ul > li

후손 선택자와 자손 선택자

- 후손 선택자 여러 개를 함께 사용할 경우 주의 사항

- ① div 태그의 후손인 h1 태그와 일반적인 h2 태그 선택

```
<style>  
  div h1, h2 {color : red;}  
</style>
```

- ① div 태그의 후손인 h1 태그와 div 태그의 후손인 h2 태그 선택

```
<style>  
  div h1, div h2 {color : red;}  
</style>
```

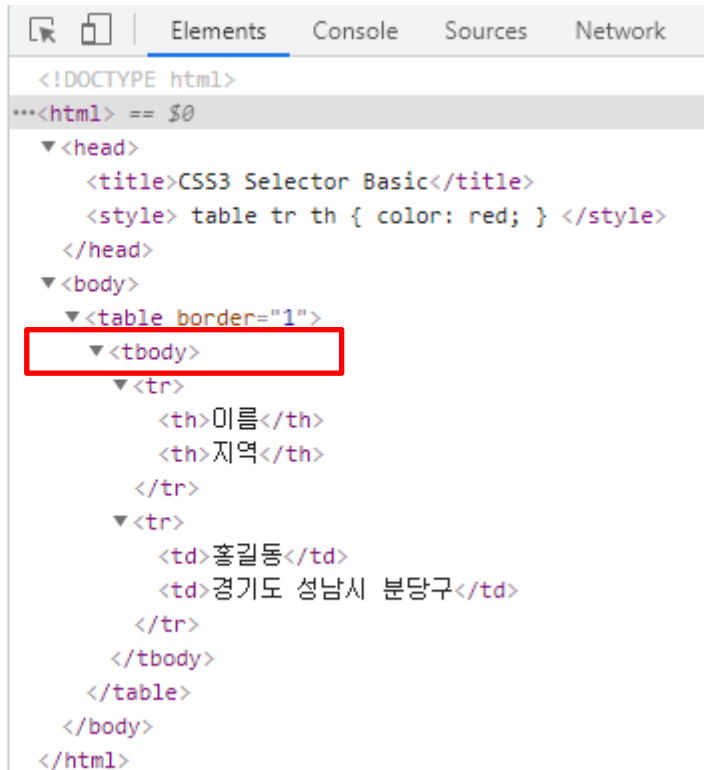
후손 선택자와 자손 선택자

- <table> 태그 요소 선택할 때 자손 선택자 주의 사항

이름	지역
홍길동	경기도 성남시 분당구

html

```
<body>
  <table border="1">
    <tr>
      <th>이름</th>
      <th>지역</th>
    </tr>
    <tr>
      <td>홍길동</td>
      <td>경기도 성남시 분당구</td>
    </tr>
  </table>
</body>
```



속성 선택자

형태	설명
선택자[속성]	특정한 속성이 있는 태그 선택
선택자[속성=값]	속성 내부 값이 지정 값과 같은 태그 선택

[html]

```
<form>
  <input type="text" />
  <input type="password" />
</form>
```

[css]

```
<style>
  input[type="text"] { background: tomato; }
  input[type="password"] { background: lightblue; }
</style>
```


반응 선택자

형태	설명
:active	사용자가 마우스로 클릭한 태그 선택
:hover	사용자가 마우스 커서를 올린 태그 선택

```
<head>
  <title>CSS3 Selector Basic</title>
  <style>
    h1:hover { color: red; }
    h1:active { color: blue; }
  </style>
</head>
<body>
  <h1>마우스를 올리면 글자색상이 바뀜</h1>
</body>
```

마우스를 올리면 글자색상이 바뀜
마우스를 올리면 글자색상이 바뀜

상태 선택자

형태	설명
:checked	체크 상태의 input 태그 선택
:focus	포커스를 맞춘 input 선택 태그
:enabled	사용 가능한 input 태그 선택
:disabled	사용 불가능한 input 태그 선택

```
<body>
  <h2>사용 가능</h2>
  <input />
  <h2>사용 불가능</h2>
  <input disabled="disabled"/>
</body>
```

```
<style>
  input:enabled { background-color: white; }
  input:disabled { background-color: gray; }
  input:focus { background-color: orange; }
</style>
```

