

29일 (목)

1. Codecademy - Building Interactive Websites - JavaScript Interactive Websites

Lesson 1. The Script Element

- JavaScript and HTML

js 는 웹을 동적으로 만들 수 있다. `<script>` 태그는 js 가 html 로 들어가게 하는 문과 같다.



- The `<script>` tag

```
<h1>This is an embedded JS  
example</h1>  
<script>  
    function Hello() {  
        alert ('Hello World');  
    }  
</script>
```

다음과 같이 html 에 직접 script 를 작성할 수 있다.

- The `src` attribute

`src` 와 `path` 를 사용하여 js 를 연결할 수 있다.

```
<script src="./exampleScript.js">  
</script>
```

- **How are scripts loaded?**

브라우저는 태그들을 렌더링 하는 것을 돕는 HTML parsers 를 장착하고 있다. html 파서가 <script> 태그를 만나게 되면 나머지 html 모든 태그를 파싱하기 전에 script 를 로드하고 내용을 실행시킨다.

여기서 알아야 할 포인트가 있다.

⇒ html 파서는 script 가 로드될 때까지 다음 태그로 넘어가지 않는다. 이럴 경우 나머지 html 파일을 읽어오는데 딜레이가 생기게 되어 유저에게 안좋은 경험을 선사하게 된다.

⇒ 추가적으로, 스크립트는 순차적으로 로드된다. 그래서 만약 스크립트가 다른 스크립트에 의존한다면, html 내에 script 태그의 순서를 신경써서 잘 놓아야한다.

- **Defer attribute**

script 태그의 defer 태그는 이러한 문제를 방지해준다.

html 파서가 defer 속성을 가진 script 태그를 만나게 되면 일단 script 를 로드 하지만, 나머지 html 의 태그를 모두 실행하고 나서 마지막에 스크립트를 실행하게 된다.

defer 는 DOM 과 함께 쓸 때 유용하다. 모든 html 을 읽고나서 script 를 실행하는 것이다.

```
7      <script defer id="blue"
      src="turnBlue.js"></script>
```

- **Async attribute**

async 속성은 나머지 웹페이지와 함께 스크립트를 비동기적으로 로드하고 실행시킨다. 이는 defer 속성과 비슷하다. 다운로드를 해놓고 html 이 모두 파싱 돼서야 script 를 실행한다. 그

러나 async 속성은 전체 페이지가 파싱될 때까지 기다리지 않는다. 즉, 다운로드 되자마자 script 가 실행된다.

```
<script src="example.js" async>  
</script>
```

async 는 언제 유용할까? 다른 스크립트와 독립적인 스크립트를 사용할 때 유용하다. 만약 스크립트 파일이 실행되는 지점에 정확히 문제가 없다면, 비동기적인 로딩은 웹페이지의 로딩 시간을 최적화 하기 때문에 가장 적합한 옵션이다.

