

# 내부 스타일

# 외부 스타일

# 선택자

# Css3 형식

- 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



# Css3 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

# Css 정의 문법

- CSS의 정의 문법



*/\* 한 줄로 작성 \*/*

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

*/\* 여러 줄로 작성 \*/*

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

# 내부 스타일과 외부스타일

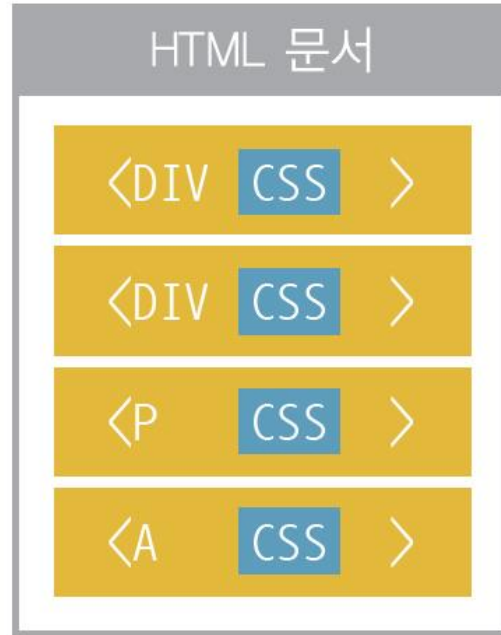
- 내부 스타일 시트

- `</head>` 태그가 나타나기 전에, `<style>` 태그와 `</style>` 태그 사이에 스타일 규칙 기록

- 외부 스타일시트

- 스타일 정보를 별도의 파일로 저장. 파일 확장자 : \*.css
  - `<link>` 태그를 사용해 웹 문서에 링크
  - `<link href="외부 스타일 파일 경로" rel="stylesheet" >`

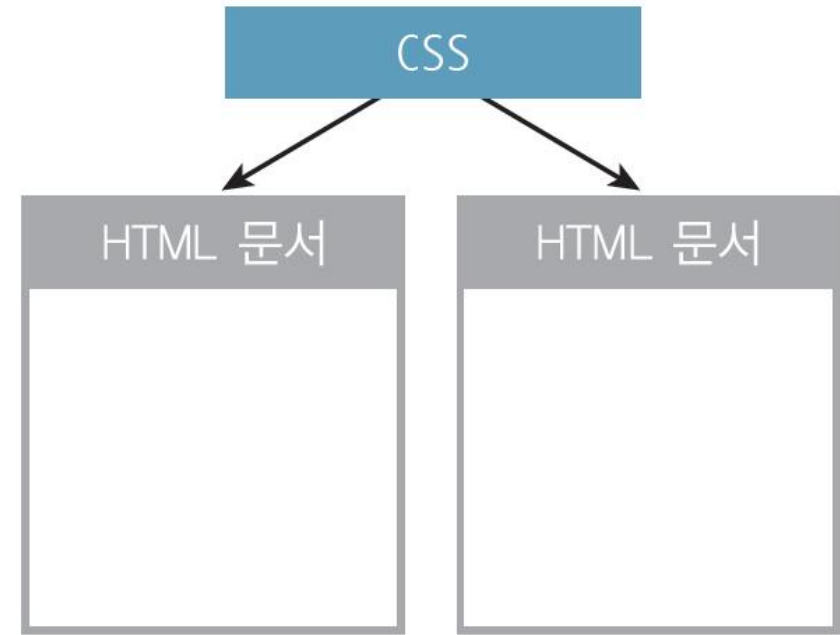
# 내부 스타일과 외부 스타일형식



인라인 스타일 시트



내부 스타일 시트



외부 스타일 시트

# 내부 스타일과 외부 스타일

```
h3 {  
  color: green;  
  background-color: yellow;  
}
```

```
<head>  
  <!-- 외부 스타일 시트 적용 방법 -->  
  <link rel="stylesheet" href="css/style.css">  
  <style>  
    h2 { /* 내부 스타일 시트 적용 방법 */  
      color: blue;  
    }  
  </style>  
</head>  
<body>  
  <!-- 인라인 스타일 시트 적용 방법 -->  
  <p style="color: red;">인라인 스타일 시트 적용</p>  
  <h2>내부 스타일 시트 적용</h2>  
  <h3>외부 스타일 시트 적용</h3>  
</body>
```

