

화면 설계

chapter02

HTML 개요

Contents

part.1

HTML 구조

part.2

HTML, CSS, JS 연결하기

part.3

<head>에 들어가는 태그

part.4

화면에 이미지 출력하기

part.5

상대경로와 절대경로

part.6

페이지 나누고 연결하기

part.7

개발자 도구 사용하기

학습목표

이장에서는 코드의 구조를 학습하고 HTML, CSS, JS를 연결한다.
그리고 HTML 파일이 브라우저위에서 어떻게 출력되는지 확인한다.

```
<!DOCTYPE html>  
<html>  
  <head>  
  
  </head>  
  <body>  
    hello world!  
  </body>  
</html>
```

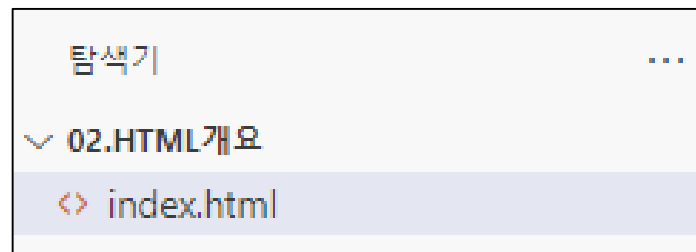
HTML문서(편집기)

← → ↻ ⓘ 127.0.0.1:5500/index.html

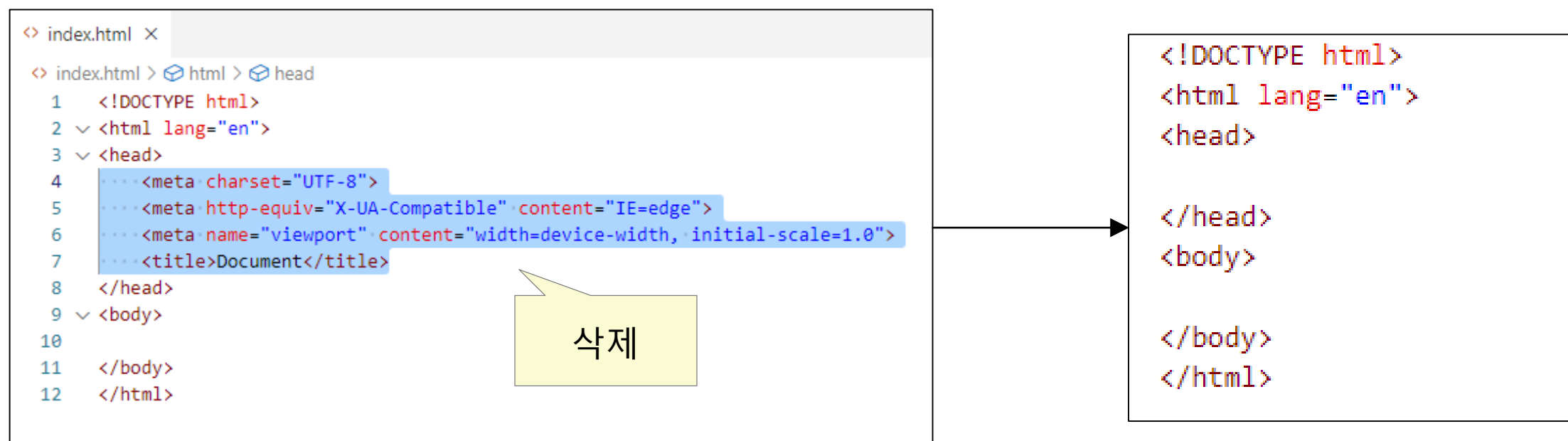
hello world!

브라우저(뷰어)

1. [02.HTML개요]라는 이름으로 프로젝트를 만든다.
2. 프로젝트에 [index.html] 파일을 추가한다.



3. 태그의 구조를 선명하게 보기 위해 `<head></head>` 태그 안의 내용을 모두 삭제한다.



<!DOCTYPE html>

문서의 html 버전을 지정한다.

HTML버전에 따라 브라우저에서 해석되는 방식이 달라진다.

해당 버전은 HTML5을 의미하고, <!DOCTYPE html PUBLIC "-//W3C//... " 는 옛날 버전인 XHTML을 의미한다.

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

<html>

문서의 시작과 끝, 전체 범위를 나타내는 부분이다.

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

<head>

문서의 정보를 포함하는 부분이다.

문서제목, 스타일 및 스크립트 연결, 인코딩과 같이 눈에 보이지 않는 정보가 들어간다.

```
<!DOCTYPE html>
<html>
  <head>

  </head>
  <body>
  </body>
</html>
```

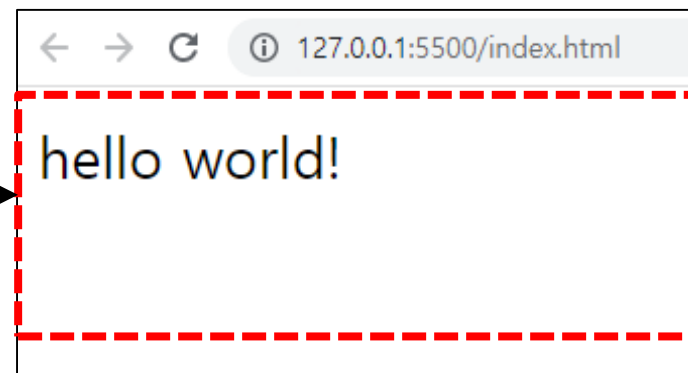
<body>

화면에 표시되는 콘텐츠를 포함하는 부분이다.

이부분에는 다양한 요소들이 (이미지, 링크, 버튼, 테이블) 등이 들어간다.