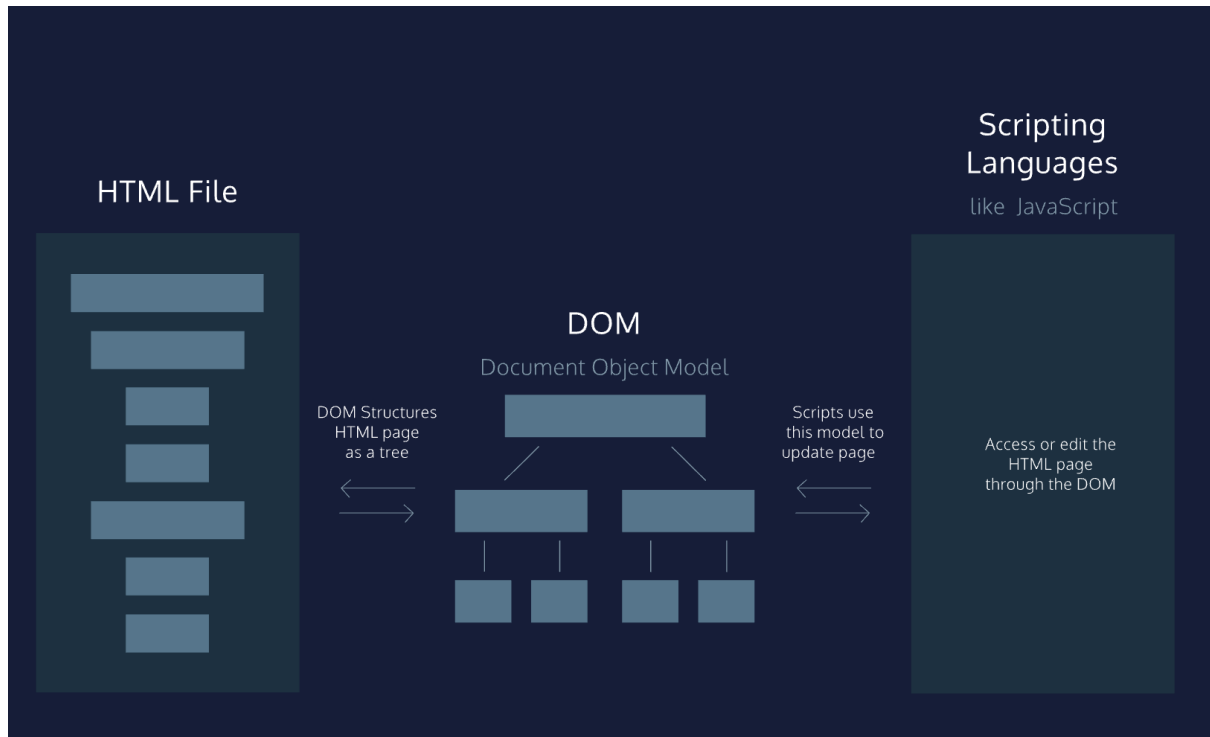
****** defer 과 async 의 차이점은 다운로드되고나서의 즉시 실행여부이다.**

과거에는 html 파싱 문제때문에 </body> 태그 바로 위에 script 태그를 놓았지만, 현대에는 async 나 defer 을 이용하여 head 태그 안에 입력한다.

2. Codecademy - Building Interactive Websites - The Document Object Model

Lesson 1. The Document Object Model (DOM)

- What is the DOM ?