인라인 스타일 시트 적용  
내부 스타일 시트 적용  
외부 스타일 시트 적용

# 초기화 스타일 만들기

초기화 파일을 만들어서 외부 css 연동하기

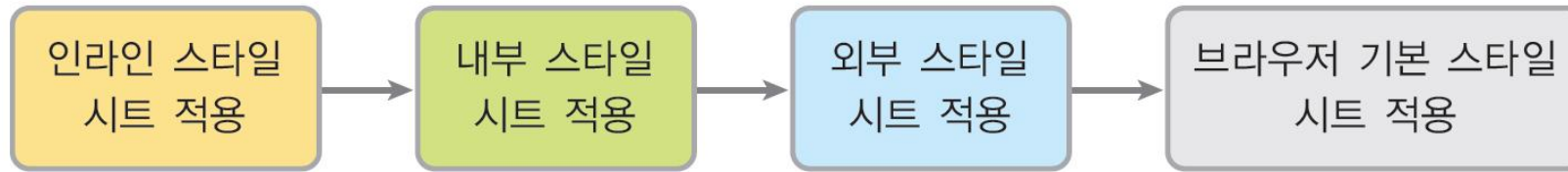
1. css 폴더 만들기
2. Google 에서 reset.css 검색 ( 초기화 )
3. CSS Tools: Reset CSS - Meyerweb 클릭
4. 내용 복사
5. css 폴더에 reset.css 만들기
6. 내용넣기
7. html문서에 head문안에 title문 아래
8. `<link rel="stylesheet" href="css/reset.css">`

요소는 각각의 기본적인 특성을 가지고 있기 때문에 요소의 특성을 초기화 시킨다

나중에는 normalize.css 사용함 ( reset.css와 비슷한듯하지만 다름 )



# Css 우선순위



CSS 적용 우선순위

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

# Css 우선순위

```
<head>
  <!-- 외부 스타일 시트를 정의하는 위치가 중요 -->
  <link rel="stylesheet" href="css/style.css">
  <style>
    p { color: blue; }
    p { color: yellow; }
    p { color: red; }
  </style>
</head>
<body>
  <p>CSS가 중복 정의된 경우 어떤 것이 적용될까?</p>
</body>
```

CSS가 중복 정의된 경우 어떤 것이 적용될까?

```
<head>
  <style>
    p { color: blue!important; }
  </style>
  <!-- 외부 스타일 시트를 정의하는 위치가 중요(최종 적용) -->
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <p>CSS가 중복 정의된 경우 어떤 것이 적용될까?</p>
</body>
```

CSS가 중복 정의된 경우 어떤 것이 적용될까?

# 기본선택자

#아이디 - 100점, .클래스 10 점, 태그 1점

## CSS의 기본 선택자

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.

# 기본선택자

#아이디 - 100점, .클래스 10 점, 태그 1점

## CSS

```
h1 {  
  font-size: 20px;  
}  
...  
h1 {  
  font-size: 18px;  
}
```

상세도가 같다(1점)



최종적으로 적용되는 스타일

```
font-size: 18px;
```

예를 들어, 다음과 같은 HTML의  
<li>에 CSS를 적용하고 싶을 때

```
<nav>  
  <ul>  
    <li>홈</li>  
    <li>다운로드</li>  
    <li>자주 하는 질문</li>  
  </ul>  
</nav>
```

자손 선택자를 사용하는 경우는  
가능한 한 선택자 수가 적어지도록 작성한다

✗ nav ul li {  
○ nav li {

# 기본선택자

#아이디 - 100점, .클래스 10 점, 태그 1점

## HTML

```
<p class="notice">주의</p>
```

저(1점)