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>

  </body>
</html>
```

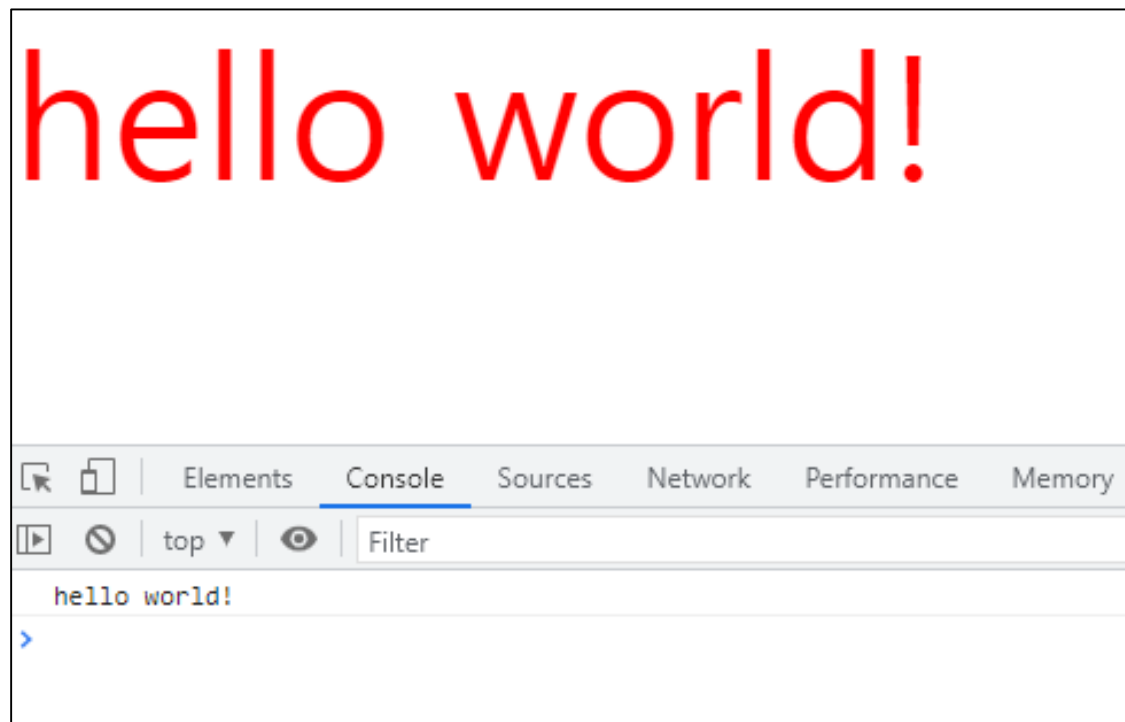


HTML 연결하기

HTML, CSS, JS 구조

- HTML, CSS, JS 파일은 서로 연결되어야 하며, 함께 동작하여 하나의 페이지를 만든다.
- HTML은 페이지의 뼈대를 만들 수 있고, 혼자서 동작할 수 있다.
- CSS와 JS는 HTML에 종속되어 있으므로 혼자서 동작할 수 없다.

HTML



CSS

```
# main.css > ...  
1  div {  
2      color: red;  
3      font-size: 100px;  
4  }
```

JAVASCRIPT

```
JS main.js  X  
js > JS main.js  
1  console.log('hello world!');  
2  
3
```

HTML과 CSS 파일을 연결하는 방법을 알아본다.

1. index.html 파일을 연다.
2. body태그 안에 div 태그를 입력하고, div 태그 안에 "hello world!"를 입력한다.

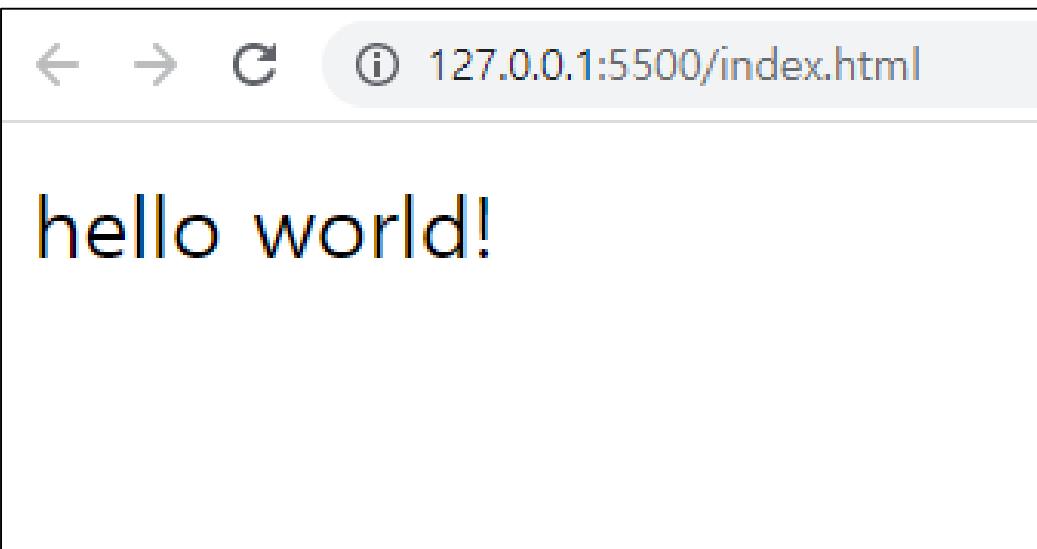
```
<body>
```

```
  <div>hello world!</div>
```

Enter키로 자동완성

```
</body>
```

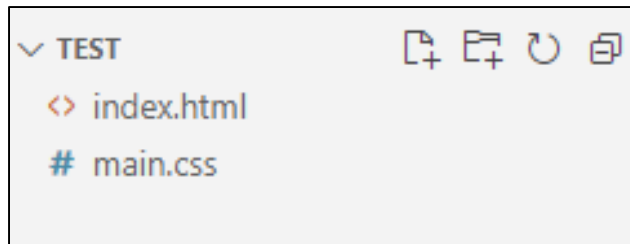
3. 상태바의 [Go Live] 버튼을 클릭하여 웹페이지를 실행한다.
브라우저가 자동으로 열리면서 hello world!라는 글자가 나타난다.



HTML 연결하기

HTML과 CSS 연결하기

4. 사이드바에서 파일 추가 버튼을 클릭하고 [main.css] 라는 이름의 파일을 생성한다.



5. 패널에 div를 입력하고 띄어쓰기 후 { 여는 중괄호를 입력하면 자동으로 } 닫는 중괄호가 만들어진다.
중괄호 사이에 엔터입력하면 자동으로 들여쓰기가 된다.



6. div의 글자 색은 빨강으로, 크기는 100px로 지정한다.

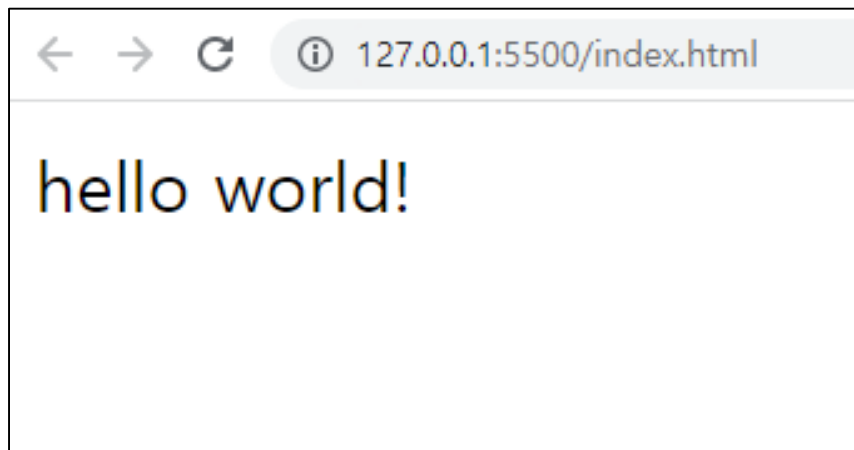
```
div {  
  color: red;  
  font-size: 100px;  
}
```

color와 font-size는 속성을 의미함
red와 100px는 값을 의미함

7. 그런데 브라우저를 보면 이전 화면과 달라진 점이 없다.

왜 변한 것이 없을까? 이유는 앞서 만들었던 HTML파일과 CSS파일이 연결되지 않았기 때문이다.

앞에서 설명했던 것과 같이 css와 js는 혼자 실행할 수 없다.



8. 다시 index.html로 돌아와서 <head>에서 main.css 파일을 연결한다.

head태그 안에 link태그를 입력한다.