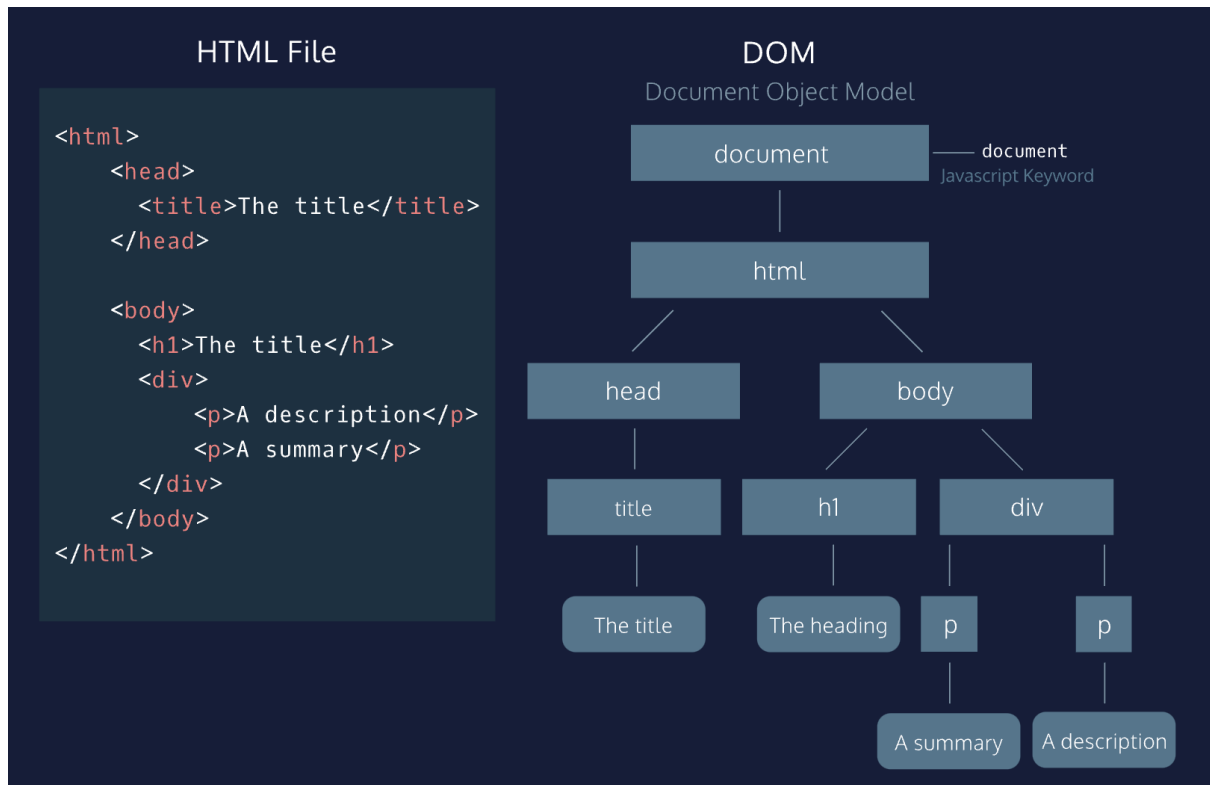


DOM 은 프로그래머들이 계층을 개념화하고, 웹페이지의 태그에 접근할 수 있게 하는 강력한 트리 자료구조이다.

Note: DOM 구조로 구성된 XML, SVG 와 같은 다른 타입의 document 들이 있다.

DOM 은 자바스크립트가 구조화된 html에 접근하여 구조를 수정하거나 , 업데이트 할 수 있게 한다. 이러한 이유로 위 그림과 같이 DOM 은 HTML 와 다른 여러 스크립트 언어의 오작교같은 역할을 한다고 볼 수 있다.

