

삼성 청년 SW 아카데미

HTML & CSS

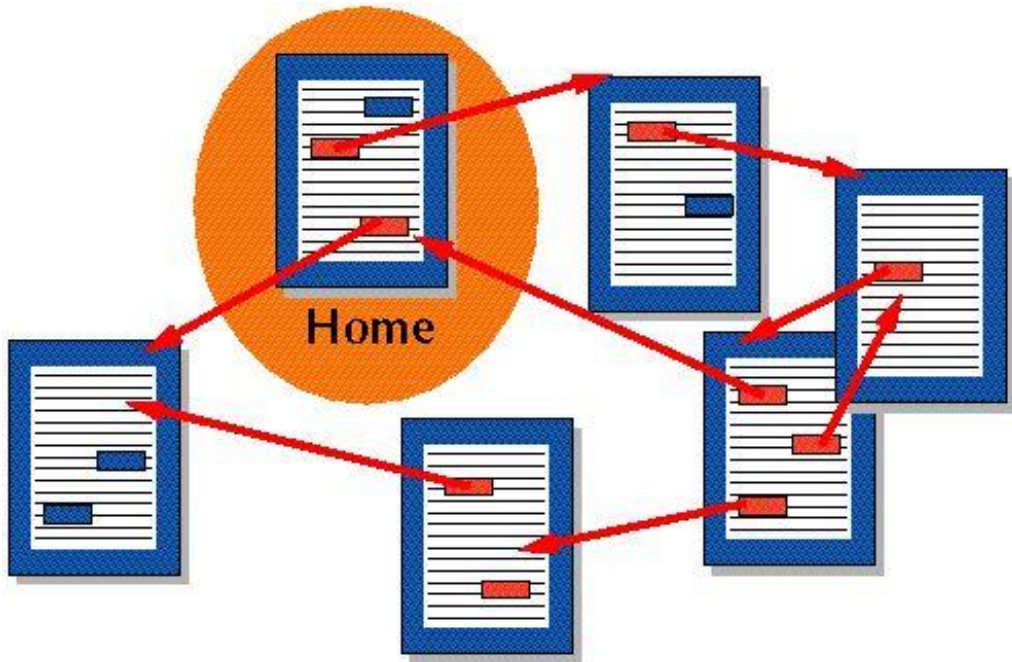
HTML & CSS

- HTML
- CSS
- CSS Selector

HTML

✓ HTML

- Hyper Text Markup Language
- Hyper Text : 참조를 통해 한 문서에서 다른 문서로 즉시 접근할 수 있는 텍스트
- Markup : 태그 등을 이용하여 문서나 데이터의 구조를 명기하는 언어



```
<html>
<head></head>
<body>
  <header>
    <title>http://info.cern.ch</title>
  </header>

  <h1>http://info.cern.ch - home of the first website</h1>
  <p>From here you can:</p>
  <ul>
    <li>
      <a href="http://info.cern.ch/hypertext/WWW/TheProject.html">
        Browse the first website
      </a>
    </li>
    <li>
      <a href="http://line-mode.cern.ch/www/hypertext/WWW/TheProject.html">
        Browse the first website using the line-mode browser simulator
      </a>
    </li>
    <li>
      <a href="http://home.web.cern.ch/topics/birth-web">
        Learn about the birth of the web
      </a>
    </li>
    <li>
      <a href="http://home.web.cern.ch/about">
        Learn about CERN, the physics laboratory where the web was born
      </a>
    </li>
  </ul>
</body>
</html>
```

✓ HTML

- 웹페이지를 작성하기 위한 언어
- .html (확장자)를 가짐.
- 태그는 대소문자 구분이 없음.
- 엔터, 스페이스바, 탭이 적용되지 않음.

SAMSUNG
SW
ACADEMY
FOR
YOUTH

About SSAFY

SSAFY 소개
교육 프로그램
자문위원단
교육생 지원 내용

SSAFY Life

SSAFY 소식
SSAFY 후기
캠퍼스 소개

Apply

지원하기
알림받기

[본문 바로가기](#)
[뒤로가기](#)

SSAFY

- [About SSAFY](#)
- [SSAFY Life](#)
- [Apply](#)

[FAQ](#) [1:1문의](#)

- [로그인](#)
- [회원가입](#)

Open the Menu

[home](#)

로그인해주세요.

- [로그인](#)
- [회원가입](#)
- [공지사항](#)
- [1:1문의](#)
- [FAQ](#)

- [About SSAFY](#)
 - [SSAFY 소개](#)
 - [교육 프로그램](#)
 - [자문위원단](#)
 - [교육생 지원 내용](#)
 - [취업지원](#)

[로그인](#) | [회원가입](#)