```
<head>  
|   link  
</head>
```

link 입력 후 엔터키로 태그 자동완성

9. link 태그의 href 속성에 “./main.css” 라고 입력한다.

```
<head>  
  <link rel="stylesheet" href="./main.css">  
</head>
```

./는 현재 폴더를 의미

10. 다시 브라우저를 보면 글자의 색과 크기가 바뀌었다.



HTML 연결하기

HTML과 JS 연결하기

이번에는 html과 js을 연결한다.

1. 파일 추가 버튼을 클릭하고 [main.js] 라는 이름의 파일을 생성한다.

main.css

```
JS main.js  X  <> index.html  # main.css
js > JS main.js
1  console.log('hello world!');
2
3
```

2. 다시 index.html로 가서 script 태그를 추가한다.

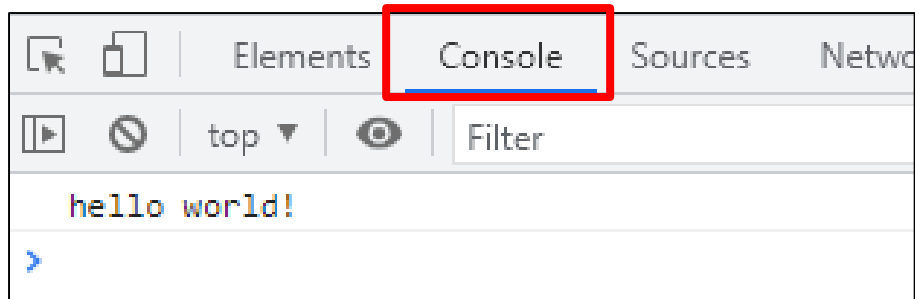
3. script태그에 src속성을 추가하고 "./main.js"를 입력한다.

index.html

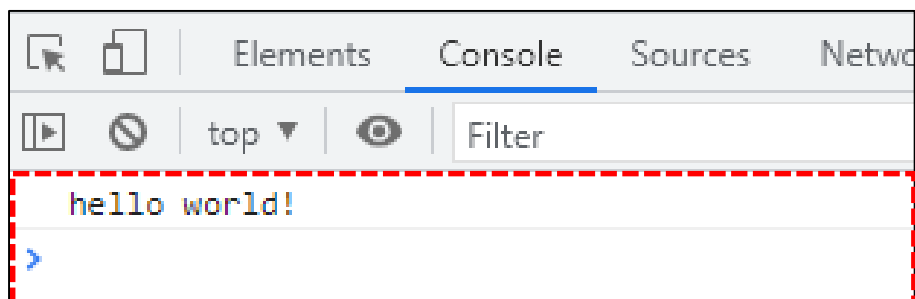
```
<head>
  <link rel="stylesheet" href="./main.css">
  script
</head>
<body>
```

→ `<script src="./main.js"></script>`

4. 브라우저로 가서 F12키를 눌러 개발자도구를 연다.
5. 상단에서 [console] 탭을 클릭한다.



6. main.js에 작성했던 텍스트가 출력된 것을 확인한다.



<head>에 들어가는 태그

<head> 태그 사이에 들어가는 요소들을 알아본다.

1) <title> 태그

HTML 문서의 제목을 정의하는 태그이다.

이 태그 사이에 입력한 내용은 브라우저 탭에 표시된다.

```
<head>  
  <title>TEST</title>  
</head>
```



<head>에 들어가는 태그

2) <link> 태그

CSS, 이미지 등 외부 파일을 가져와서 연결할 때 사용하는 태그이다.

rel 속성은 파일의 종류를 의미하고, href 속성은 파일의 경로를 의미한다.

```
<head>
  <link rel="stylesheet" href="./main.css">
</head>
```

3) <style> 태그

CSS와 관련된 내용을 외부에서 가져오는 것이 아니라, HTML 문서 안에서 직접 작성할 때는 사용한다.

다음과 같이 <style>태그 사이에 css 내용을 작성한다.

```
<head>
  <!-- <link rel="stylesheet" href="./main.css"> -->
  <style>
    div {
      color: blue;
    }
  </style>
</head>
```



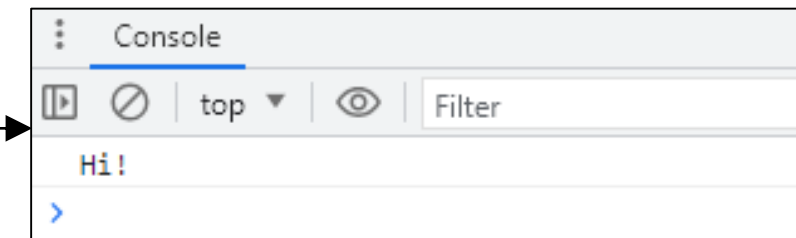
<head>에 들어가는 태그

4) <script> 태그

JS와 관련된 내용을 외부에서 가져오는 것이 아니라, HTML 문서 안에서 직접 작성할 때는 사용한다.

다음과 같이 <script>태그 사이에 js 내용을 작성한다.

```
<head>
  <!-- <script src="./main.js"></script> -->
  <script>
    console.log('Hi!');
  </script>
</head>
```



5) <meta> 태그

Meta태그는 문서의 제작자, 설명, 문자인코딩 같은 정보를 나타낸다.

name속성은 정보의 종류, content속성은 정보의 값을 의미한다.

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
```

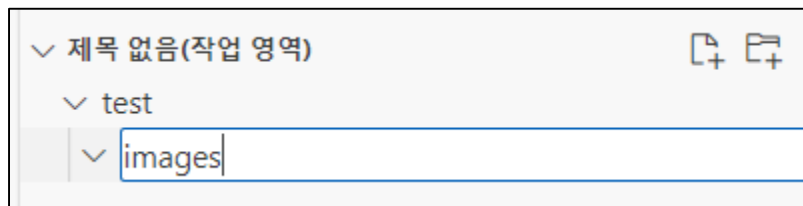

화면에 이미지 출력하기 준비하기

이번에는 화면에 이미지를 출력해본다.