구조 선택자

형태	설명
: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-child와 nth-last-child 선택자
 - $2n+1$ 수열에 0부터 숫자를 넣을 경우... "1, 3, 5, 7, 9....? "
 - 구조 선택자는 수열의 결과 값에 해당하는 태그에 스타일을 적용함

구조 선택자

- 구조 선택자 사용시 주의 사항

```
<head>
  <title>CSS3 Selector Basic</title>
  <style>
    li > a:first-child { color: red; }
  </style>
</head>
<body>
  <ul>
    <li><a href="#">구조선택자 지정 주의사항</a></li>
    <li><a href="#">구조선택자 지정 주의사항</a></li>
    <li><a href="#">구조선택자 지정 주의사항</a></li>
    <li><a href="#">구조선택자 지정 주의사항</a></li>
    <li><a href="#">구조선택자 지정 주의사항</a></li>
  </ul>
</body>
```

- 구조선택자 지정 주의사항
- 구조선택자 지정 주의사항
- 구조선택자 지정 주의사항
- 구조선택자 지정 주의사항
- 구조선택자 지정 주의사항

CSS 스타일 속성 적용하기

학습내용

박스 모델

Display, inline

배경

글꼴, 문자

레이아웃

정렬

플렉스(정렬)

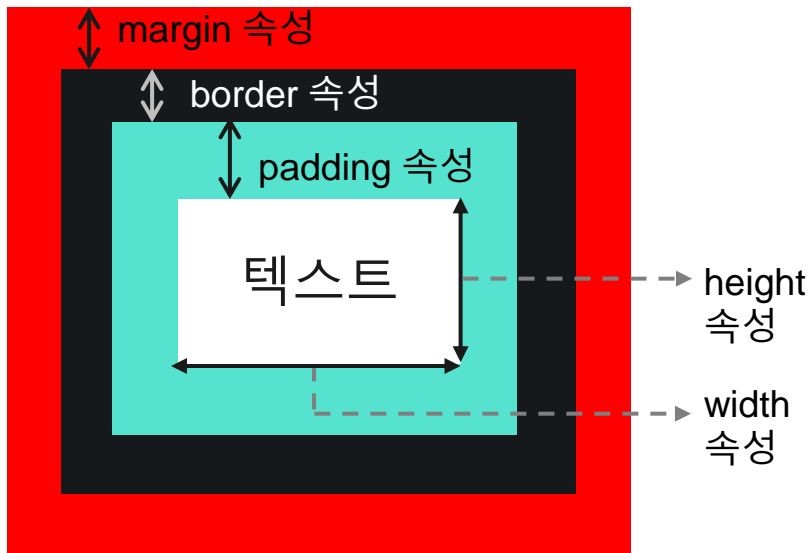
전환, 변환

다양한 단위

반응형 레이아웃

박스 모델 속성

- 웹 페이지의 레이아웃을 구성할 때 가장 중요



속성	설명
margin	요소 box의 바깥쪽 여백
border	테두리
padding	테두리 안쪽 여백, 배경색 적용
width	요소의 가로 영역의 너비
height	요소의 세로 영역의 높이

박스 속성

- 박스 크기와 여백, 테두리 속성
 - 박스 크기 : width, height
 - 테두리 스타일 지정 : border
 - 바깥쪽 여백 : margin, 안쪽 여백 : padding



```
<body>
  <div class="box1">
    <div class="box2">
      <div class="box3">
        <p>마진, 패딩, 여백 알아보기</p>
      </div>
    </div>
  </div>
</body>
```

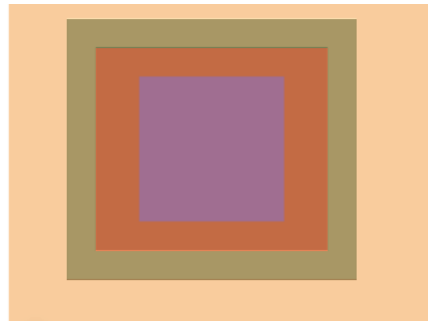
```
div {
  width: 100px; height:100px;
  background-color: red;
  border: 20px solid black;
  margin: 10px; padding:30px;
}

div.box3{
  padding : 20px;
  border: 10px dashed rgb(0, 0, 0);
}
```


