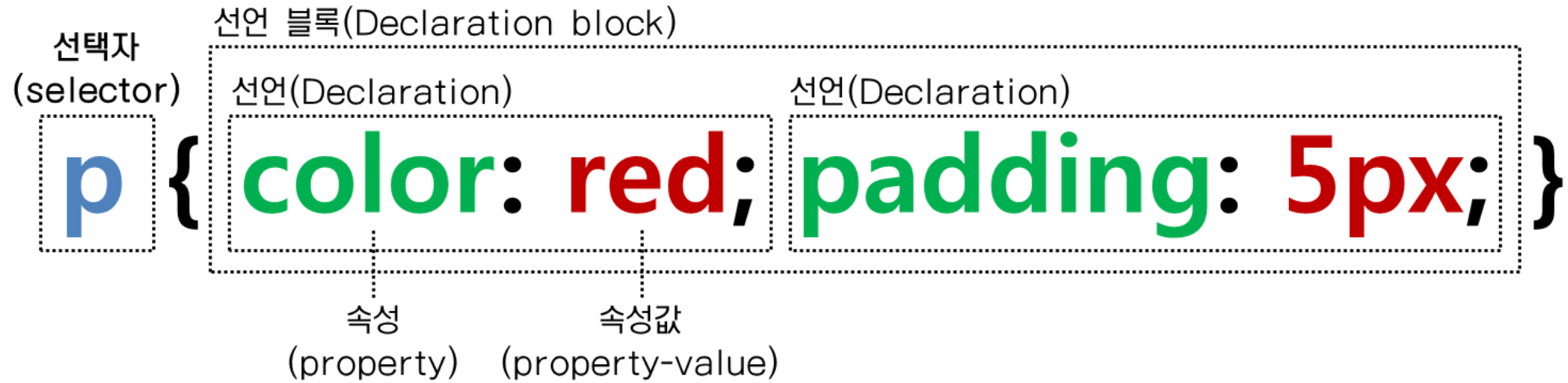


CSS

Part. 2

CSS 선택자

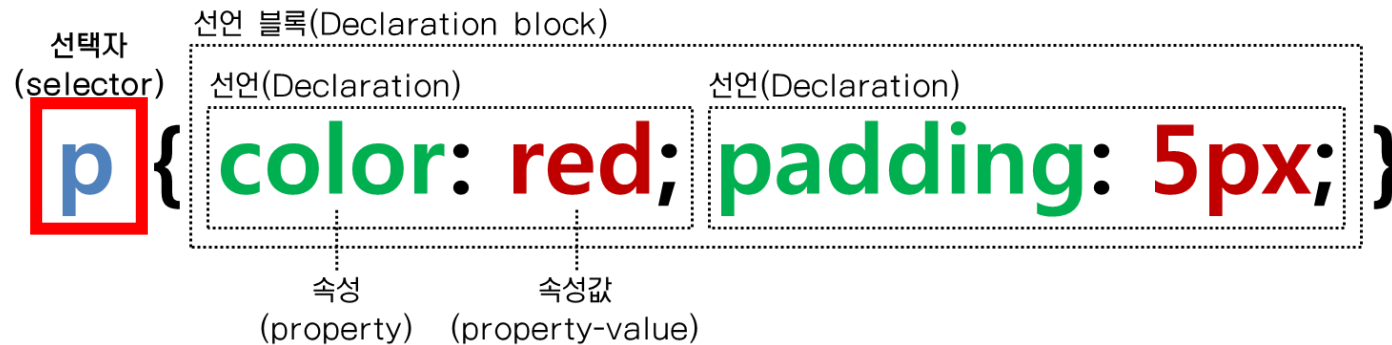
CSS 작성법



선택자 { 프로퍼티: 값; }

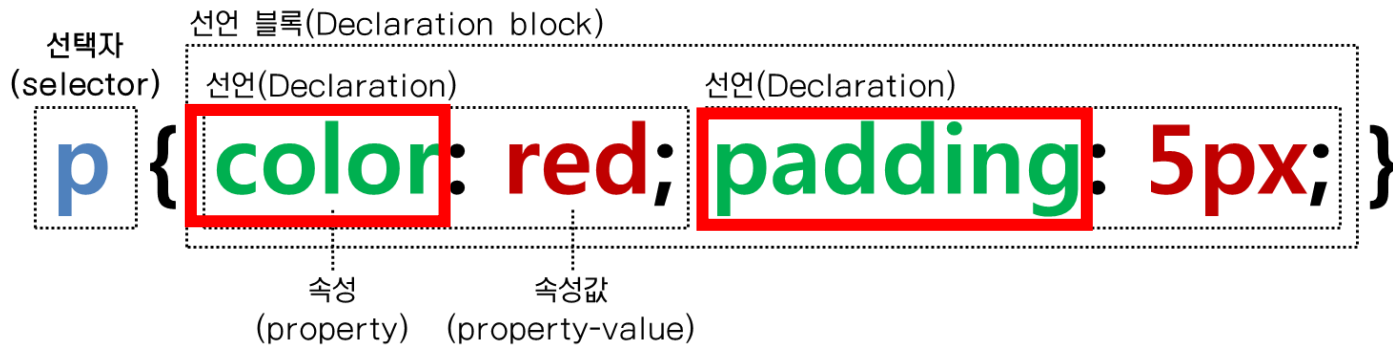
= 선택자의 프로퍼티를 값으로 변경하겠음!

CSS, 선택자(Selector)



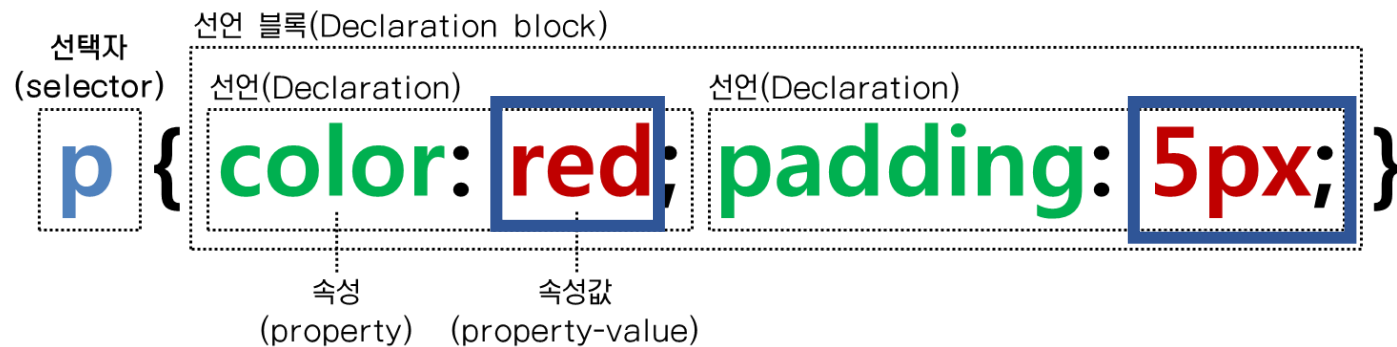
- CSS는 HTML 요소의 style을 정의하여야 하므로 스타일을 적용하고자 하는 HTML 요소를 선택할 수 있어야 한다
- 선택자는 스타일을 적용하고자 하는 HTML 요소를 선택하기 위해 CSS에서 제공하는 수단

CSS, 속성(Property)



- 선택자로 HTML 요소를 선택하고 { } 내에 속성 값을 지정하여 다양한 style을 정의
- 여러 개의 프로퍼티를 연속해서 지정할 수 있으며 세미콜론(;)으로 구분
- 속성은 표준 스펙으로 이미 지정되어 있는 것을 사용, 사용자가 임의로 정의할 수 없다

CSS, 값(Value)



- 값은 해당 속성에 사용할 수 있는 값을 키워드나 크기 단위 또는 색상 단위 등의 특정 단위로 지정

CSS 선택자

기본

복합

가상 클래스

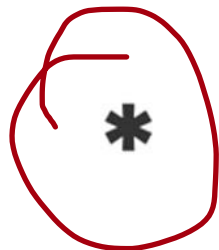
가상 요소

속성



기본 선택자!