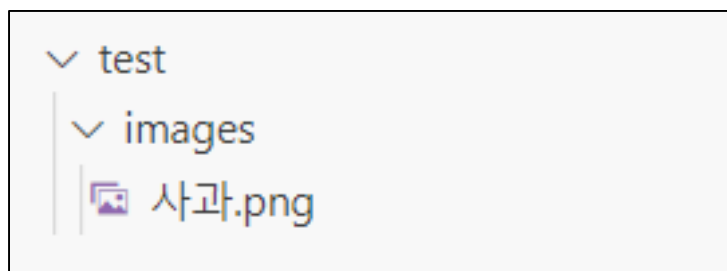
1. 사용할 이미지(사과.png)는 구글드라이브-수업자료에서 다운로드한다.



2. VS Code로 이동하고 폴더 추가 버튼을 클릭하고 [images]라는 폴더를 생성한다.



3. 드래그하여 images 폴더 밑에 사과 이미지를 넣는다.



화면에 이미지 출력하기 준비하기

프로젝트 안에 있는 파일들은 목적에 따라 폴더를 만들어서 정리해야 한다.

4. css 라는 이름의 폴더를 만들고 밑에 main.css 파일을 넣는다.
5. js 라는 이름의 폴더를 만들고 밑에 main.js 파일을 넣는다.



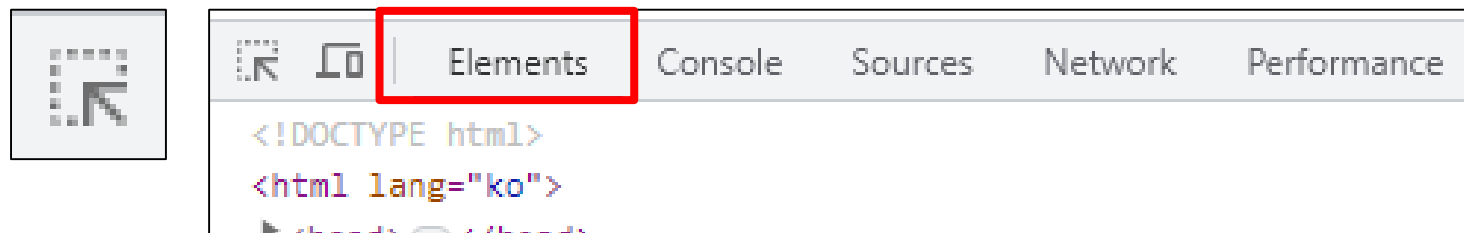
6. 다시 index.html로 가서 파일의 경로를 변경한다.

```
<head>
  <title>Document</title>
  <link rel="stylesheet" href="./css/main.css">
  <script src="./js/main.js"></script>
</head>
```

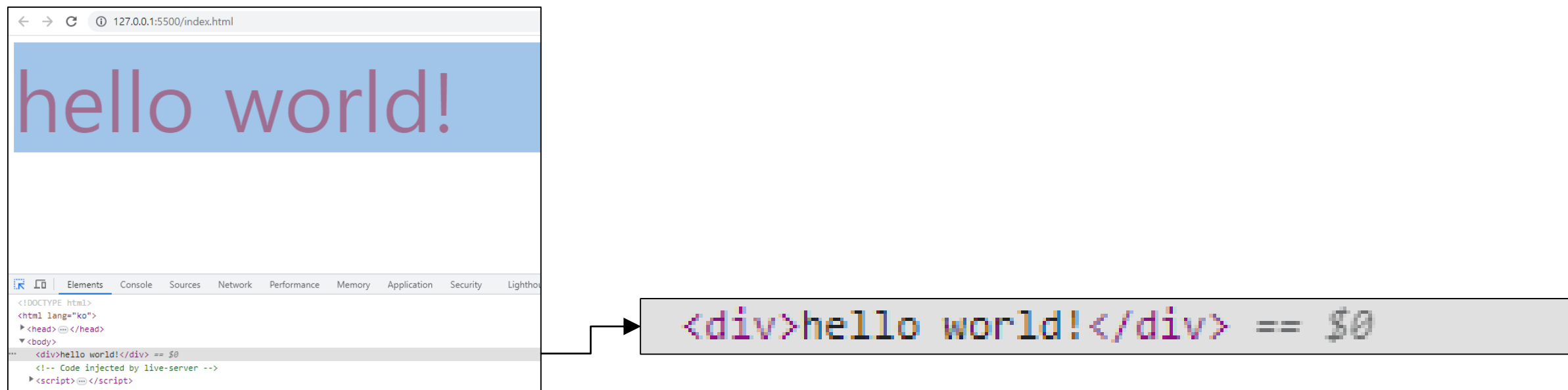
속성을 입력할 때 현재 폴더에 있는
파일, 폴더 목록이 자동으로 나타남

화면에 이미지 출력하기 준비하기

7. index.html 파일을 열고 [Go Live] 버튼을 클릭한다.
8. 브라우저에서 [F12] 키를 누르면 개발자 도구가 열린다.
9. [Elements] 탭을 선택한 상태에서 Select 아이콘을 클릭한다.



10. 이 상태로 브라우저에서 마우스 커서를 이동시키면 웹페이지의 요소가 어떤 코드로 작성되었는지 볼 수 있다.



화면에 이미지 출력하기

이미지 출력하기

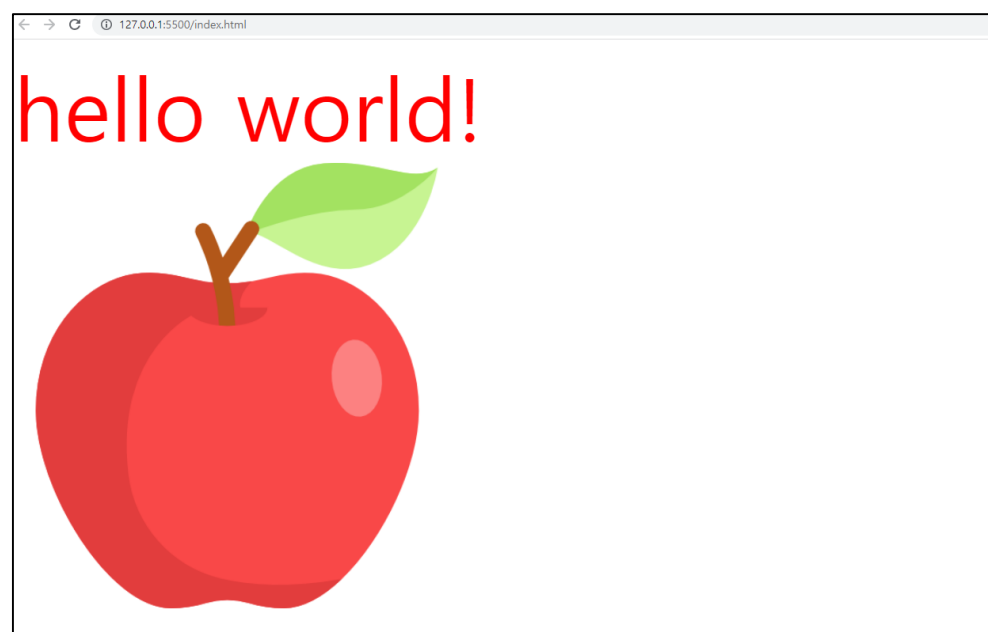
이제 다운로드한 이미지를 화면에 출력한다.