상세도

고(10점)

## CSS

```
p {  
  margin: 0;  
  color: #000000;  
}  
  
.notice {  
  color: #ff0000;  
}
```



최종적으로 적용되는 스타일

```
margin: 0;  
color: #ff0000;
```

# 전체 선택자와 태그 선택자

- 선택자, 셀렉터(selector)
  - 스타일을 적용하기 위해 특정한 부분을 선택하는 방법
- 전체 선택자
  - 페이지에 있는 모든 요소를 대상으로 스타일을 적용할 때 사용
  - 다른 선택자와 함께 모든 하위 요소에 한꺼번에 스타일을 적용하려고 할 때 주로 사용
  - 예) `* { margin:0; padding:0; }`
- 태그 선택자
  - 문서 안의 특정 태그에 스타일이 모두 적용됨
  - 예) `p { font-size:12px; font-family:"돋움"; }`

# 클래스 선택자와 아이디 선택자

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

예)

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

```
#pic2 { clear:both; float:left; }
```

# 클래스 선택자 형식

```
<style>
p{
  font-size:12px; /* 글자 크기 12px */
  margin-left:20px; /* 왼쪽 여백 20px */
}

.accent {
  color:blue; /* 글자색 파랑(blue) */
  font-weight:bold; /* 글자 굵게(bold) */
}
</style>
</head>

<body>
<section>
  <h2>요안도라 소개</h2>
  <h3>α 요안도라는 게스트 하우스(Guest House) 형식의 농어촌 민박입니다</h3>
  <p class="accent">성산의 날씨는 다음주 내내 높은 구름에 햇살 가득이라고
  합니다.</p>
  <p>목요일이면 섭씨 27도까지 오른다고 하지만, </p>
  <p>늘 부는 바람이 쾌적한 균형을 잡아 마당에 마당에 나가 앉아 있는 시간이 많아질
  듯합니다. </p>
  <p>오늘은 사진에 보이는 긴 돌담을 따라 들어오는 <span class="accent">
  요안도레 올레 입구</span>에 특곤색의 대문을 달았습니다.</p>
</section>
</body>
```

① <p> 태그 선택자 지정, <p> 태그를 사용한 부분에 스타일 적용

② .accent 클래스 선택자 스타일 지정, class="accent"라고 사용한 부분에만 스타일 적용

<p> 태그 단락에 accent 클래스 선택자 스타일 적용

<span> 태그를 사용한 '요안도레 올레 입구'에 .accent 클래스 선택자 스타일 적용

AWS 기반 풀스택 웹개발(react.js, node.js)\_D : 14월 14일





# 클래스 선택자

```
<head>
  <style>
    .class1 {
      color: blue;
      background-color: yellow;
    }
    .class2 {
      color: red;
      background-color: green;
    }
    h3.class1 { /*h3태그의 클래스 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>
```

Class 1 입니다.

Class 1 입니다.

Class 2 입니다.

Class 2 입니다.

Element+Class Selector

# 아이디 선택자 - 유일성

<style>

```
#title{
  clear:both; /* 이전 요소와 줄바뀌서 표시 */
  float:left; /* 왼쪽으로 플로팅 */
  margin-left:10px; /* 왼쪽 여백 10px */
  width:200px; /* 너비 200px */
  height:110px; /* 높이 110px */
  background-color:orange; /* 배경색 오렌지색 */
  border:1px solid black; /* 테두리 1px짜리 검은색 실선 */
  text-align:center; /* 텍스트 가운데 정렬 */
}
```

title id 선택자  
스타일 지정

```
#pic1{
  clear:none; /* 이전 요소와 연결해서 표시 */
  float:left; /* 왼쪽으로 플로팅 */
  margin-left:10px; /* 왼쪽 여백 10px */
  width:200px; /* 너비 200px */
  height:110px; /* 높이 110px */
}
```