박스 여백 속성

- margin 속성과 padding 속성 지정 방법 3가지
 - margin: 30px;, margin: 20px 30px;, margin: 10px 20px 0px 40px;

여백 위치	속성
바깥쪽 여백	margin-top
	margin-right
	margin-bottom
	margin-left
안쪽 여백	padding-top
	padding-right
	padding-bottom
	padding-left



div200 × 180

Background

#FF0000

Margin10px 20px 30px 40px

Padding20px 30px

ACCESSIBILITY

Name

Rolegeneric

Keyboard-focusable

ElementsConsoleSourcesNetwork

htmlbodydiv

StylesComputedLayoutEvent ListenersDOM Breakpoints

Filter

element.style {

}

div {

width: 100px;

height: 100px;

background-color: red;

margin: 10px 20px 30px 40px;

padding: 20px 30px;

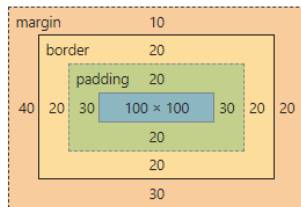
border: 20px solid black;

}

div {

display: block;

}



박스 테두리 속성

- border-width : 테두리 두께
- border-style : 테두리 형태
- border-color : 테두리 색상

[결과]

border test1

border test2

border test3

```
div.box1 {  
    margin: 10px;  
    border-width: thick;  
    border-style: dashed;  
    border-color: black;  
    border-radius: 20px;  
}  
div.box2 {  
    margin: 10px;  
    border: thick dashed rgb(141, 14, 14);  
    border-radius: 20px 0px;  
}  
div.box3 {  
    margin: 10px;  
    border: thick dotted rgb(3, 110, 114);  
    border-radius: 20px 20px 0px 0px;  
}
```

display 속성

- 요소가 화면에 보이는 방식을 지정
 - 속성 값 none, block, inline, inline-block
 - 예) display: block;
- 블록 (block) 요소 태그
 - div, h1~h6, p, ul, pre, header, footer, audio, video article, aside
 - vertical-align 적용되지 않음.
 - 인라인 요소와 일부 블록요소 포함
- 인라인(inline) 요소 태그
 - span, a, b, button, i, em, select, sub, sup, textarea, input, label, small
 - height, width 요소가 적용되지 않음.
 - 블록요소에 포함됨.

display 속성

- display: inline-block;
- 인라인 속성과 블록 속성을 가짐

```
<body>
  <span class="txt">인라인 요소</span>
  <span class="txt">인라인 요소</span>
  <div id="box">블록요소</div>
  <div id="box">블록요소</div>
</body>
```

인라인 요소 인라인 요소
블록요소
블록요소

* 스타일 적용 전

* 스타일 적용 후

인라인 요소

인라인 요소

```
<style>
  #box {
    width:200px; height: 200px;
    display: inline-block;
  }
  .txt{
    width:200px; height: 200px;
    display: inline-block;
  }
</style>
```

블록요소

블록요소

배경 속성

- 배경 속성

속성	설명
background-image	배경 이미지 삽입
background-size	배경 이미지의 크기 지정
background-repeat	배경 이미지의 반복 형태 지정
background-attachment	배경 이미지의 부착 형태 지정
background-position	배경 이미지의 위치 지정
background	한 번에 모든 배경 속성 입력

배경 속성

- background 이미지 추가

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

```
<style>
  body {
    background-image: url('./imgs/BackgroundFront.png'),
                      url('./imgs/BackgroundBack.png');
  }
</style>
```

배경 속성

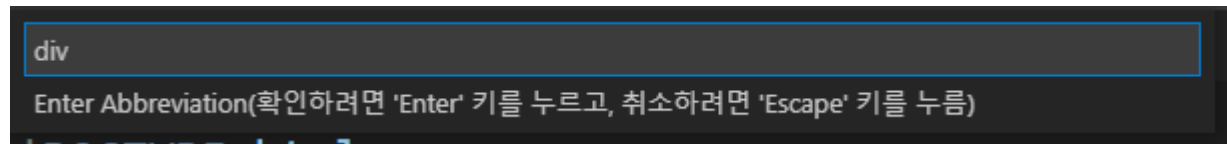
- 배경 이미지 반복형태 지정
 - background-size 속성
 - background-repeat 속성
 - repeat-x : X축 방향으로만 이미지 반복
 - repeat-y : Y축 방향으로만 이미지 반복