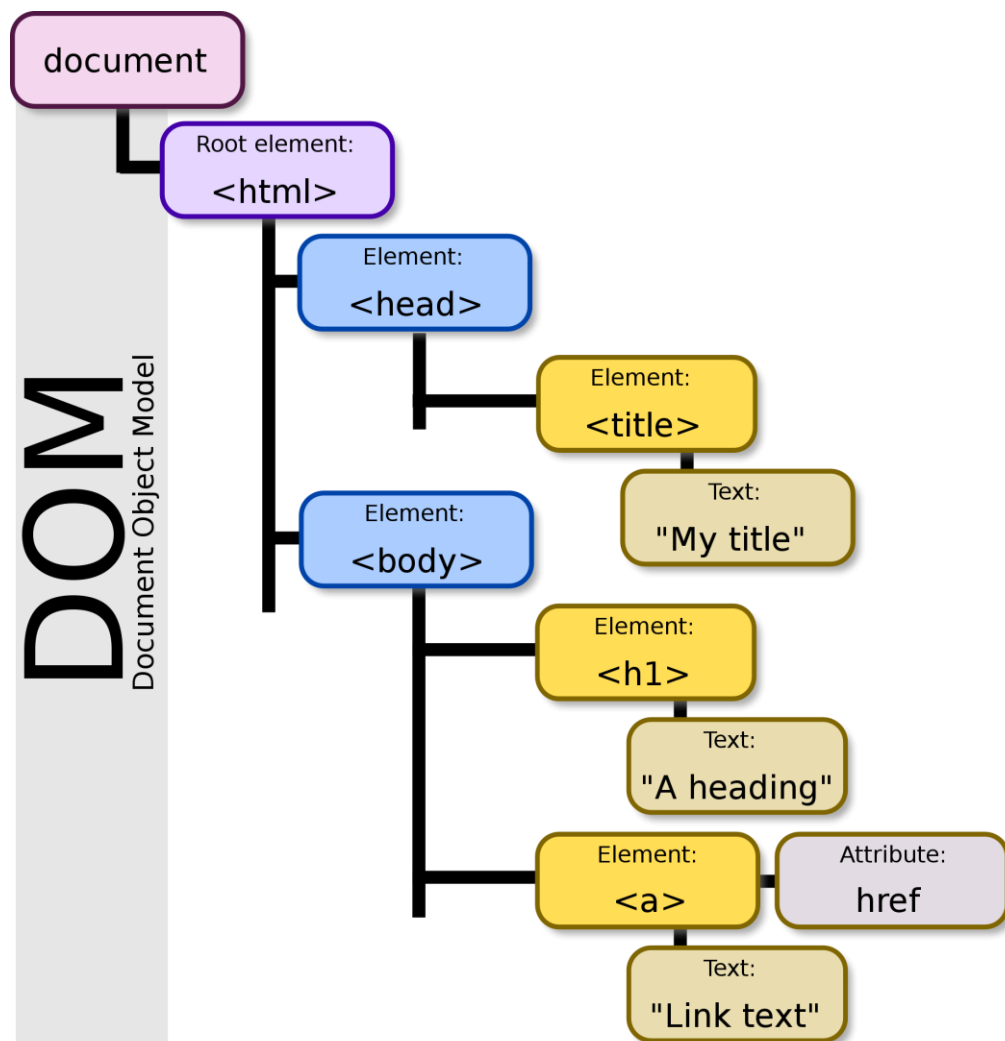
n

삼성의 SW 교육 경험과 고용노동부의 취업 지원 노하우를 바탕으로,
취업 준비생에게 SW 역량 향상 교육 및 다양한 취업지원 서비스를 제공하여
취업에 성공하도록 돕는 프로그램입니다.

본 교육과정은 삼성이 주관하고 고용노동부가 후원하며, 아이들과 미래세대가 함께합니다.



✓ DOM(Document Object Model) 트리

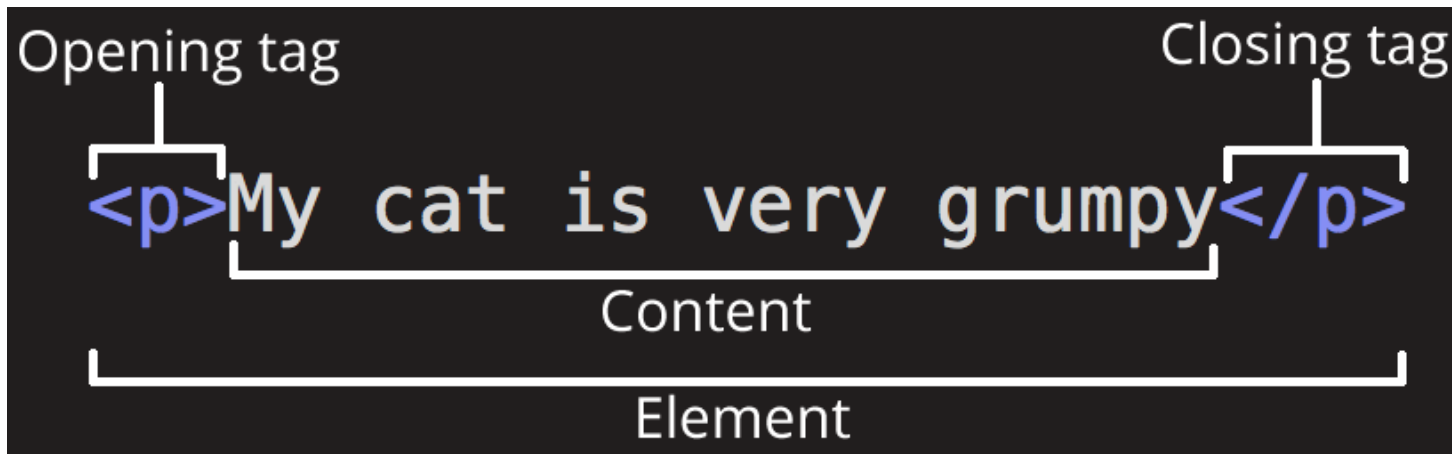


✓ 태그 (Tag)

- HTML의 요소는 태그와 내용으로 구성

`에듀싸피`

- 시작태그 / 종료태그 로 쌍을 이루거나 시작 tag만 존재하는 tag도 있다.
- 각각의 시작태그는 속성과 속성값을 가질 수 있음.



✓ 주석

- 주석의 내용은 브라우저에 출력이 되지 않음.
- HTML tag의 내용을 설명하기 위한 용도로 사용.
- `<!-- 주석 내용 -->`

The screenshot illustrates the behavior of HTML comments in a web browser. On the left, a code editor shows the following HTML code:

```
1 <h1>이건 보인다.</h1>
2 <!-- 이걸 안보인다.-->
3 <p>이건 보인다.</p>
4
```

The browser's viewport displays the rendered output. The text "이건 보인다." is shown in a large font, corresponding to the `<h1>` tag. The comment `<!-- 이걸 안보인다.-->` is not visible, demonstrating that comments are ignored by the browser. Below the main text, the text "이건 보인다." is shown in a smaller font, corresponding to the `<p>` tag.