1. body 태그 안에 img태그를 만든다.
2. src 속성에 이미지 파일 경로를 입력하고, alt 속성에 이미지를 찾지 못할 때 이미지대신 보여줄 문자를 입력한다.
src에 다운로드한 이미지경로를 넣고, alt에 "이미지를 찾을 수 없습니다"라는 문자열을 넣는다.

```

```

3. 브라우저를 새로고침하면 다음과 같이 이미지 파일이 나타난다.



A rectangular box containing a small icon of a broken image followed by the Korean text '이미지를 찾을 수 없습니다' (Image not found).

상대경로와 절대경로

상대경로

파일에서 다른 파일을 접근하려면 파일의 경로를 알아야 한다.

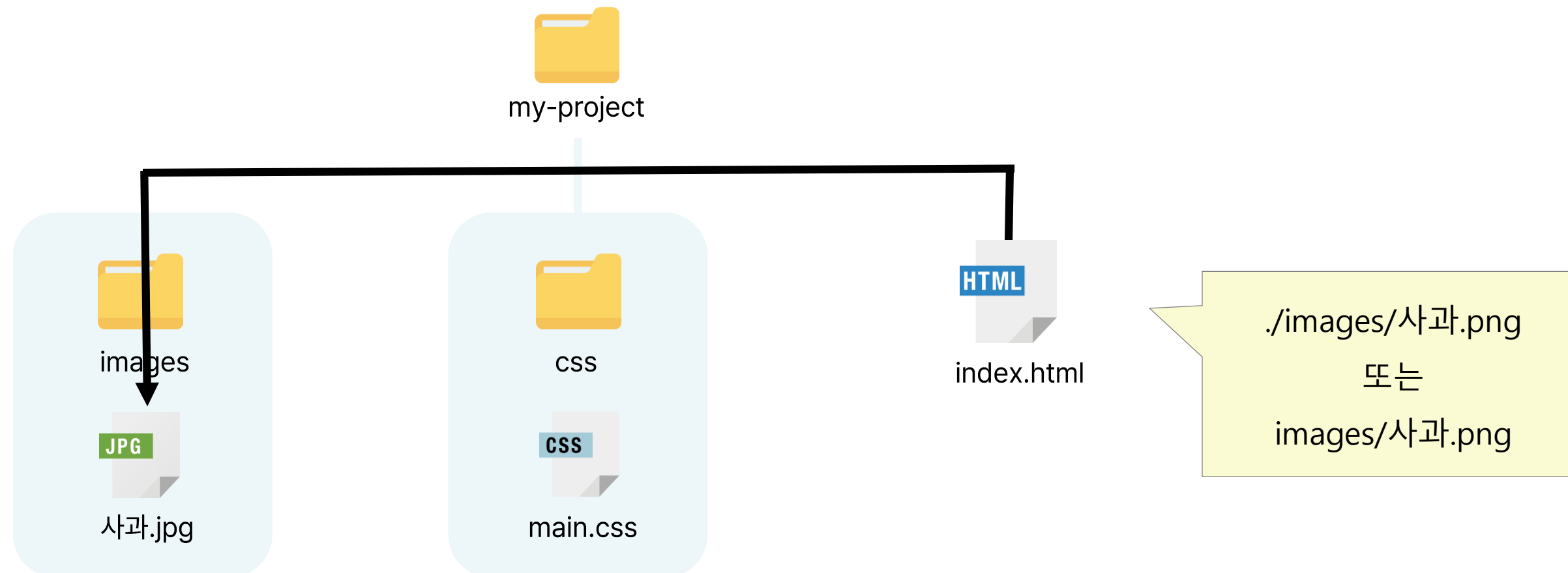
여기에서는 파일의 경로를 크게 상대경로와 절대경로로 구분하여 알아본다.

먼저 상대경로는 현재 폴더를 기준으로 다른 파일을 찾는다.

상대경로에서 사용되는 표현으로 ./ 와 ../ 가 있다.

./는 현재폴더 위치를 의미하며 생략 가능하다. ../는 상위 폴더를 의미한다.