- 배경 이미지 화면 고정 형태
 - background-attachment: 속성값;
 - Ex) Background-attachment: scroll;
- 배경 이미지 위치의 배치 위치
 - background-position: 속성값;
 - background-position: bottom;
 - background-position: X축 위치;
 - background-position: X축 위치 Y축 위치;

잠깐 쉬었다 가기

- vscode 단축기

*약어로 wrapping : ctrl+shift+p



글자 속성

- 글자 크기와 글꼴
- generic-family – 다국적 사용자 대상용 글꼴

사용자 OS에 설치된 폰트를
범용 폰트 패밀리로 구분하여 관리함

```
div p {  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 12pt;  
  font-style: italic;  
}
```

[크롬-설정-글꼴-글꼴 맞춤설정] 내용

표준 글꼴

Malgun Gothic

16: 재빠른 갈색 여우가 게으른 개를 뛰어넘습니다.

Serif

Batang

16: 재빠른 갈색 여우가 게으른 개를 뛰어넘습니다.

Sans-serif

Malgun Gothic

16: 재빠른 갈색 여우가 게으른 개를 뛰어넘습니다.

고정폭 글꼴

맞춤설정

13: 재빠른 갈색 여우가 게으른 개를 뛰어넘습니다.

글자 속성

- OS의 기본 글꼴 적용 및 글자 크기

선택자 {

font-family : "

}

```
div p {  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 12pt;  
  font-style: italic;  
}
```

외부 폰트 적용하기

- 구글 폰트 URL 링크 적용하기

```
@import url('https://fonts.googleapis.com/css2?family=Single+Day&display=swap');
```

<https://fonts.google.com/>

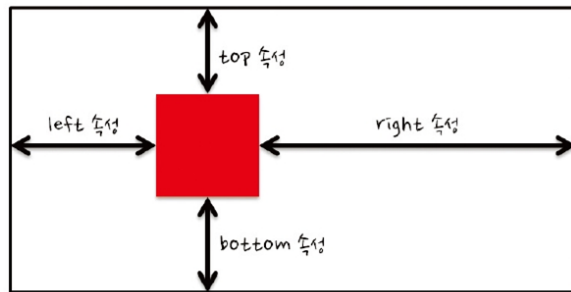
위치 속성

- 요소의 고정 위치와 상대 위치 지정

- osition – 요소의 위치 지정 형식 설정

- 절대 위치 좌표 : absolute, fixed

- 상대 위치 좌표 : relative, static



position 속성과 함께
사용하는 스타일 속성

속성	설명
absolute	절대 위치 좌표 설정
fixed	화면을 기준으로 절대 위치 자료 설정
relative	초기 위치에서 상하좌우로 위치 이동
static	위쪽에서 아래쪽으로 순서대로 배치