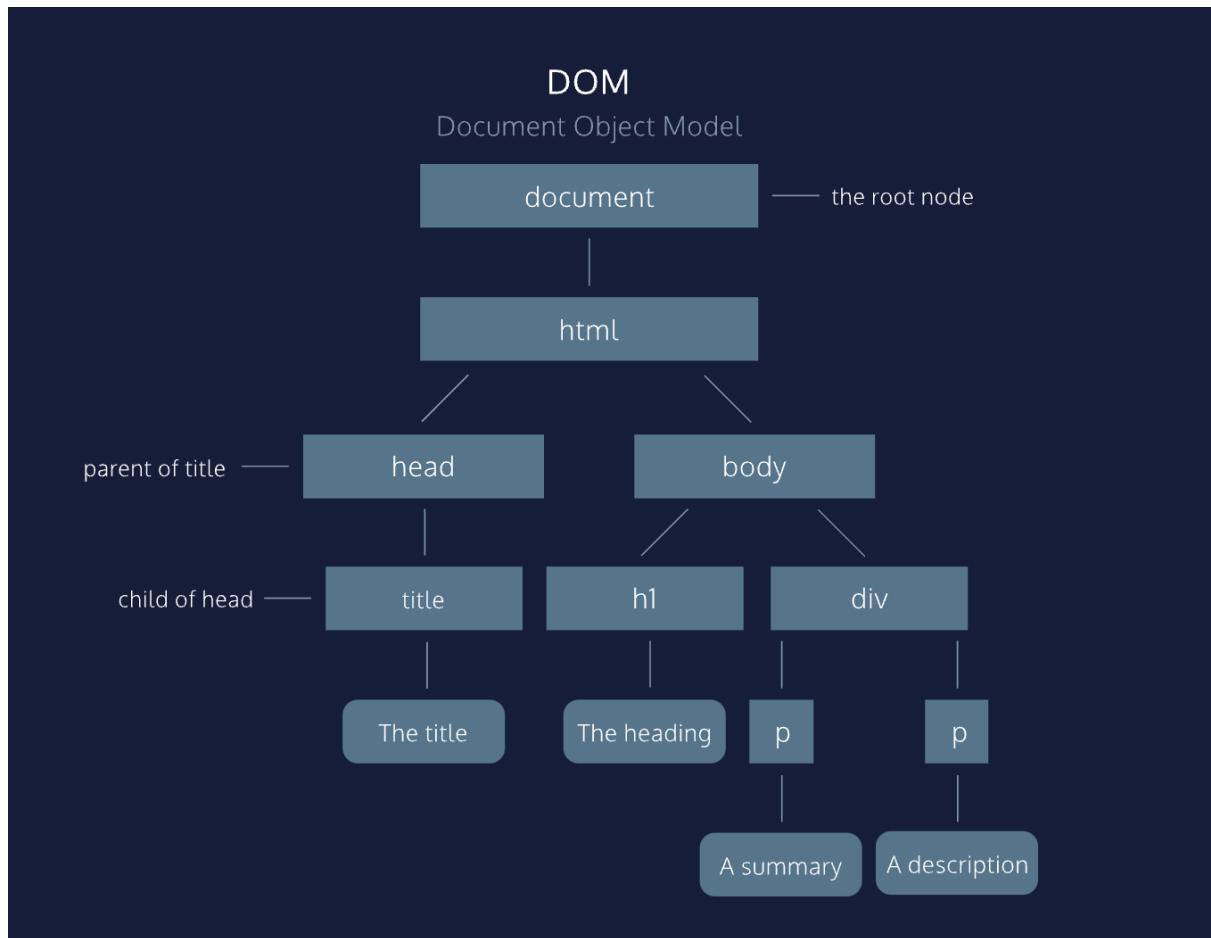
- The DOM as a Tree Structure



DOM 은 패밀리 트리 구조를 따른다.

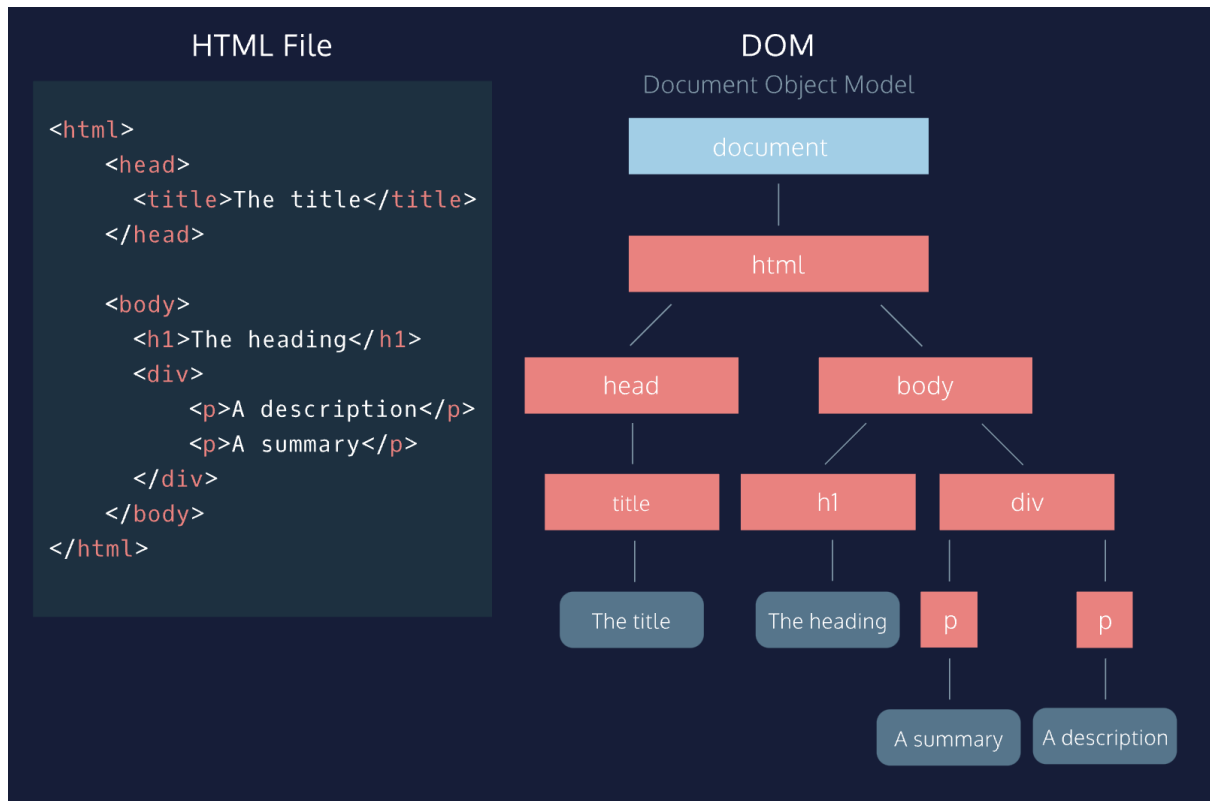
가장 위에 있는 노드는 root node 라고 부르며 보통 HTML 태그이다. 그 자식으로는 body, head 가 온다.