- 기본 선택자니까 기본적인 선택자겠죠?
- 별도의 테크닉 없이, 순수하게 무엇인가를 호출 할 때 사용합니다!
- 종류
 - 전체 선택자
 - 태그 선택자
 - Class 선택자
 - ID 선택자

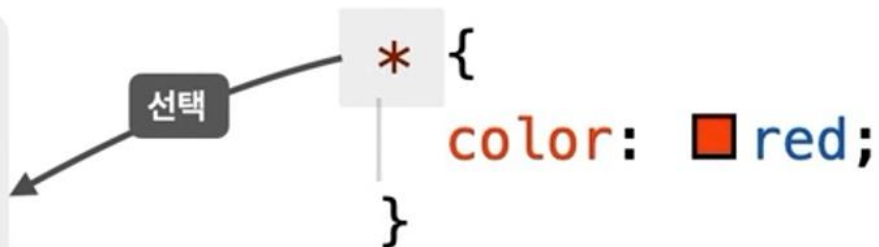


기본

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>
```



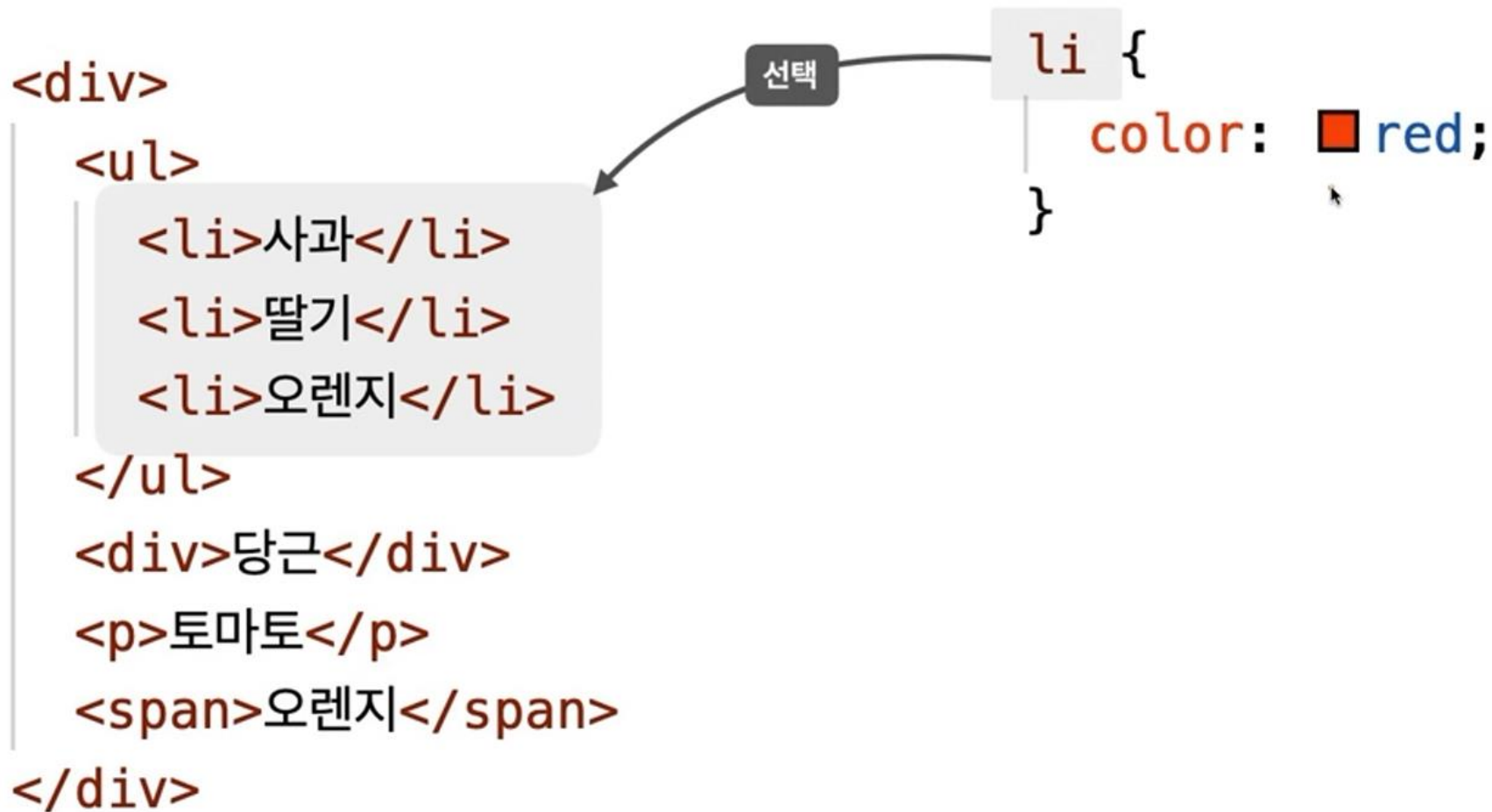