On the right, the browser's developer tools (Elements panel) show the DOM tree. The structure is as follows:

- `<html>`
 - `<head></head>`
 - `<body>` (expanded)
 - `<h1>이건 보인다.</h1>`
 - `<!-- 이걸 안보인다.-->` (comment, not rendered)
 - `<p>이건 보인다.</p>`

The comment is present in the DOM tree but is not rendered in the browser's output, confirming that comments are used for documentation and are not part of the visual content.

✓ 기본 구조

- html : HTML 최상위 요소, 문서의 root → head와 body 로 구성
- head : 문서제목 문자코드(인코딩) 등 해당 문서에 대한 정보를 가지고 있고, 브라우저 화면 출력 x
- meta : 문서의 작성자, 날짜 등 본문에 나타나지 않는 일반 정보들
- body : 브라우저 화면에 나타나는 정보
id 속성을 이용하여 문서 내에서 tag 식별 가능 (중복 x)
class 속성을 이용하여 여러 tag에 공통적인 특성 부여 (중복 o)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

✓ 특수문자

엔티티 이름	설명	화면출력
 	Non-breaking space(공백)	
<	Less than	<
>	Greater than	>
&	Ampersand	&
"	Quotation mark	"
©	Copyright	©
®	registered trademark	®

✓ Semantic tag

- 의미론적 요소를 담은 태그의 등장

태그 이름	설명
<header>	문서 전체나 섹션이 헤더
<nav>	네비게이션
<aside>	사이드에 위치한 공간
<section>	문서의 일반적인 구분
<article>	문서, 페이지, 사이트 안에서 독립적으로 구분되는 영역
<footer>	문서 전체나 섹션이 푸터
<h1> ~ <h6>	6단계 구획 제목

✓ Semantic tag

- 코드의 가독성을 높이고 유지보수를 쉽게 함

```
<div>
  <div></div>
</div>
<div>
  <div>
    <div></div>
    <div></div>
  </div>
  <div></div>
</div>
<div></div>
```



```
<header>
  <nav></nav>
</header>
<main>
  <section>
    <article></article>
    <article></article>
  </section>
  <aside></aside>
</main>
<footer></footer>
```

✓ Text content

태그 이름	설명
<blockquote>	긴 인용문, 주로 들여쓰기를 한 것으로 그려짐
<hr>	구분선
<pre>	공백, 줄바꿈 등 입력된 그대로 화면에 표시
<p>	하나의 문단
	정렬되지 않은 목록 (번호 없는)
	정렬된 목록 (번호 있는)
<div>	구문 콘텐츠를 위한 블록 컨테이너

✓ Inline text semantics

태그 이름	설명
<a>	href 특성을 통해 다른 페이지나 같은 페이지의 어느 위치, 파일, 이메일 주소와 그 외 다른 URL로 연결할 수 있는 하이퍼링크 생성
, 	굵은 글씨, 중대하거나 긴급한 콘텐츠 (strong)
 	텍스트 안에 줄바꿈을 생성
<i>, 	기울게 표시, 특정 문자열을 강조(em)
<q>	짧은 인라인 인용문
<s>	취소선
<u>	밑줄
<sup>, <sub>	위 첨자, 아래 첨자
	구문 콘텐츠를 위한 인라인 컨테이너

✓ Image & Multimedia

태그 이름		설명
<audio>	소리 콘텐츠를 포함할 때 사용	
	문서에 이미지 삽입	
<video>	미디어 플레이어를 문서에 삽입	

✓ Table content

태그 이름	설명
<table>	행과 열로 이루어진 표를 나타낸다.
<thead>	테이블의 열의 머리글인 행들의 집합
<tbody>	표의 여러 행(tr)을 묶어서 표 본문 구성
<tfoot>	테이블의 열을 요약하는 행들의 집합
<tr>	테이블 행
<th>, <td>	머리글, 데이터
<col>	표의 열을 나타냄
<colgroup>	표의 열을 묶는 그룹 정의
<caption>	표의 설명 또는 제목을 나타냄

✓ Table content

지역	반	학생수
서울	5반	30명
	6반	30명
	7반	28명
	8반	28명
대전	3반	28명
	4반	28명
총합		150명

✓ Forms

- 사용자로부터 데이터를 입력 받아 서버에서 처리하기 위한 용도로 사용
- 사용자가 작성한 데이터를 서버로 전송 (submit)

속성	설명
method	GET : 주소 표시줄에 사용자가 입력한 내용이 표시. 256 ~ 2048bytes(길이 제한)의 데이터만 서버로 전송.
	POST : HTTP 메시지의 Body에 담아서 전송하기 때문에 전송 내용의 길이에 제한이 없다. 사용자가 입력한 내용이 표시되지 않는다.
name	form의 이름을 지정.
action	<form> 태그 안의 내용들을 처리해 줄 서버상의 프로그램 지정.(URL)
target	<action> 태그에서 지정한 스크립트 파일을 현재 창이 아닌 다른 위치에 열도록 지정.