- Parent Child Relationships in the DOM



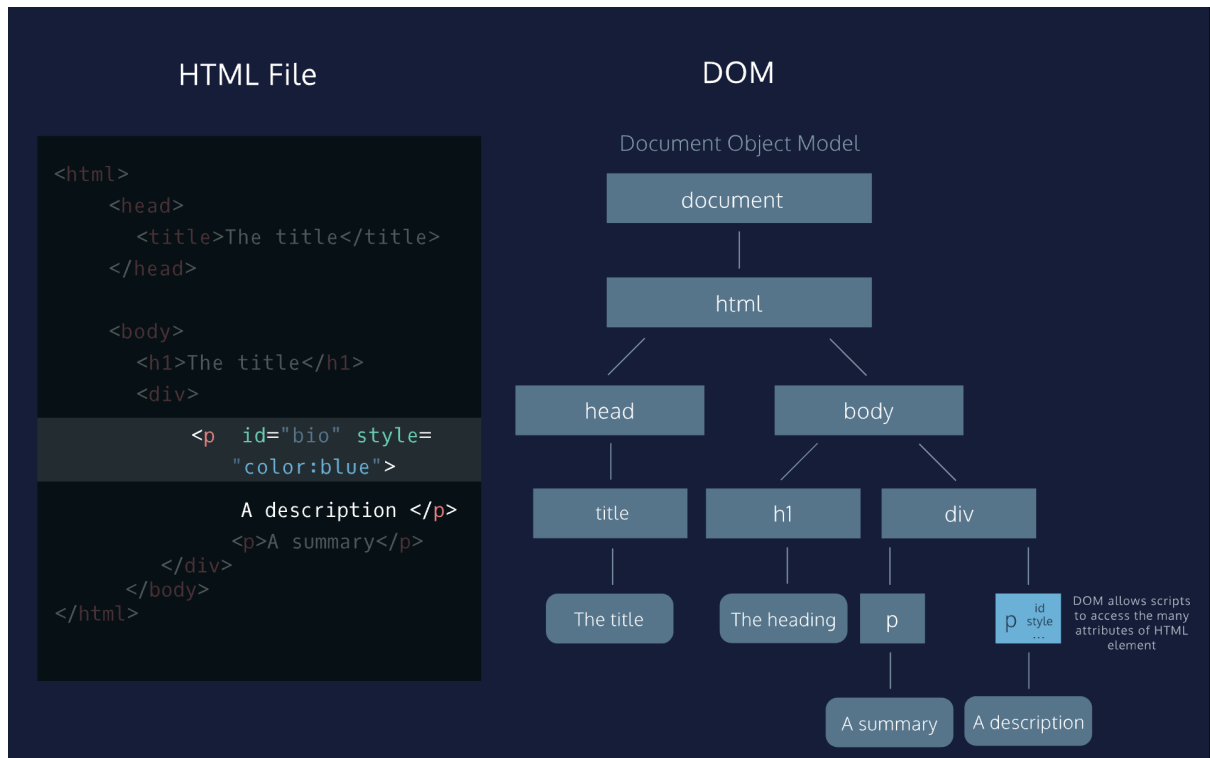
```
1  <!DOCTYPE html>
2  ▼ <html>
3  ▼  <head>
4      <title>The title
5      </title>
6  </head>
7
8  ▼  <body>
9      <h1>The heading</h1>
10
11 ▼    <div>
12        <p>A summary</p>
13        <p>A description</p>
14    </div>
15 </body>
16 </html>
17
```