별 표는 모든 all 을 의미한다

ABC

기본

태그 선택자 (Type Selector)

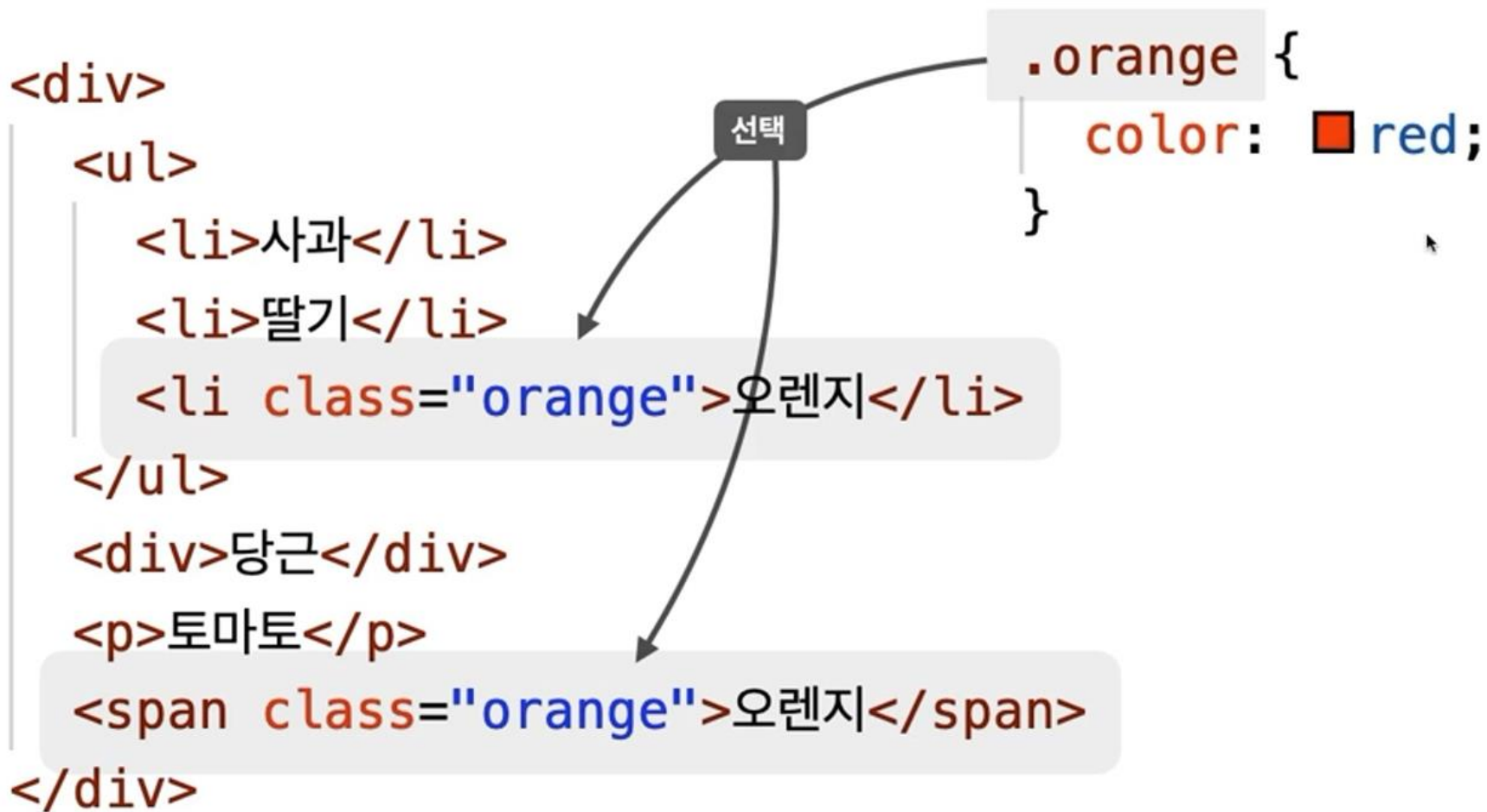
태그 이름이 ABC인 요소 선택.



.ABC

기본

클래스 선택자 (Class Selector)

HTML class 속성의 값이 ABC인 요소 선택.

#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;
}
```