pic1 id 선택자  
스타일 지정

```
#pic2{
  clear:both; /* 이전 요소와 줄바뀌서 표시 */
  float:left; /* 왼쪽으로 플로팅 */
  margin-left:10px; /* 왼쪽 여백 10px */
  margin-top:20px; /* 위쪽 여백 20px */
  width:200px; /* 너비 200px */
  height:120px; /* 높이 120px */
}
```

pic2 id 선택자  
스타일 지정

```
#pic3{
  clear:none; /* 이전 요소와 연결해서 표시 */
  float:left; /* 왼쪽으로 플로팅 */
  margin-left:10px; /* 왼쪽 여백 10px */
  margin-top:20px; /* 위쪽 여백 20px */
  width:200px; /* 너비 200px */
  height:120px; /* 높이 120px */
}
```

pic3 id 선택자  
스타일 지정

# 아이디 선택자

아이디는 #아이디명 #아이디명 #아이디명 경로를 쓰지않는다

유일성이기 때문에

해당 #아이디부터 경로사용

예) #header #nav{}

#nav => 좋은예

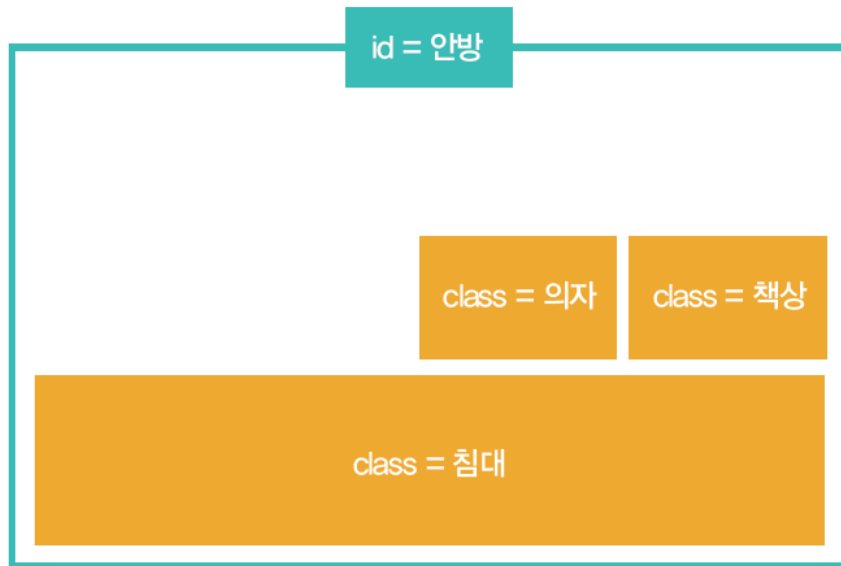
```
<head>
  <style>
    #id1 {color: blue; background-color: pink;}
    #id2 {color: blue; background-color: yellow;}
    #id3 {color: red; background-color: green}
  </style>
</head>
<body>
  <h1 id="id1">ID1 선택자</h1>
  <p id="id2">ID2 선택자</p>
  <h1 id="id3">ID3 선택자</h1>
  <p id="id4">ID4 선택자</p>
  <h2 id="id5">Element+ID Selector</h2>
</body>
```