- Nodes and Elements in the DOM



위 DOM 의 트리구조에서 빨간색은 태그들을 나타내며, 둥근 사각형은 텍스트를 나타낸다. 웹페이지를 수정하고자 할 때, 스크립트는 태그들과 거의 대부분 상호작용할 것이다.

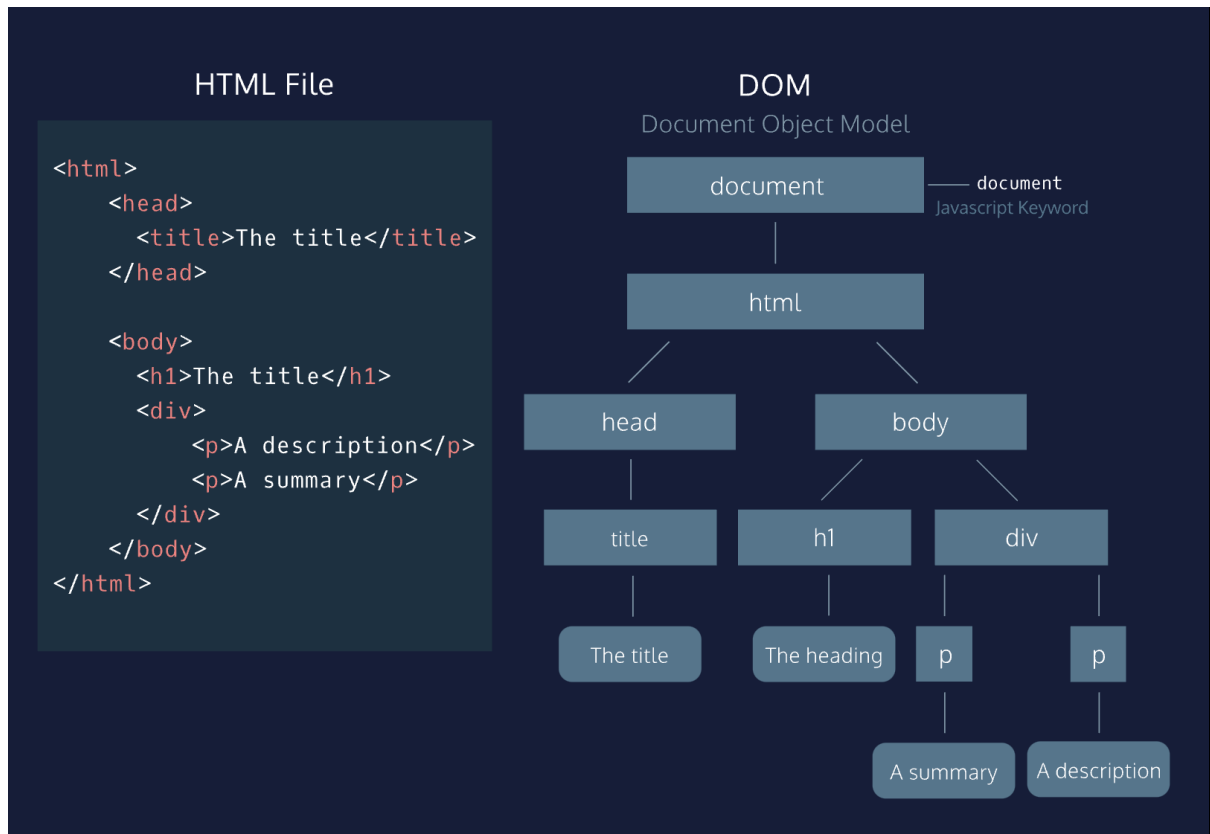
- Attributes of Element Node



DOM 또한 HTML 과 같이 id , class, inline style 속성에 접근할 수 있다.