id: 는 전체적으로 하나만 있어야 한다

실습1 기본 선택자 연습하기

```
<h1>기본 선택자 연습해보기</h1>
<p>기본선택자 연습을 해보아요!</p>
<span>안녕하세요</span>
<ol>
  <li id="first">1</li>
  <li class="second">2</li>
  <li>3</li>
  <li>4</li>
  <li>5</li>
</ol>
```

- 모든 글씨의 색은 white
- 브라우저의 배경색은 gray
- span 태그의 배경색 red
- first라는 아이디를 가진 태그엔 배경색 blue
- second라는 클래스를 가진 태그엔 배경색 green

CSS 선택자

기본

복합

가상 클래스

가상 요소

속성



복합 선택자!

- 특수한 요소를 호출하고 싶을 때, 기본 선택자만으로는 선택이 불가능한 경우에
사용
- 종류
 - 일치 선택자
 - 자식 선택자
 - 후손 선택자
 - 인접 형제 선택자
 - 일반 형제 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

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

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;
}
```

ABC + XYZ

복합

인접 형제 선택자 (Adjacent Sibling Combinator)

선택자 ABC의 다음 형제 요소 XYZ 하나를 선택.

선택

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

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

ABC ~ XYZ

복합

일반 형제 선택자 (General Sibling Combinator)

선택자 ABC의 다음 형제 요소 XYZ 모두를 선택.

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

선택

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

실습2 조건!

- 야생의 사자만 선택, 배경을 빨간색
- 사파리 안의 사자만 선택- 배경을 greenyellow
- 호랑이만 선택, 배경을 노란색
- 사자를 전부 선택, 글자를 sienna 색
- 동물원 안의 사자만 선택, 폰트를 두껍게
- 호랑이 뒤에 있는 팬더를 선택, 배경을 하늘색(skyblue)
- 사파리 사자 뒤에 있는 사육사를 모두 선택, 배경을 오렌지색

실습2 코드