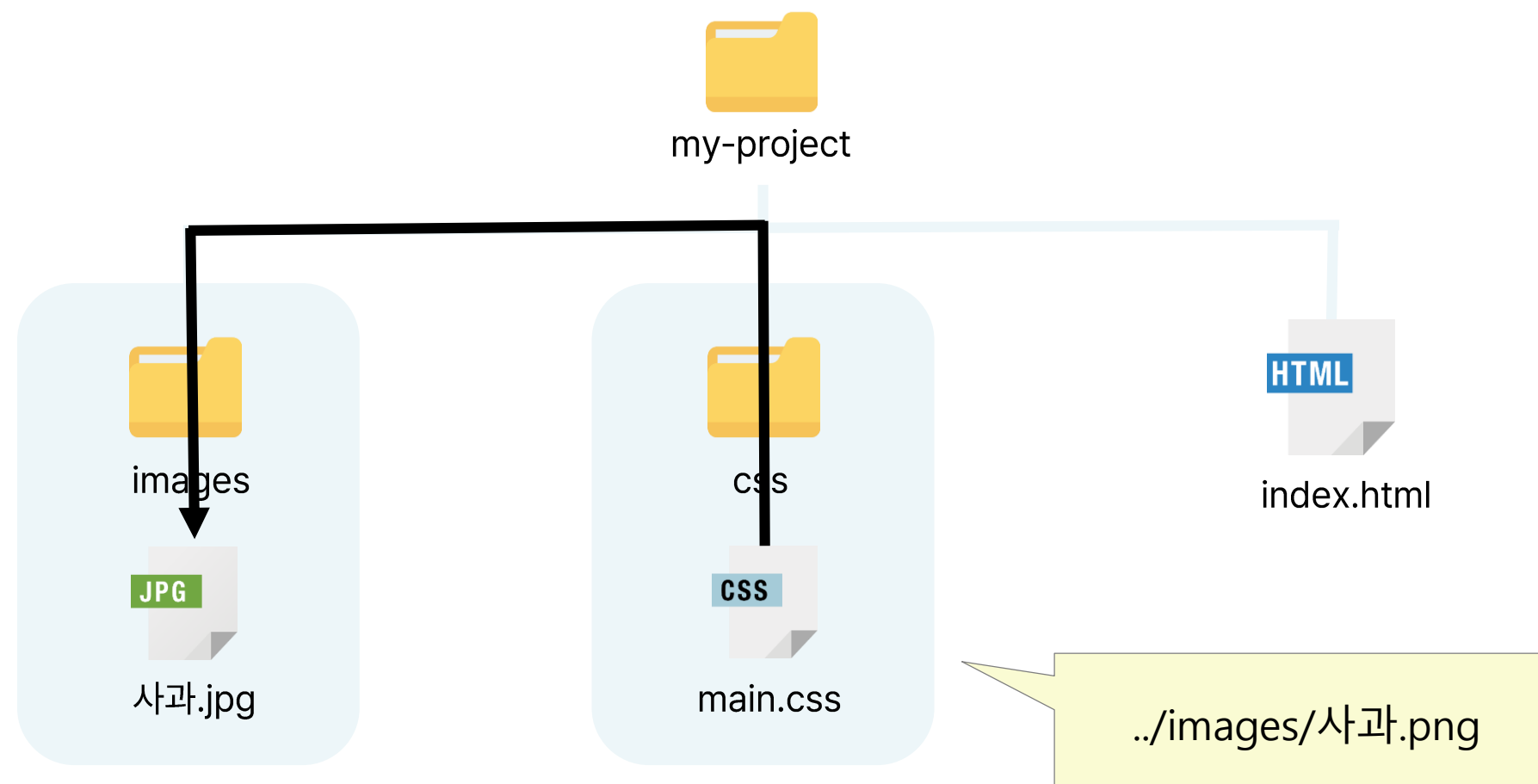
예를 들어, 다음과 같은 프로젝트 구조를 가진 index.html에서 사과.png를 찾는다면 파일의 경로는 `./images/사과.jpg` 이 된다.



상대경로와 절대경로

상대경로

이번에는 main.css 파일에서 사과.png파일의 경로를 표시한다.
같은 폴더 내에는 사과.png가 없으므로 바로 접근할 수 없다.
여기에서는 상위 경로 가서 파일을 찾는다. 즉, 경로가 "../images/사과.png"가 된다.



상대경로와 절대경로

절대경로

이번에는 index.html에서 사과.png 파일의 경로를 절대경로로 표시한다.

절대경로는 / 로 표시하는데, 최상위경로를 의미한다.

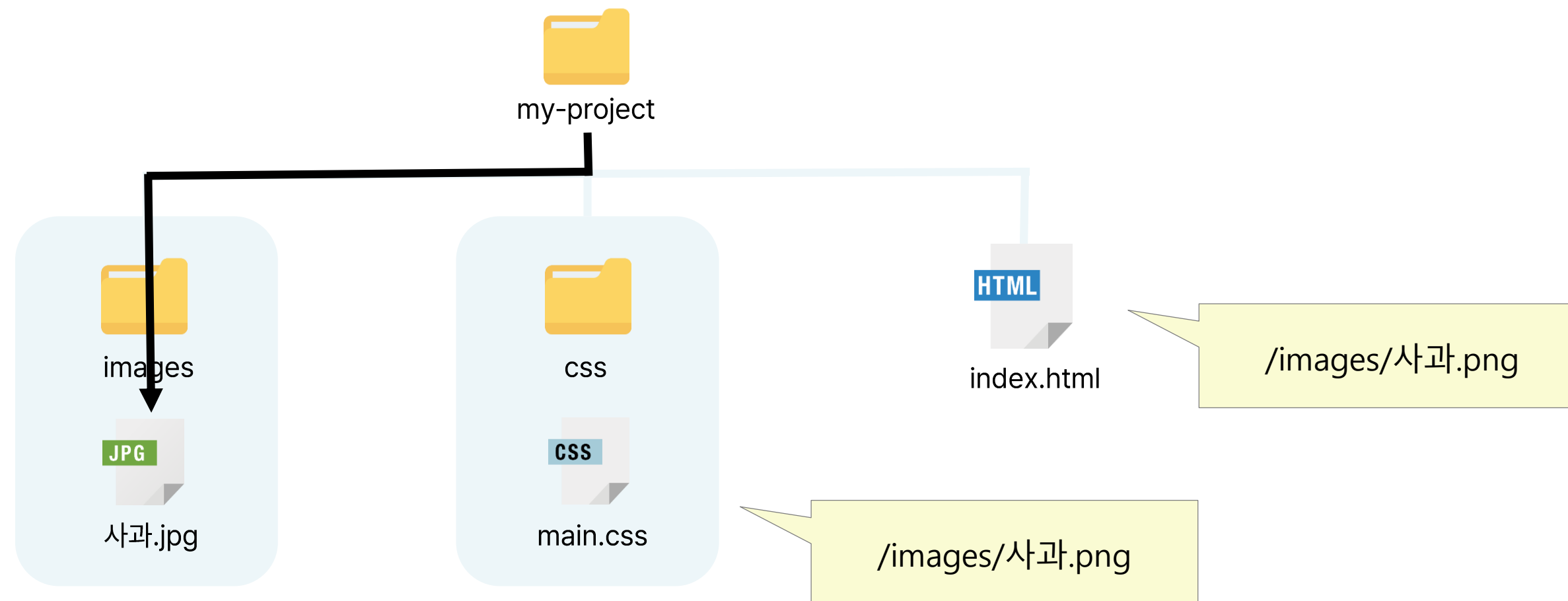
먼저 최상위경로 /를 표시하고, 그 다음에 하위폴더와 파일을 표시한다.

index.html 파일에서 절대경로로 사과.png 파일을 찾는다면 파일의 경로는 `"/images/사과.jpg"`가 된다.

절대경로는 변함이 없으므로 main.js에서도 `"/images/사과.jpg"`로 표시하면 된다.