Lesson 2. JavaScript and the DOM

- The document keyword



이번 레슨에서는 유용한 DOM Interface 의 프로퍼티와 메서드들에 대해 배워볼 것이다. 이 인터페이스는 모든 현대 브라우저에 의해 구현된다.

먼저 javascript의 **document** 오브젝트이다. 이 오브젝트는 DOM 구조의 root node로 들어가는 문과 같다.

만약 DOM 의 body 로 접근하기 위해서 프로퍼티인 document.body 를 입력하여서 접근하면 된다.

- Tweak an Element

또한 태그의 속성, 프로퍼티 ,p태그 내부의 내용을 변경할 수 있는 것부터해서 div 태그의 색을 변경할 수 있는 기능을 갖고 있다.

컨텐츠 내용을 변경하기 위해서는 .innerHTML 프로퍼티를 사용하면 된다.

```
document.body.innerHTML = 'The cat  
loves the dog.';
```

아래와 같이 콘텐츠 내용안에 태그를 추가할 수 있다. 즉, body 안에 자식태그인 h2 를 생성하는 것이다.

```
document.body.innerHTML = '<h2>This  
is a heading</h2>';
```

- Select and Modify Elements

DOM 인터페이스는 css 셀렉터로 특정한 태그에 접근할 수 있도록 한다.

.querySelector() 메서드는 특정 태그나 클래스, id 를 지정하여 연결할 수 있다.

이 메서드는 html 내부의 가장 첫번째 태그만 불러온다는 것을 주의하자.

```
document.querySelector('p');
```

html 태그 내부의 가장 첫번째 p 태그를 연결 id 나 클래스를 사용하려면 # 이나 . 을 붙여서 쓰면 된다.

아래와 같이 다른 옵션 또한 존재한다.

```
document.getElementById('bio').innerHTML = 'The description';
```

id 태그를 부르며 메서드 인자에 # 을 붙이지 않는다.

- Style an element

document 객체의 .style 프로퍼티를 사용하면 태그의 inline style 에 접근할 수 있다.

```
element.style.property
```

포맷은 위와 같다.

css 프로퍼티를 사용할 때는 '-' 를 사용하지 않고 카멜표기법을 사용한다.

```
let blueElement  
= document.querySelector('.blue');  
blueElement.style.backgroundColor  
= 'blue';
```

- Create and Insert Elements

html 수정이 가능한 것처럼 생성도 가능하다.

document 객체의 .createElement(tagname) 메서드를 이용하면 된다. 하지만 html 에 바로 연결되지 않으며, 텅빈 inner html 을 여전히 가지고 있다.

생성한 태그를 html에 연결하기 위해서는 .appendChild() 메서드를 사용해야한다. 이는 마지막 자식으로서 추가된다.

```
let paragraph
= document.createElement('p');

paragraph.id = 'info';

paragraph.innerHTML = 'The text
inside the paragraph';

document.body.appendChild(paragraph);
```

id 속성을 위와 같이 id 프로퍼티를 사용하여 지정할 수 있다.

```
1 let newDestination = document.  
  createElement('li');  
2  
3 newDestination.id = 'oaxaca';  
4  
5 newDestination.innerHTML = 'Oaxaca,  
  Mexico';  
6  
7 document.getElementById  
  ('more-destinations').appendChild  
  (newDestination);
```

먼저 태그를 생성한 뒤, id 속성을 넣고, 태그내 내용을 입력한 뒤 , 원하는 위치를 찾아서 자식태그로 append 한다.

- Remove an Element

.removeChild() 메서드를 사용하여 부모태그로 부터 특정한 자식태그를 제거할 수 있다.

오늘의 단어

- agnostic : 인지불능의, 불가지론의
- just as : 꼭 ~ 처럼