위치 속성

- Position : absolute와 relative 속성 값의 차이 이해

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

```
.box {
  width: 100px; height: 100px;
  position: absolute;
  /* position: relative; */
}

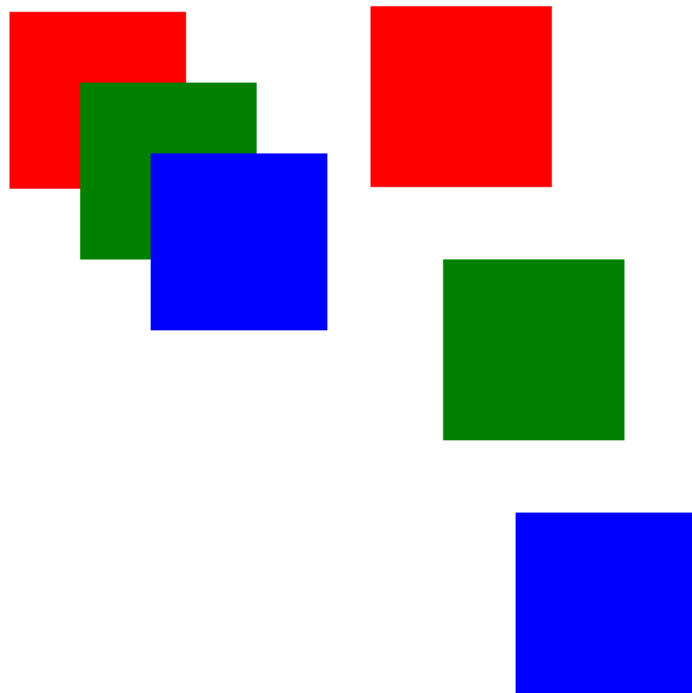
.box:nth-child(1) { background-color: red; }
.box:nth-child(2) { background-color: green; }
.box:nth-child(3) { background-color: blue; }
```



위치 속성

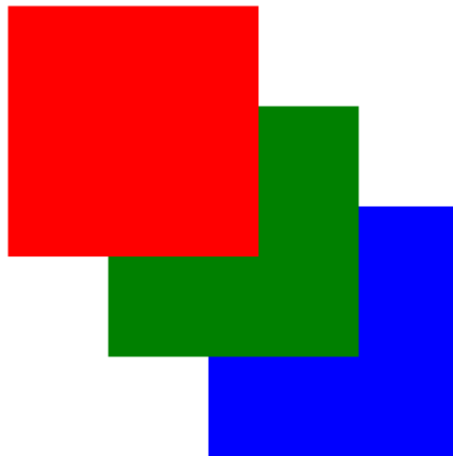
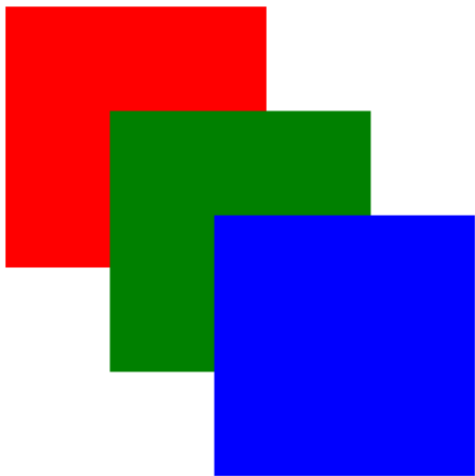
- position : absolute 와 relative
 - left 속성과 top 속성 적용하기

```
.box {  
  width: 100px; height: 100px;  
  position: absolute;  
  /* position: relative; */  
.box:nth-child(1) {  
  background-color: red;  
  left: 10px; top: 10px; }  
.box:nth-child(2) {  
  background-color: green;  
  left: 50px; top: 50px; }  
.box:nth-child(3) {  
  background-color: blue;  
  left: 90px; top: 90px; }
```



위치 속성

- position의 레이어 순서 변경 속성
 - z-index 속성 적용 : 숫자를 입력하면 숫자가 클수록 앞에 위치



z-index: 100

z-index: 10

z-index: 1

위치 속성

• 위치 속성 공식 이해

HTML 이해

하이퍼텍스트 마크업 언어(HyperText Markup Language, HTML, 문
화어: 초본문표식달기언어, 하이퍼본문표식달기언어)는 웹 페이지를
위한 지배적인 마크업 언어다. ... HTML은 웹 페이지 콘텐츠 안의 꺾
쇠 괄호에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한
다.

CSS 이해

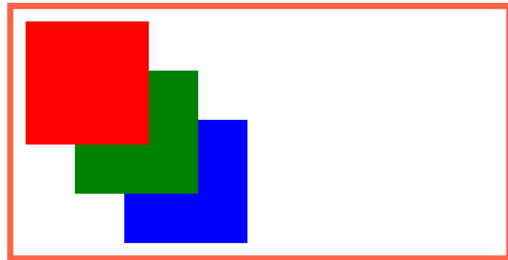
종속형 시트 또는 캐스케이딩 스타일 시트(Cascading Style Sheets,
CSS)는 마크업 언어가 실제 표시되는 방법을 기술하는 언어로,
HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있다. W3C
의 표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높다.