✓ Forms

태그 이름	설명
<form>	정보를 제출하기 위한 대화형 컨트롤을 포함하는 문서 구획을 나타냄
<button>	클릭 가능한 버튼을 나타냄
<input>	웹 기반 양식에서 사용자의 데이터를 받을 수 있는 대화형 컨트롤
<label>	사용자 인터페이스 항목의 설명을 나타냄
<select>	옵션 메뉴를 제공하는 컨트롤을 나타냄
<option>	항목
<textarea>	멀티라인 일반 텍스트 편집 컨트롤을 나타냄
<fieldset>	웹 양식의 여러 컨트롤과 레이블을 묶을 때 사용함.
<legend>	fieldset 콘텐츠의 설명을 나타냄

✓ input

- 요소의 동작은 type 속성에 따라 달라짐.

속성	설명
autofocus	페이지 로딩 후 폼의 요소 중에서 해당 요소에 마우스 커서를 표시. html5이전에는 자바스크립트로 구현.
placeholder	텍스트를 입력할 때 도움이 되도록 입력란에 적당한 힌트 내용을 표시. 클릭 시 자동으로 내용이 사라짐.
readonly	입력란에 텍스트를 사용자가 직접 입력하지 못하게 읽기 전용으로 지정. readonly, readonly="readonly", readonly="true"로 표현.
required	form에 data를 입력한 후 submit 클릭 시 data를 서버로 전송하기전 필수 입력 항목을 체크. required, required="required"로 표현.
min, max, step	min, max는 해당 필드의 최대, 최소값지정. step은 일정 간격 지정. type이 date, datetime, datetime-local, month, week, time, number, range에서 사용.
size, minlength, maxlength	minlength, maxlength는 텍스트 입력 시 최대, 최소길이 지정. size는 화면에 보여지는 글자의 길이 지정.
height, width	type="image"일 때 이미지의 너비와 높이를 지정.
multiple	type="email"이나 type="file"일 때 두 개 이상의 값을 입력. <input> 태그 안에 속성 이름만 표시하면 됨.

CSS

✓ CSS

- Cascading Style Sheets
- HTML 문서를 화면에 표시하는 방식을 정의한 언어

삼성 청년 SW 아카데미(SSAFY)는
삼성의 SW 교육 경험과
고용노동부의 취업지원 노하우를 바탕으로 취업 준비생에게 SW 역량 향상 교육 및
다양한 취업지원 서비스를 제공하여 취업에 성공하도록 돕는 프로그램입니다.

- SSAFY는
최고 수준의 교육을 제공합니다.

전문분야별 자문교수단과 삼성의 SW 전문가가
함께 참여한 명품 커리큘럼을 제공하여 경쟁력 있는
차세대 SW 인력을 양성합니다.
- SSAFY는
맞춤형 교육을 제공합니다.

SW 전공자와 비전공자의 수준에 따라
맞춤형 교육을 제공하여
최적의 학습 효과를
지향합니다.
- SSAFY는
자기주도적 학습을 지향합니다.

단순히 지식을 전달하기 보다

스스로 문제를 해결할 수 있는 역량을
강화시키고,
기업에서 실제로 수행하는 형태의 프로젝트를 통해
실무 적응력을 향상 시킵니다.
- SSAFY는
취업 경쟁력을 높일 수 있는 효율적인
취업지원 서비스를 제공합니다.

고용노동부의 취업지원 노하우를
기반으로
교육생에게 최적의 일자리
정보를 제공하고
취업 실천 교육과
컨설팅 서비스를 통해 취업에
성공하도록 지원합니다.



홈 > About SSAFY > SSAFY 소개

삼성 청년 SW 아카데미(SSAFY)는 삼성의 SW 교육 경험과
고용노동부의 취업지원 노하우를 바탕으로 취업 준비생에게 SW 역량 향상 교육 및
다양한 취업지원 서비스를 제공하여 취업에 성공하도록 돕는 프로그램입니다.



SSAFY는
최고 수준의 교육을 제공합니다.

전문분야별 자문교수단과 삼성의 SW
전문가가 함께 참여한 명품 커리큘럼을
제공하여 경쟁력 있는 차세대 SW 인력을
양성합니다.



SSAFY는
맞춤형 교육을 제공합니다.

SW 전공자와 비전공자의 수준에 따라
맞춤형 교육을 제공하여
최적의 학습 효과를 지향합니다.



SSAFY는
자기주도적 학습을 지향합니다.

단순히 지식을 전달하기 보다
스스로 문제를 해결할 수 있는 역량을
강화시키고, 기업에서 실제로 수행하는 형태의
프로젝트를 통해 실무 적응력을 향상 시킵니다.



SSAFY는
취업 경쟁력을 높일 수 있는 효율적인
취업지원 서비스를 제공합니다.

고용노동부의 취업지원 노하우를
기반으로 교육생에게 최적의 일자리
정보를 제공하고 취업 실천 교육과
컨설팅 서비스를 통해 취업에 성공하도록
지원합니다.

✓ CSS

- 웹 문서의 내용과 관계없이 디자인만 바꿀 수 있음.
- 다양한 기기에 맞게 반응형으로 바뀌는 문서를 만들 수 있음.
- History
 - 1997 ~ 2001, HTML 4.01, CSS1
 - 2001 ~ 2006, XHTML 1, CSS2
 - 2007 ~ , HTML5, CSS3
- 웹 브라우저 별 CSS3 지원.