```
<body>
  <h1>복합선택자 사용 연습</h1>
  <p>동물원에 왔어요</p>
  <div class="zoo">
    <ul>
      <span>여긴 사파리!</span>
      <li>곰</li>
      <li id="tiger">호랑이</li>
      <li>팬더</li>
      <span class="lion">사자1</span>
      <li class="lion">사자2</li>
      <li>사육사 sarah</li>
      <li>사육사 세령</li>
    </ul>
    <span class="lion">사파리 밖의 사자</span>
  </div>
  <p class="lion">야생의 사자</p>
</body>
```

CSS 선택자

기본

복합

가상 클래스

가상 요소

속성



가상 클래스 선택자!

- 사용자의 행동에 따라 변화하는 가상 상황에 따라서 요소 선택 시
- 각 요소의 상황에 따라 사용자가 원하는 요소를 선택 할 때 사용
- 특정 요소를 부정 할 때 사용
- 종류
 - 가상 클래스 선택자
 - 사용자의 행동에 따라 변화 : Hover, Active, Focus, ...
 - 요소의 상황 : first-child, last-child, nth-child, ...
 - 부정 선택 : not

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>
```

ABC:active

가상 클래스 선택자 (Pseudo-Classes)

ACTIVE

선택자 ABC 요소에 마우스를 클릭하고 있는 동안 선택.

화면에 출력!

NAVER

NAVER

선택

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

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

ABC:focus

가상 클래스 선택자 (Pseudo-Classes)

FOCUS

선택자 ABC 요소가 포커스되면 선택.

화면에 출력!

선택

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

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

실습 3. 가상 클래스 선택자 1

- `<h1>`, `<a>`, `<input type="text">` 태그 만들기
- `<h1>` 태그에 마우스가 올라가면 글자가 빨간색
- `<a>` 태그를 클릭 하고 있으면 배경이 파란색
- `<input>` 태그에 포커스가 가면 입력창이 오렌지색

ABC:first-child

가상 클래스 선택자 (Pseudo-Classes)

FIRST CHILD

선택자 ABC가 형제 요소 중 첫째라면 선택.

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

선택

```
.fruits span:first-child {
  color: red;
}
```

?

```
.fruits div:first-child {
  color: red;
}
```

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;
}
```

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;
}
```


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)

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

n은 0부터 시작!
(Zero-Based Numbering)

ABC:not(XYZ)

부정 선택자 (Negation)

NOT

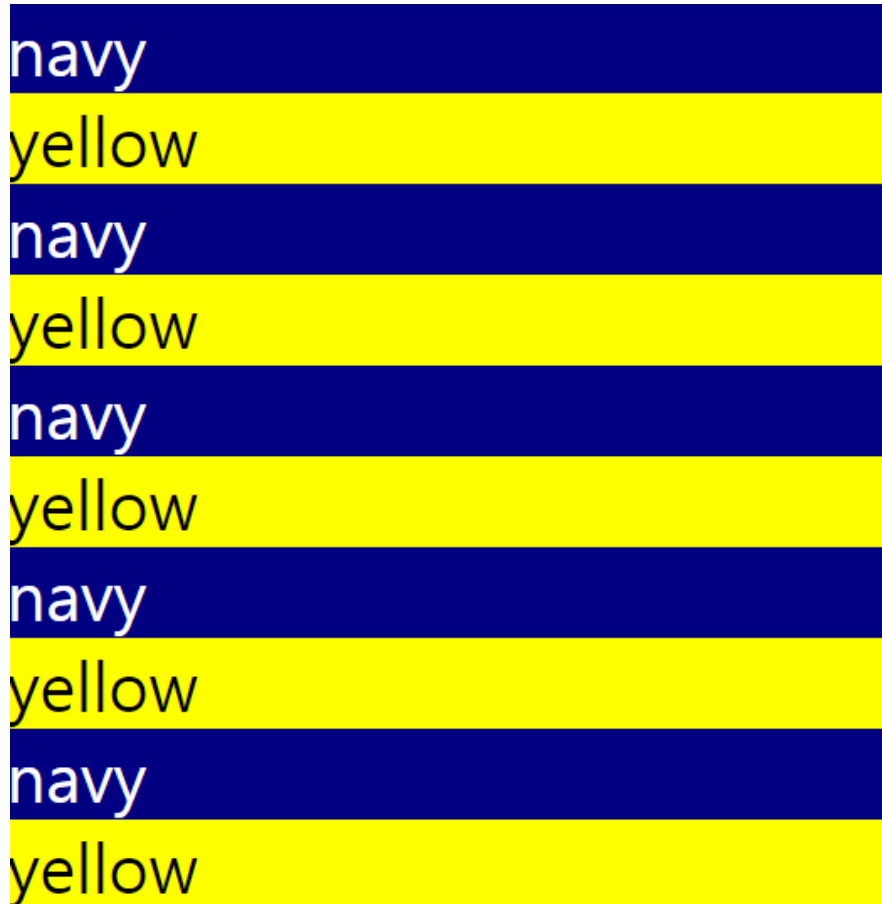
선택자 XYZ가 아닌 ABC 요소 선택.