# 아이디 선택자

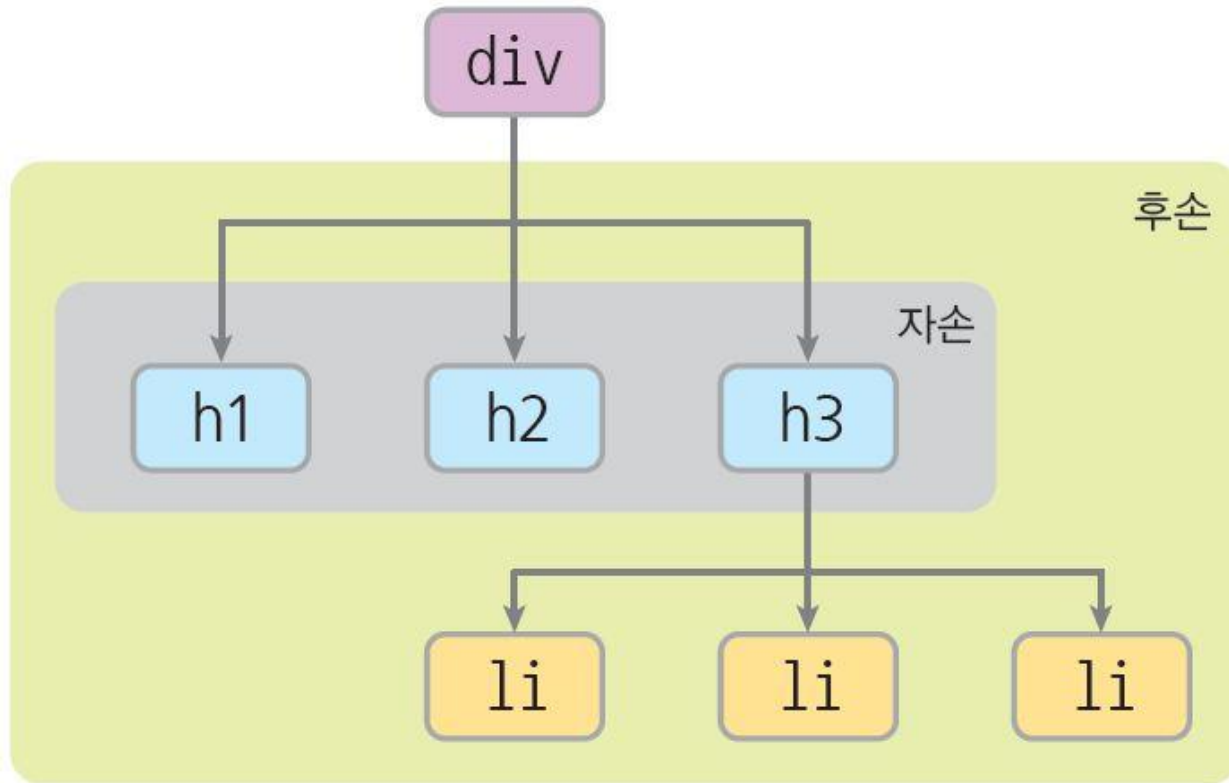
```
<head>
  <style>
    h1#id1 {  h1태그의 아이디 id1
      color: blue;
      background-color: pink;
    }
    #id1 {
      color: blue;
      background-color: yellow;
    }
    #id3 {
      color: red;
      background-color: green;
    }
  </style>
</head>
<body>
  <h1 id="id1">ID1 선택자</h1>
  <p id="id2">ID2 선택자</p>
  <h1 id="id3">ID3 선택자</h1>
  <p id="id4">ID4 선택자</p>
  <h2 id="id5">Element+ID Selector</h2>
</body>
```

# 요소 구조 - 선택자 형식

- id 속성과 class 속성의 사용
  - id 속성은 큰 공간을 만들 때 사용
  - class 속성은 큰 공간 안에 있는 작은 구역을 만들 때 사용
  - class 속성은 id 속성 안에서 작은 구역을 만들 때 사용하고, 이름이 같은 class 속성값을 여러 번 사용할 수 있다



## 요소 구조 - 선택자 형식



HTML 문서 트리 구조

# 선택자 형식 - 하위 선택자(descendant selector)

- 부모 요소에 포함된 모든 하위 요소에 스타일이 적용된다
- 하위 선택자를 정의할 때는 상위 요소와 하위 요소를 나란히 쓴다.