**CSS
IS
AWESOME**

✓ 기본 구조

선택자(Selector)

`.box` {

속성(Property) 값(Value)

`background-color: red;`

`width: 100px;`

`height: 100px;`

선언(Declaration)

`/* CSS 주석 */`

}

✓ 적용 방법 (External style sheet)

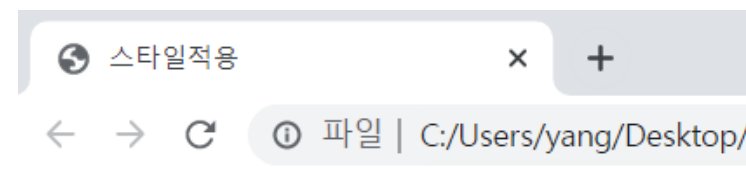
- <link>를 사용하여 외부 스타일 시트 적용
- <head> 안에 작성

index.html > ...

```
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8">
5   <title>스타일 적용</title>
6   <link rel="stylesheet" href="external.css">
7 </head>
8 <body>
9   <h1>외부 스타일 시트</h1>
10  <p>이곳은 내용입니다.</p>
11 </body>
12 </html>
```

external.css > ...

```
1 h1 {
2   color: red;
3 }
4
5 p {
6   background-color: rgb(42, 100, 226);
7   color: white;
8 }
```

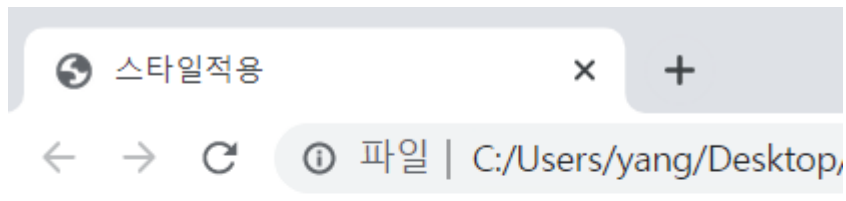


외부 스타일 시트

이곳은 내용입니다.

✓ 적용 방법 (Internal style sheet)

- 파일 내에 스타일을 적용하는 방식
- <style> 태그 사이에 CSS 규칙 작성
- <head> 안에 작성
- 외부 스타일 시트보다 우선 적용



내부 스타일 시트

이곳은 내용입니다.

```
index.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3  <head>
4    <meta charset="UTF-8" />
5    <title>스타일 적용</title>
6    <style>
7      h1 {
8        color: red;
9      }
10     p {
11       background-color: rgb(42, 100, 226);
12       color: white;
13     }
14   </style>
15 </head>
16 <body>
17   <h1>내부 스타일 시트</h1>
18   <p>이곳은 내용입니다.</p>
19 </body>
20 </html>
```

✓ 적용 방법 (Inline style)

- tag에서 style 속성을 사용하고 속성값으로 CSS 규칙 작성
- 스타일 적용 우선순위는 '인라인 → 내부 스타일 시트 → 외부 스타일 시트' 순

index.html > ...

```
1 <!DOCTYPE html>
2 <html lang="ko">
3   <head>
4     <meta charset="UTF-8" />
5     <title>스타일 적용</title>
6   </head>
7   <body>
8     <h1 style="color: red">인라인 스타일</h1>
9     <p style="background-color: rgb(42, 100, 226); color: white">
10      | 이곳은 내용입니다.
11     </p>
12   </body>
13 </html>
```

스타일적용

← → ↻ ⓘ 파일 | C:/Users/yang/Desktop/

인라인 스타일

이곳은 내용입니다.

CSS Selector

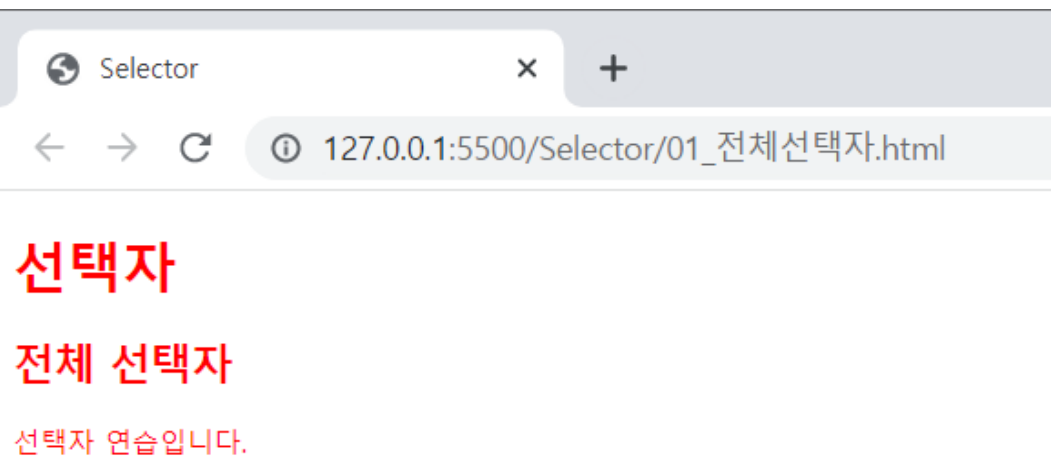
✓ CSS 선택자

- HTML 문서에서 CSS 규칙을 적용할 요소를 정의
- 기본 선택자
 - 전체 선택자
 - 유형 선택자
 - 클래스 선택자
 - id 선택자
 - 특성 선택자
- 그룹 선택자
- 결합자
 - 자손 결합자, 자식 결합자
 - 일반 형제 결합자, 인접 형제 결합자, 열 결합자
- 의사 클래스 / 요소
 - 의사 클래스
 - 의사 요소

✓ Universal selector (전체 선택자)

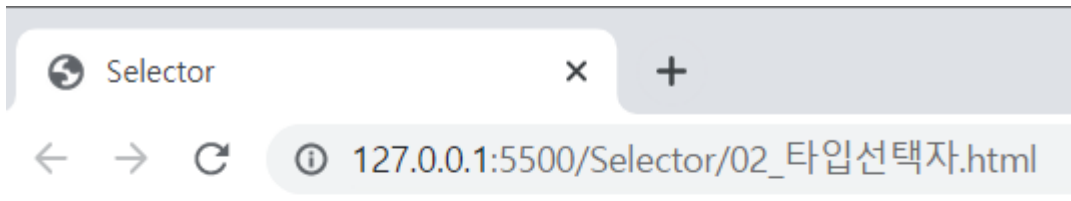
- HTML 문서 내 모든 element를 선택
- 사용법 → * { style properties }

```
Selector > 01_전체선택자.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3    <head>
4      <meta charset="UTF-8" />
5      <title>Selector</title>
6      <style>
7        * {
8          color: red;
9        }
10     </style>
11   </head>
12   <body>
13     <h1>선택자</h1>
14     <h2>전체 선택자</h2>
15     <p>선택자 연습입니다.</p>
16   </body>
17 </html>
```