상대경로 - `./ ../`

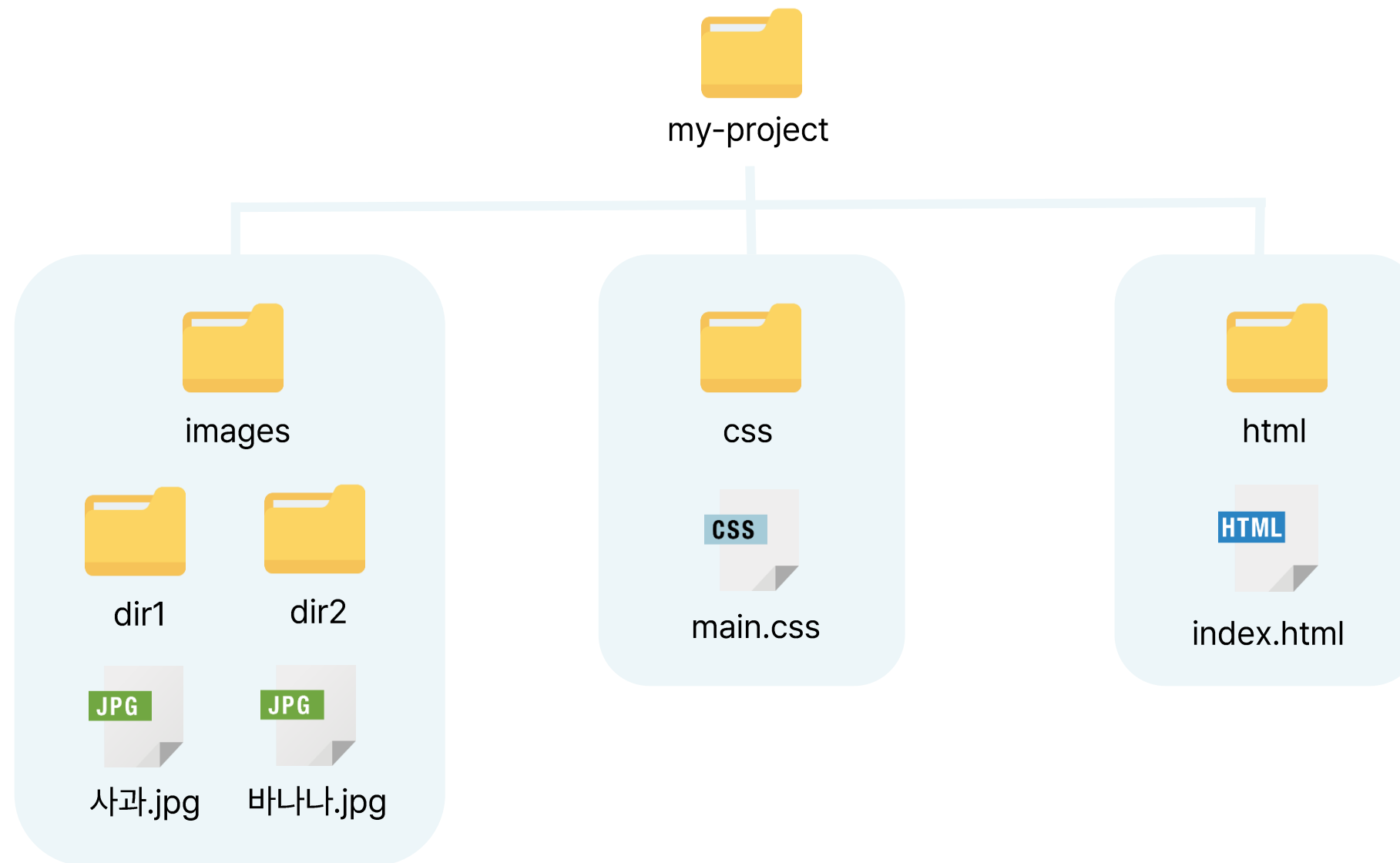
절대경로 - `/`



Q1. index.html에서 css폴더의 main.css를 연결하세요.

Q2. index.html에서 images폴더의 사과.jpg를 연결하세요.

메모장을 열고 파일 경로를
작성해보세요!



정답

1. ../css/main.css
2. ../images/dir1/사과.jpg

메인페이지와 상세페이지를 작성하고 연결하는 방법을 알아본다.

먼저 index2.html 파일을 생성한다.

```
<> index2.html x
<> index2.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
12
```

페이지 나누고 연결하기

NAVER 홈페이지로 이동하기

메인페이지에 네이버로 이동하는 링크를 만든다.

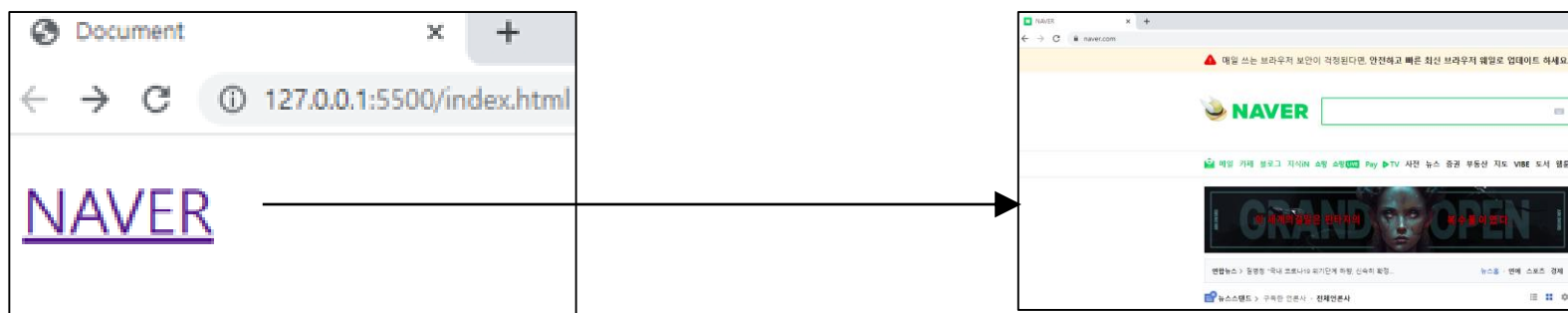
a 태그를 작성하고, href 속성에 "https://www.naver.com" 라는 주소를 입력하고, 태그 사이에는 "NAVER"를 입력한다.

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

이동할 주소 화면에 표시할 문자열

`NAVER`

브라우저를 띄우면 NAVER라는 링크가 나타난다. 링크를 클릭하면 네이버 홈페이지로 이동한다.

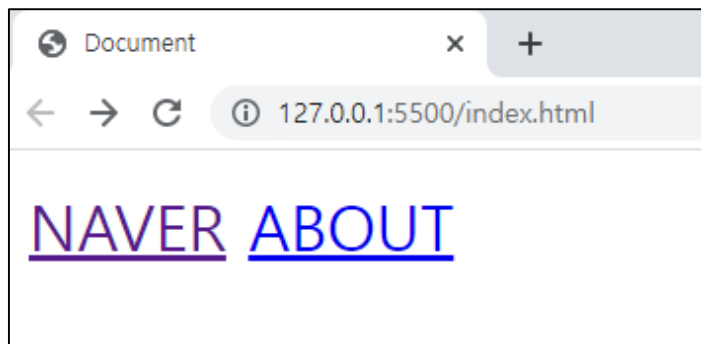


메인페이지에 상세페이지와 연결되는 새로운 링크를 추가한다.

a 태그를 작성하고, href 속성에 “./about.html” 라는 주소를 입력하고, 태그 사이에는 “ABOUT”를 입력한다.

```
<body>  
  <a href="https://www.naver.com">NAVER</a>  
  <a href="./about.html">ABOUT</a>  
</body>
```

브라우저를 새로고침하면 ABOUT이라는 링크가 나타난다.