상위 요소   하위 요소   {속성:속성 값; ...}

기본형

예) `section p { color:blue; }`

# 하위 선택자

```
<head>
  <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>
```

## Descendant Selector\_1

후손 선택자

- 자식의 자식 (후손1)
- 자식의 자식 (후손2)

## Descendant Selector\_2



# 하위 선택자

```
<style>
  #container ul {
    border:1px dotted black;
  }
</style>
</head>
<body>

  <section id="container">
    <header><h1>예약 방법 및 요금</h1></header>
    <p>요안도라에 예약하려면?</p>
    <ul>
      <li>예약 방법
        <ul>
          <li>직접 통화</li>
          <li>문자 남기기</li>
        </ul>
      </li>
      <li>요금
        <ul>
          <li>1인 : 40,000원</li>
          <li>2인 : 60,000원</li>
          <li>3인 : 80,000원</li>
          <li>4인 : 100,000원</li>
        </ul>
      </li>
    </ul>
  </section>
</body>
```

container 요소 아래의 모든 ul 요소에 1px 두께의 검은색 테두리 지정

모든 <ul> 태그에 스타일 적용

## 구현된 결과

### 예약 방법 및 요금

요안도라에 예약하려면?

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

# 자식 선택자 (child selector) , >

- 부모 요소의 자식 요소에만 스타일이 적용된다.
- 정말 써야할 경우만 작성한다 (유지보수때문에)

부모 요소 > 자식 요소 {속성:속성 값; ....}

예) `section > p { color:blue; }`

```
<style>
  #container > ul {
    border:1px dotted black;
  }
</style>
</head>

<body>
  <section id="container">
    <header><h1>예약 방법 및 요금</h1></header>
    <p>요안도라에 예약하려면?</p>
    <ul>
      <li>예약 방법
        <ul>
          <li>직접 통화</li>
          <li>문자 남기기</li>
        </ul>
      </li>
      <li>요금</li>
      <ul>
        <li>1인 : 40,000원</li>
        <li>2인 : 60,000원</li>
        <li>3인 : 80,000원</li>
        <li>4인 : 100,000원</li>
      </ul>
    </ul>
  </section>
</body>
```

id가 container인 요소 아래의 자식 ul 요소에 1px 두께의 테두리 지정

<section> 태그의 자식 <ul> 태그에 적용

id가 container인 <section> 태그

# 자식 선택자 (child selector) , >

Sno 등 태그는 가상입니다. 참고만..

```
<head>
  <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>
    <std>홍민성</std>
    <tel>
      <office>02-4567-1010</office>
      <home>010-1234-5678</home>
    </tel>
  </h3>
</body>
```

## Child Selector\_1

자식 선택자

123456

홍민성

02-4567-1010

010-1234-5678

# 인접 형제 선택자(adjacent selector)

- 문서 구조상 같은 부모를 가진 형제 요소 중 첫 번째 동생 요소에만 스타일 적용.
- 같은 부모 요소를 가지는 요소들을 형제 관계라고 부른다.  
먼저 나오는 요소를 '형 요소', 나중에 나오는 요소를 '동생 요소'라고 한다.  
해당 요소의 바로 아래 형제 요소를 찾아간다

요소1 + 요소2 {속성:속성 값; ...}

기본형

예) `h1 + p { text-decoration : underline; }`

# 인접 형제 선택자(adjacent selector)

```
<head>
  <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>
```

## 인접 형제 선택자1

### 인접 형제 선택자2

- 목록
- 주제1
- 주제2

### Adjacent Selector\_1

인접 형제 선택자에 의한 스타일 적용

### Adjacent Selector\_2

# 인접 형제 선택자(adjacent selector)

```
<style>
  h1+ul {
    color:blue; /* 글자색 파랑 */
    font-weight:bold; /* 글자 굵게 */
  }
</style>
</head>
<body>
  <section id="container">
    <h1>예약 방법 및 요금</h1>
    <ul>
      <li>직접 통화</li>
      <li>문자 남기기</li>
    </ul>
    <ul>
      <li>1인 : 40,000원</li>
      <li>2인 : 60,000원</li>
      <li>3인 : 80,000원</li>
      <li>4인 : 100,000원</li>
    </ul>
  </section>
</body>
```

h1 태그 다음의 첫 번째 ul 태그에 적용할 스타일 지정

h1 태그 다음에 오는 첫 번째 ul 태그

## 구현된 결과

### 예약 방법 및 요금

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

# 형제 선택자(sibling selector) ~

- 형제 선택자(sibling selector)
- 형제 요소들에 스타일 적용
- 인접 형제 선택자와 다른 점은 모든 형제 요소에 다 적용된다는 것이다.