✓ Type selector (유형 선택자)

- 태그명을 이용하여 스타일을 적용할 태그를 선택
- HTML 내에서 주어진 유형의 모든 요소를 선택
- 사용법 → element { style properties }



선택자

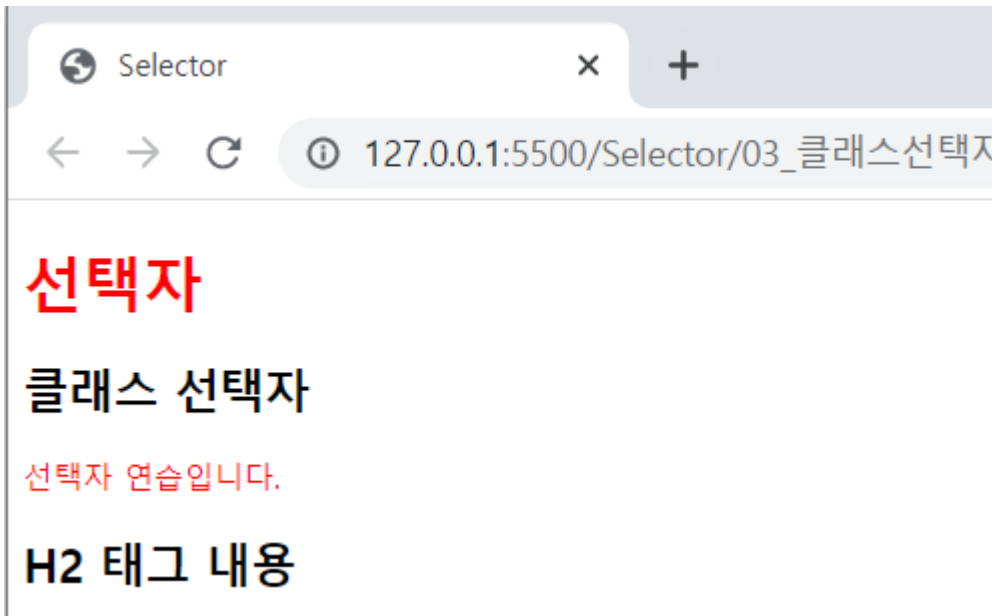
유형 선택자

선택자 연습입니다.

H2 태그 내용

```
Selector > 02_타입선택자.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3    <head>
4      <meta charset="UTF-8" />
5      <title>Selector</title>
6      <style>
7        h2 {
8          color: red;
9        }
10     </style>
11   </head>
12   <body>
13     <h1>선택자</h1>
14     <h2>유형 선택자</h2>
15     <p>선택자 연습입니다.</p>
16     <h2>H2 태그 내용</h2>
17   </body>
18 </html>
```

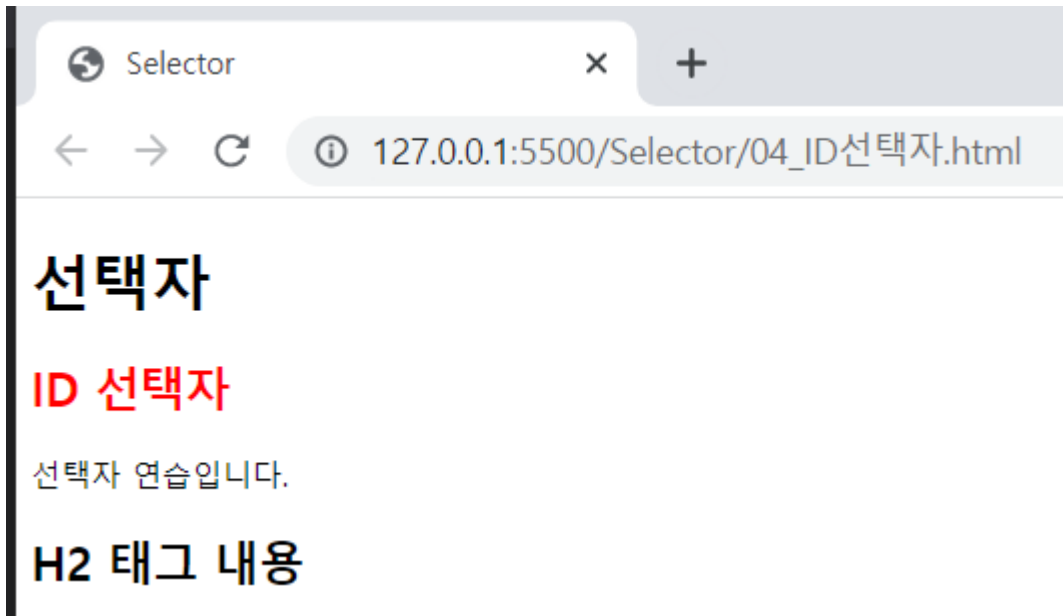
- ✓ Class selector (클래스 선택자)
 - class 가 적용된 모든 태그를 선택
 - HTML 내에서 동일한 클래스 명을 중복해서 사용가능
 - 사용법 → .class-name { style properties }



```
Selector > 03_클래스선택자.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3    <head>
4      <meta charset="UTF-8" />
5      <title>Selector</title>
6      <style>
7        .red {
8          color: red;
9        }
10     </style>
11   </head>
12   <body>
13     <h1 class="red">선택자</h1>
14     <h2>클래스 선택자</h2>
15     <p class="red">선택자 연습입니다.</p>
16     <h2>H2 태그 내용</h2>
17   </body>
18 </html>
```