페이지 나누고 연결하기

상세페이지와 연결하기

이제 메인페이지와 연결할 상세페이지(about.html)를 만든다.

body 태그 안에 About!!이라고 입력한다.

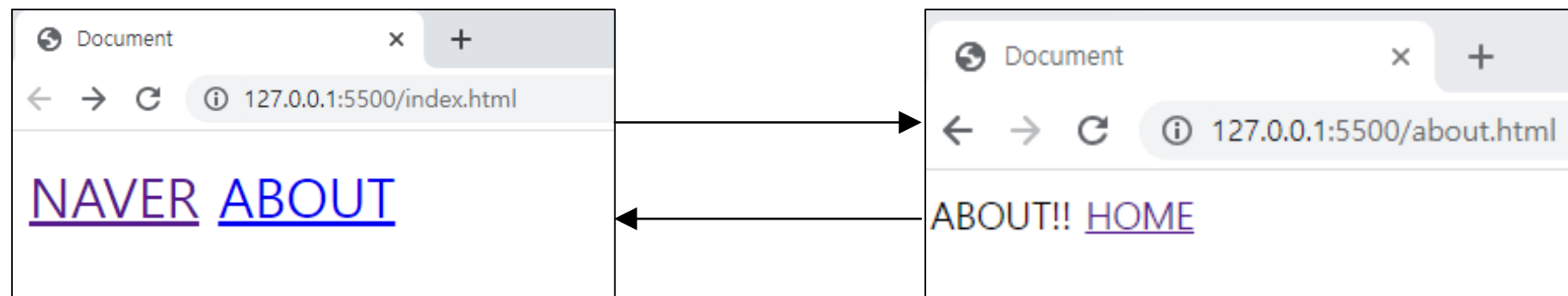
그리고 문자 위에 다시 메인페이지로 가는 링크를 만든다.

a 태그를 작성하고, href 속성에 "./index2.html"이라는 주소를 입력하고, 태그 사이에는 "HOME"를 입력한다.

```
<body>
  ABOUT!!
  <a href="./index2.html">HOME</a>
</body>
```

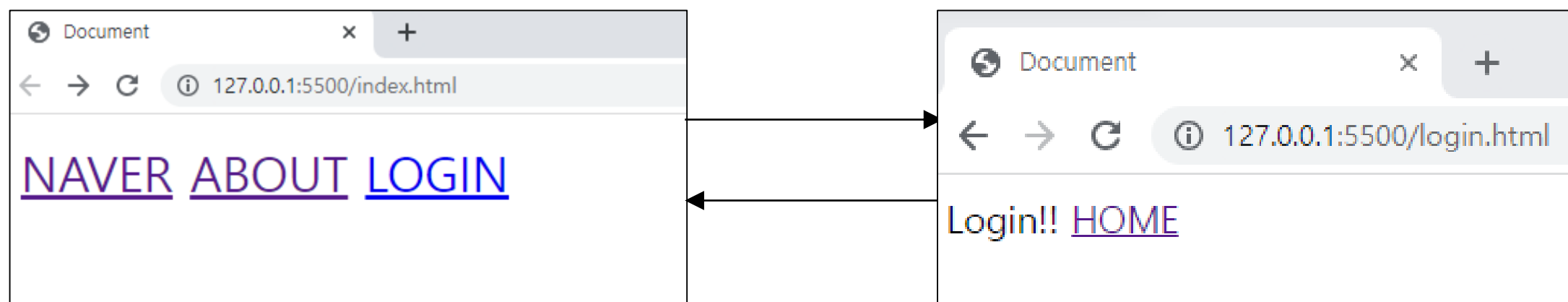
다시 브라우저로 가서 ABOUT 링크를 클릭하면, 페이지가 전환되면서 상세페이지로 이동한다.

그리고 상세페이지에서 HOME 링크를 클릭하면 메인페이지로 이동한다.



이러한 방식으로 페이지 사이를 이동할 수 있다.

Q1. 앞서 배운 a태그를 활용하여, 다음과 같이 새로운 상세페이지(login.html)를 만들어 보세요.
그리고 메인페이지와 로그인 페이지에서 서로 이동할 수 있도록, 두 페이지를 연결하세요.



브라우저에서 제공하는 개발자 도구를 이용하면 사이트를 분석하고 테스트할 수 있다.
F12키를 누르면 개발자도구가 나타난다. 먼저 [Elements] 탭에 대해 살펴본다.
Select아이콘을 사용하면 화면을 구성하고 있는 HTML코드 에대한 정보를 확인 할 수 있다.

NAVER ABOUT LOGIN

3. 정보를 확인하고 싶은 부분을 클릭한다

2. Select 아이콘을 클릭한다

1. Elements 탭을 누른다

4. 선택한 HTML태그가 표시된다

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <a href="https://www.naver.com">NAVER</a>
    <a href="./about.html">ABOUT</a>
    ... <a href="./login.html">LOGIN</a> == $0
    <!-- Code injected by live-server -->
    <script>...</script>
  </body>
</html>
```

Styles Computed Layout Event List

Filter

element.style {

}

a:-webkit-any-link {

color: -webkit-link;

cursor: pointer;

text-decoration: underline;

특정 html 요소에 어떤 스타일이 적용되었는지 알고 싶다면, [Styles] 탭을 사용한다.
그 전에 index.html에 div 하나를 추가하고 다음과 같이 글자에 빨강색을 적용한다.

```
<body>
  <div>hello world!</div>
</body>
```

```
<head>
  <style>
    div {
      color: red;
    }
  </style>
</head>
```

다시 개발자도구로 가서 Elements > Styles 탭을 선택한다.

Styles탭을 보면 HTML요소에 어떤 스타일이 적용되었는지 알 수 있다.

그리고 Computed탭을 보면 더 상세한 정보를 볼 수 있다.

NAVER ABOUT LOGIN
hello world!

Styles	Computed	Layout	Event Listener
Filter			
element.style {			
}			
div {			
color: ■ red;			
}			

Styles	Computed	Layout	Event Listeners	DOM Break
Filter				
▶ color				
■ rgb(255, 0, 0)				
▶ display				
block				
height				
21.3333px				
width				
1264px				