해결 방법

“자손 position 속성에 absolute 키워드를 적용하려면
부모 position 속성에 relative 키워드를 적용.”

HTML 이해

하이퍼텍스트 마크업 언어(HyperText Markup Language, HTML, 문
화어: 초본문표식달기언어, 하이퍼본문표식달기언어)는 웹 페이지를
위한 지배적인 마크업 언어다. ... HTML은 웹 페이지 콘텐츠 안의 꺾
쇠 괄호에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한
다.



CSS 이해

종속형 시트 또는 캐스케이딩 스타일 시트(Cascading Style Sheets,
CSS)는 마크업 언어가 실제 표시되는 방법을 기술하는 언어로,
HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있다. W3C의
표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높다.

위치 속성

- overflow 속성

- 내용이 요소 크기를 벗어났을 때 어떻게 보여 줄지 지정
- float 속성과 함께 자주 사용

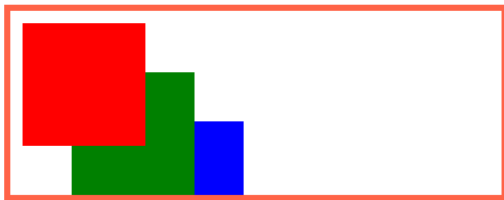
속성	설명
hidden	영역을 벗어나는 부분 감춤
scroll	영역을 벗어나는 부분을 스크롤로 만듦
auto	영역을 벗어나면 자동으로 스크롤이 생김

위치 속성

overflow: hidden;

HTML 이해

하이퍼텍스트 마크업 언어(HyperText Markup Language, HTML, 문화어: 초본문표식 달기언어, 하이퍼본문표식달기언어)는 웹 페이지를 위한 지배적인 마크업 언어다. ... HTML은 웹 페이지 콘텐츠 안의 꺾쇠 괄호에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한다.



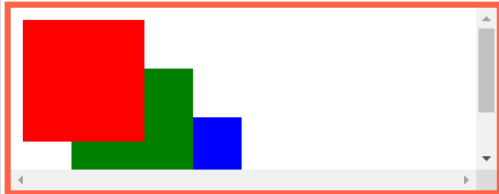
CSS 이해

종속형 시트 또는 캐스케이딩 스타일 시트(Cascading Style Sheets, CSS)는 마크업 언어가 실제 표시되는 방법을 기술하는 언어로, HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있다. W3C의 표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높다.

overflow: scroll;

HTML 이해

하이퍼텍스트 마크업 언어(HyperText Markup Language, HTML, 문화어: 초본문표식 달기언어, 하이퍼본문표식달기언어)는 웹 페이지를 위한 지배적인 마크업 언어다. ... HTML은 웹 페이지 콘텐츠 안의 꺾쇠 괄호에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한다.



CSS 이해

종속형 시트 또는 캐스케이딩 스타일 시트(Cascading Style Sheets, CSS)는 마크업 언어가 실제 표시되는 방법을 기술하는 언어로, HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있다. W3C의 표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높다.

float 속성

- float 속성

- 웹 페이지의 레이아웃을 잡을 때 많이 사용
- 웹 브라우저 크기에 상관없이 공지 등을 일정한 위치에 고정할 때 적합
- 규칙 : 자손에게 float 속성을 지정하고, 부모의 overflow 속성에는 hidden 키워드 적용