✓ ID selector (ID 선택자)

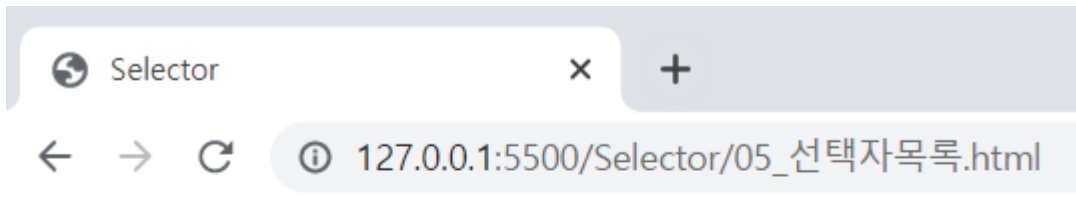
- id 특성 값을 비교하여, 동일한 id를 가진 태그를 선택
- HTML 내에서 주어진 ID를 가진 요소가 하나만 존재 해야함
- 사용법 → #id-value { style properties }



```
Selector > 04_ID선택자.html > ...
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8" />
5   <title>Selector</title>
6 <style>
7   #title {
8     color: red;
9   }
10 </style>
11 </head>
12 <body>
13   <h1>선택자</h1>
14   <h2 id="title">ID 선택자</h2>
15   <p>선택자 연습입니다.</p>
16   <h2>H2 태그 내용</h2>
17 </body>
18 </html>
```

✓ Selector list (선택자 목록)

- , 를 이용하여 선택자 그룹을 생성하는 방법
- 모든 일치하는 노드를 선택
- 사용법 → element, element { style properties }



선택자

선택자 목록

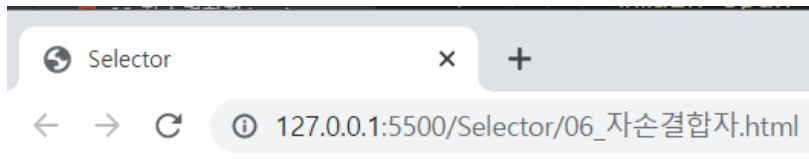
선택자 연습입니다.

H2 태그 내용

```
Selector > 05_선택자목록.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3    <head>
4      <meta charset="UTF-8" />
5      <title>Selector</title>
6      <style>
7        h1, p {
8          color: red;
9        }
10     </style>
11   </head>
12   <body>
13     <h1>선택자</h1>
14     <h2>선택자 목록</h2>
15     <p>선택자 연습입니다.</p>
16     <h2>H2 태그 내용</h2>
17   </body>
18 </html>
```

✓ Descendant combinator (자손 결합자)

- 첫 번째 요소의 자손인 노드를 선택
- 사용법 → selector1 selector2 { style properties }



선택자

자손 결합자

div 태그

span 태그

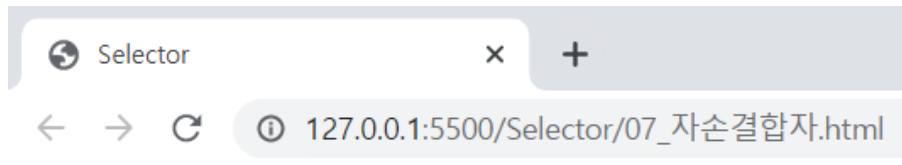
- li 태그1
- li 태그2
- li 태그3
- li 태그4
- li 태그5

div 태그

```
Selector > 06_자손결합자.html > ...
1  <!DOCTYPE html>
2  <html lang="ko">
3    <head>
4      <meta charset="UTF-8" />
5      <title>Selector</title>
6      <style>
7        #main span {
8          color: red;
9        }
10     </style>
11   </head>
12   <body>
13     <h1>선택자</h1>
14     <h2>자손 결합자</h2>
15     <div id="main">
16       <div>div 태그</div>
17       <span>span 태그</span>
18       <ul>
19         <li>li 태그1</li>
20         <li><span>li 태그2</span></li>
21         <li><span>li 태그3</span></li>
22         <li>li 태그4</li>
23         <li><span>li 태그5</span></li>
24       </ul>
25       <div>div 태그</div>
26     </div>
27   </body>
28 </html>
```

✓ Child combinator (자식 결합자)

- 첫 번째 요소의 바로 아래 자식인 노드를 선택
- 사용법 → `selector1 > selector2 { style properties }`



선택자

자식 결합자

div 태그

span 태그

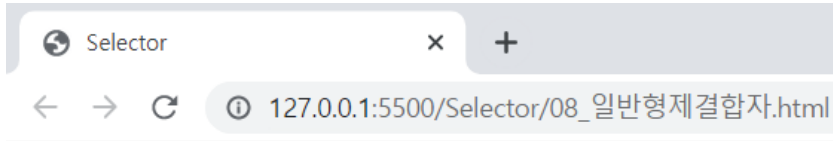
- li 태그1
- li 태그2
- li 태그3
- li 태그4
- li 태그5

div 태그

```
Selector > 07_자손결합자.html > ...
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8" />
5   <title>Selector</title>
6   <style>
7     #main > span {
8       color: red;
9     }
10  </style>
11 </head>
12 <body>
13   <h1>선택자</h1>
14   <h2>자식 결합자</h2>
15   <div id="main">
16     <div>div 태그</div>
17     <span>span 태그</span>
18     <ul>
19       <li>li 태그1</li>
20       <li><span>li 태그2</span></li>
21       <li><span>li 태그3</span></li>
22       <li>li 태그4</li>
23       <li><span>li 태그5</span></li>
24     </ul>
25     <div>div 태그</div>
26   </div>
27 </body>
28 </html>
```