요소1 ~ 요소2 {속성:속성 값; ...}

기본형

예) `h1~p { text-decoration : underline; }`

# 형제 선택자(sibling selector)

```
<style>
h1~ul {
  color:blue; /* 글자색 파랑 */
  font-weight:bold; /* 글자 굵게 */
}
</style>
</head>

<body>
  <section id="container">
    <h1>예약 방법 및 요금</h1>
    <ul>
      <li>직접 통화</li>
      <li>문자 남기기</li>
    </ul>
    <ul>
      <li>1인 : 40,000원</li>
      <li>2인 : 60,000원</li>
      <li>3인 : 80,000원</li>
      <li>4인 : 100,000원</li>
    </ul>
  </section>
</body>
```

〈h1〉 태그 다음에 오는 모든 형제 요소  
〈ul〉 태그에 적용할 스타일 지정

〈h1〉 태그 다음에 오는 〈ul〉 태그

## 구현된 결과

### 예약 방법 및 요금

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



# 형제 선택자(sibling selector)

```
<head>
  <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>
```

## 형제 선택자1

## 형제 선택자2

- 목록
- 주제1
  - 주제2

## Sibling Selector-1

같은 레벨 형제

일반 형제 선택자에 의한 스타일 적용

## Sibling Selector\_2

## Sibling Selector\_3

# 그룹 선택자(group selector)

- 하위 선택자(descendant selector)
- 같은 속성을 적용해야 할 경우 똑같은 스타일을 두번 정의하지 않고 한번에 묶어서 정의한다
- 심표로 선택자 구분

```
요소1, 요소2, ... {속성:속성 값; ...}
```

기본형

- 예) `h1, h2, h3, li { color: #ffffff; }`

```
<section>
```

```
<h1>요안도라 소개</h1>
```

```
<h3>요안도라는 게스트 하우스(Guest House) 형식의 농어촌 민박입니다</h3>
```

```
<style>
```

```
h1, h3{  
  font-family: "돋움"; /* 돋움 글꼴 */  
  color:purple; /* 글자색 보라 */  
}
```

{h1}, {h3} 태그 선택자 스타일 지정

## 구현된 결과

### 요안도라 소개

요안도라는 게스트 하우스(Guest House) 형식의 농어촌 민박입니다

성산의 날씨는 다들 주 내내 좋은 구름에 햇살 가득이라고 합니다. 꼭 필요하면 새벽 27도까지 오른다고 하지만, 날씨는 바람에 따라 급변을 겪어 대낮에 대낮에 나가 있어 있는 시간이 많아질듯 합니다. 오늘은 시간에 보이는 건 물담을 따라 돌아오는 요안도라 출제 입구에 특근색의 대문을 달았습니다.

# 그룹 선택자(group selector)

```
h1 { font-family: sans-serif; }  
h2 { font-family: sans-serif; }  
h3 { font-family: sans-serif; }
```

그룹 선택자로 지정하기 전



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

그룹 선택자로 지정한 후

```
<head>  
  <style>  
    h3, p, h2 { color: red; background-color: yellow; }  
  </style>  
</head>  
<body>  
  <h2>Group Selector_1</h2>  
  <p>스타일 지정은 그룹으로 적용</p>  
  <div>  
    <h3>Group Selector_2</h3>  
  </div>  
</body>
```

Group Selector\_1

스타일 지정은 그룹으로 적용

Group Selector\_2