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

선택

```
.fruits *:not(span) {  
  color: red;  
}
```

실습4. 줄무늬 모양 만들기



```
<div class="stripes">  
  <div>navy</div>  
  <div>yellow</div>  
  <div>navy</div>  
  <div>yellow</div>  
  <div>navy</div>  
  <div>yellow</div>  
  <div>navy</div>  
  <div>yellow</div>  
  <div>navy</div>  
  <div>yellow</div>  
</div>
```

CSS 선택자

기본

복합

가상 클래스

가상 요소

속성



가상 요소 선택자!

- 선택 된 요소의 앞, 뒤에 별도의 Content 를 삽입하는 선택자
- 반드시 content 라는 속성을 사용
- 빈 값("") 이라도 넣어 주어야 적용이 됨
- 종류
 - After : 요소의 뒤에 내용 삽입
 - Before : 요소의 앞에 내용 삽입

ABC::before

인라인(글자) 요소

화면에 출력!

앞! Content!

가상 요소 선택자 (Pseudo-Elements)

BEFORE

선택자 ABC 요소의 내부 앞에 내용(Content)을 삽입.

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

::before

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

Content!

```
</div>
```

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

가상 요소 선택자!

- 실제로 의미 없는 HTML 태그를 만들지 않고 요소 삽입이 가능하여 자주 사용!
- 예를 들어 쇼핑몰 페이지에 메뉴에 Hot, 추천 등을 넣기 위해 별도의 태그를 삽입 하는 것이 아니라 가상 요소 선택자를 활용하여 처리하면 편리함!

실습 5. 가상 요소 선택자

- `<div class="box">여기요!</div>` 만들기
- 여기요 앞에, "택시" 넣기
- 여기요 뒤에, "빨리" 넣기

CSS 선택자

기본

복합

가상 클래스

가상 요소

속성



속성 선택자!

- 지정한 **특정 속성**을 가지고 있는 태그를 선택하는 선택자
- 종류
 - 특정 속성만 지정
 - 속성과 속성의 값을 지정

[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>
```

[ABC="XYZ"]

속성 선택자 (Attribute)

ATTR=VALUE

속성 ABC를 포함하고 값이 XYZ인 요소 선택.

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

선택

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

실습 6. 속성 선택자

```
<input type="text" placeholder="이름" />  
<input type="password" value="pw" />  
<input type="text" value="000-0000-0000" />  
<input type="text" placeholder="핸드폰" />  
<input type="text" placeholder="주민번호" disabled />
```

- Disabled 속성을 가진 태그 선택 → 배경 빨간색
- placeholder 속성이 "이름" 값을 가지는 태그 선택 → 배경 오렌지
- placeholder 속성을 가지지 않는 태그 선택 → 배경 파란색