다음을 시작페이지로 ▶



사회적 거리두기 야인이즈백 킹두한 판매재고관리 전효성 소감 하루살이퇴치법



카페 메일 뉴스 지도 증권 쇼핑 카카오톡 웹툰 블로그 브런치 사전 게임 같이가치 더보기 ▾

div 태그 내 중앙 정렬

- 브라우저 기준으로 중앙 정렬 방법
- div 태그 내에서 중앙 정렬 방법

다음은 시작페이지로 ▶



사회적 거리두기 야인이즈백 킹두한 판매재고관리 전효성 소감 하루살이퇴치법



카페 메일 뉴스 지도 증권 쇼핑 카카오톡 웹툰 블로그 브런치 사전 게임 같이가치 더보기 ▼

문제해결

- 3-24 예제를 기본으로 다음 부분 완성하기
- imgs/daum.png 활용



사회적 거리두기 야인이즈백 킹두한 판매재고관리 전효성 소감 하루살이퇴치법

카페 메일 뉴스 지도 증권 쇼핑 카카오TV 웹툰 블로그 브런치 사전 게임 같이가치 더보기 ▾

수평 정렬, 수직 정렬 레이아웃

- 콘텐츠 영역 레이아웃 스타일 적용



사회적 거리두기 야인이즈백 킹두한 판매 재고관리 전효성 소감 하루상이 퇴치법

카페 메일 뉴스 지도 증권 쇼핑 카카오톡 웹툰 블로그 브런치 사전 게임 같이가치 더보기

HTML5 웹표준 이해

하이퍼텍스트 마크업 언어(HyperText Markup Language, HTML, 문화어: 초본문표식달기언어, 하이퍼본문표식달기언어)는 웹 페이지를 위한 지배적인 마크업 언어다. ... HTML은 웹 페이지 콘텐츠 안의 꺾쇠 괄호에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한다.

HTML 이해

하이퍼텍스트 마크업 언어(HyperText Markup Language, HTML, 문화어: 초본문표식달기언어, 하이퍼본문표식달기언어)는 웹 페이지를 위한 지배적인 마크업 언어다. ... HTML은 웹 페이지 콘텐츠 안의 꺾쇠 괄호에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한다.

CSS 이해

중속형 시트 또는 캐스케이딩 스타일 시트(Cascading Style Sheets, CSS)는 마크업 언어가 실제 표시되는 방법을 기술하는 언어로, HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있다. W3C의 표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높다.

javascript 이해

자바스크립트(영어: JavaScript)는 객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹 브라우저 내에서 주로 사용하며, 다른 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고 있다. 또한 Node.js와 같은 런타임 환경과 같이 서버 프로그래밍에도 사용되고 있다.

절대위치를 활용한 요소배치

- Parent 요소
`position : relative ;`

- 자식요소
`position : absolute ;`

HTML 이해

하이퍼텍스트 마크업 언어(HyperText Markup Language, HTML, 문화어: 초본문 표식달기언어, 하이퍼본문표식달기언어)는 웹 페이지를 위한 지배적인 마크업 언어다. ... HTML은 웹 페이지 콘텐츠 안의 꺾쇠 괄호에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한다.

CSS 이해

중속형 시트 또는 캐스캐이딩 스타일 시트(Cascading Style Sheets, CSS)는 마크업 언어가 실제 표시되는 방법을 기술하는 언어로, HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있다. W3C의 표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높다.

javascript 이해

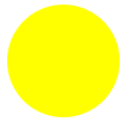
자바스크립트(영어: JavaScript)는 객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹 브라우저 내에서 주로 사용하며, 다른 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고 있다. 또한 Node.js와 같은 런타임 환경과 같이 서버 프로그래밍에도 사용되고 있다.

동그라미 만들고 절대 위치 지정하기



`left : 20px ; top : 20px`

`right : 20px ; top : 20px`



`left : 20px ; bottom : 20px`

`right : 20px ; bottom : 20px`



글자 생략

한줄일 때 끝부분

```
white-space: nowrap;  
overflow: hidden;  
text-overflow: ellipsis;
```

multiline일 때 끝부분

```
overflow: hidden;  
text-overflow: ellipsis;  
display: -webkit-box;  
-webkit-line-clamp: 5; /* 라인수 */  
-webkit-box-orient: vertical;  
word-wrap: break-word;
```

HTML 이해

하이퍼텍스트 마크업 언어(HyperText Markup Language, HTML, 문화어: 초본문표식달기언어, 하이퍼본문표식달기언어)는 웹 페이지를 위한 지배적인 마크업 언어다. ... HTML은 웹 페이지 콘텐츠 안의 객체 괄호...