✓ General sibling combinator (일반 형제 결합자)

- 첫 번째 요소를 뒤따르면서 같은 부모를 공유하는 두 번째 요소를 모두 선택
- 사용법 → `former-element ~ target-element { style properties }`



선택자

일반 형제 결합자

span 태그

div 태그

span 태그

- li 태그1
- li 태그2
- li 태그3
- li 태그4
- li 태그5

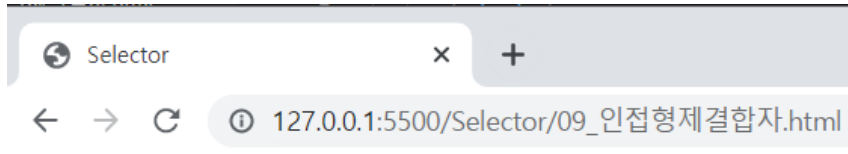
div 태그

span 태그

```
Selector > 08_일반형제결합자.html > ...
1 <!DOCTYPE html>
2 <html lang="ko">
3   <head>
4     <meta charset="UTF-8" />
5     <title>Selector</title>
6     <style>
7       #target ~ span {
8         color: red;
9       }
10    </style>
11  </head>
12  <body>
13    <h1>선택자</h1>
14    <h2>일반 형제 결합자</h2>
15    <div id="main">
16      <span>span 태그</span>
17      <div id="target">div 태그</div>
18      <span>span 태그</span>
19      <ul>
20        <li>li 태그1</li>
21        <li><span>li 태그2</span></li>
22        <li><span>li 태그3</span></li>
23        <li>li 태그4</li>
24        <li><span>li 태그5</span></li>
25      </ul>
26      <div>div 태그</div>
27      <span>span 태그</span>
28    </div>
29  </body>
30 </html>
```

✓ Adjacent sibling combinator (인접 형제 결합자)

- 첫 번째 요소의 바로 뒤에 위치하면서 같은 부모를 공유하는 두번째 요소 선택
- 사용법 → `former-element + target-element { style properties }`



선택자

인접 형제 결합자

span 태그

div 태그

span 태그

- li 태그1
- li 태그2
- li 태그3
- li 태그4
- li 태그5

div 태그

span 태그

```
Selector > 09_인접형제결합자.html > ...
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8" />
5   <title>Selector</title>
6   <style>
7     #target + span {
8       color: red;
9     }
10  </style>
11 </head>
12 <body>
13   <h1>선택자</h1>
14   <h2>인접 형제 결합자</h2>
15   <div id="main">
16     <span>span 태그</span>
17     <div id="target">div 태그</div>
18     <span>span 태그</span>
19     <ul>
20       <li>li 태그1</li>
21       <li><span>li 태그2</span></li>
22       <li><span>li 태그3</span></li>
23       <li>li 태그4</li>
24       <li><span>li 태그5</span></li>
25     </ul>
26   </div>
27   <span>span 태그</span>
28 </div>
29 </body>
30 </html>
```

✓ 우선순위

- 같은 요소에 두 개 이상의 CSS 규칙이 적용된 경우
마지막 규칙, 구체적인 규칙, !important 가 우선 적용

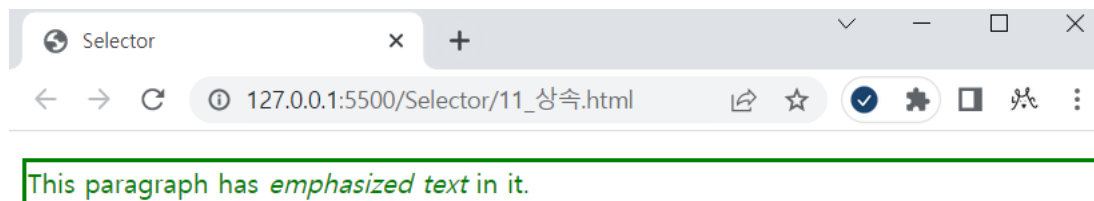
```
<style>
  div {
    width: 100px;
    height: 100px;
    background-color: orange;
    margin-bottom: 10px;
  }
  .red {
    background-color: red;
  }
  .blue {
    background-color: blue;
  }
  #green {
    background-color: green;
  }
  .yellow {
    background-color: yellow !important;
  }
</style>
```

```
<body>
  <div></div>
  <div class="red"></div>
  <div class="red blue"></div>
  <div class="blue red"></div>
  <div class="red" id="green"></div>
  <div class="red" id="green" style="background-color: grey"></div>
  <div class="yellow" id="green"></div>
  <div class="yellow" id="green" style="background-color: darkviolet"></div>
</body>
```

✓ Inheritance

- 부모 요소에 적용된 스타일이 자식 요소에게 상속이 될 수도 있고, 안 될 수도 있음.
- 상속 되는 속성
 - 요소의 상속되는 속성에 값이 지정되지 않은 경우, 요소는 부모 요소의 해당 속성의 계산 값을 얻음.
 - 대표적인 예는 color 속성
- 상속 되지 않는 속성
 - 요소의 상속되지 않는 속성에 어떤 값이 지정되지 않는 경우, 요소는 그 속성의 초기값을 얻음.
 - 대표적인 예는 border 속성

```
<style>
p {
  color: green;
  border: medium solid;
}
</style>
```



```
<body>
  <p>This paragraph has <em>emphasized text</em> in it.</p>
</body>
```


다음 방송에서 만나요!

삼성 청년 SW 아카데미