CSS 이해

종속형 시트 또는 캐스케이딩 스타일 시트(Cascading Style Sheets, CSS)는 마크업 언어가 실제 표시되는 방법을 기술하는 언어로, HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있다. W3C의 표준이며, 레이아웃...

javascript 이해

자바스크립트(영어: JavaScript)는 객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹 브라우저 내에서 주로 사용하며, 다른 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고 있다. 또한 Node.js와 같은 ...

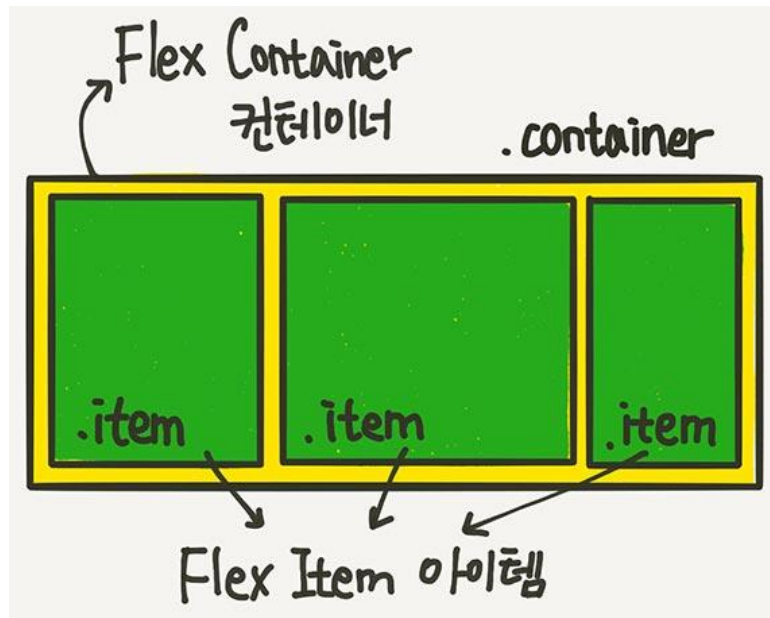
Flex 기능

- html

```
<div class="container">  
  <div class="item">helloflex</div>  
  <div class="item">abc</div>  
  <div class="item">helloflex</div>  
</div>
```

- CSS

```
.container {  
  display: flex;  
  /* display: inline-flex; */  
}
```



참고 : <https://studiomeal.com/archives/197>

CSS3 단위

- 키워드 단위

- W3C에서 미리 정의한 단어

- 키워드를 입력하면 해당하는 스타일이 자동으로 적용

속성	설명
pt(point)	1 포인트는 1/72 인치
%	백분율 단위
em	배수 단위
px(pixel)	픽셀 단위

1배=1em=100%

1.5배=1.5em=150%

초기 설정 크기: 100%

CSS3 단위

색상 단위

RGB색상	형태	rgb(red, green, blue)
	설명	- RGB 색상을 조합해 색상을 표현 - 0~255 사이의 숫자 입력
	예	div {background-color: rgb(255, 255, 255); }
RGBA색상	형태	rgba(red, green, blue, alpha)
	설명	- RGB 색상단위에 알파 값을 추가한 형태 - 알파 값은 투명도를 나타냄 - 0.0~1.0 사이 숫자 입력 - 0.0 : 완전 투명, 1.0 : 완전 불투명
	예	div {background-color: rgba(255, 255, 255, 0.5); }
HEX코드	형태	#0094FF;
	설명	- RGB 색상 단위를 16진수로 조합해서 표현
	예	div {background-color: #0094FF; }

4. 반응형 레이아웃 만들기

반응형 웹 레이아웃 1

[참고]

https://www.w3schools.com/css/tryit.asp?filename=trycss_website_layout_blog

반응형 레이아웃 2

[참고]

https://www.w3schools.com/css/css_rwd_videos.asp

HTML, CSS 정리 프로젝트(개인별)

- 주제 : 자신을 소개하는 페이지 만들기
- 2페이지 이상 만들기
- 반응형 레이아웃으로 만들기
- 자신을 소개하는 페이지
 - 텍스트, 이미지, 영상 등 다양하게 넣어도 됨
- 현재의 역량, 자신의 꿈과 계획 넣기
- **금요일 오후에 발표 – HTML, CSS 적용 등**