

K-Digital Training 웹 풀스택 과정

JavaScript (4)

Javascript

표준객체

Javascript 표준 객체

- 자바스크립트가 기본적으로 가지고 있는 객체들
- 프로그래밍을 하는데 기본적으로 필요한 도구들
- String, Number, Array, Date, Math

Date 객체

- Javascript 에서 매 순간 바뀌는 시간과 날짜에 관한 정보를 얻기 위해 사용하는 객체
- 초기화

```
new Date()  
new Date(밀리초)  
    -> new Date(8000000); // 1970년 1월 1일 0시부터 밀리초만큼 지난 날짜  
new Date(년, 월, 일, 시, 분, 초, 밀리초)  
    -> |new Date(16,5,25); // 1916년 5월 25일 00:0:00  
    -> new Date(2016,5,25,15,40) // 2016년 5월 25일 15:40:00
```

Date 객체 - 함수

- Date.now()
- Date.prototype getter 메소드 (var date = new Date();)
 - date.getFullYear()
 - date.getDate()
 - date.getDay()
 - date.getTime()
 - date.getHours()
 - date.getMinutes()
 - date.getSeconds()

Math 객체

- 수학에서 자주 사용하는 상수와 함수들을 미리 구현해 놓은 Javascript 표준 내장 객체
- 웹 브라우저마다 다른 결과를 얻을 가능성이 있기에 정확한 결과를 얻어야 할 경우에는 Math 메소드를 사용하지 않는 것이 좋다.

Math 객체 - 함수

- Math.PI
- Math.E

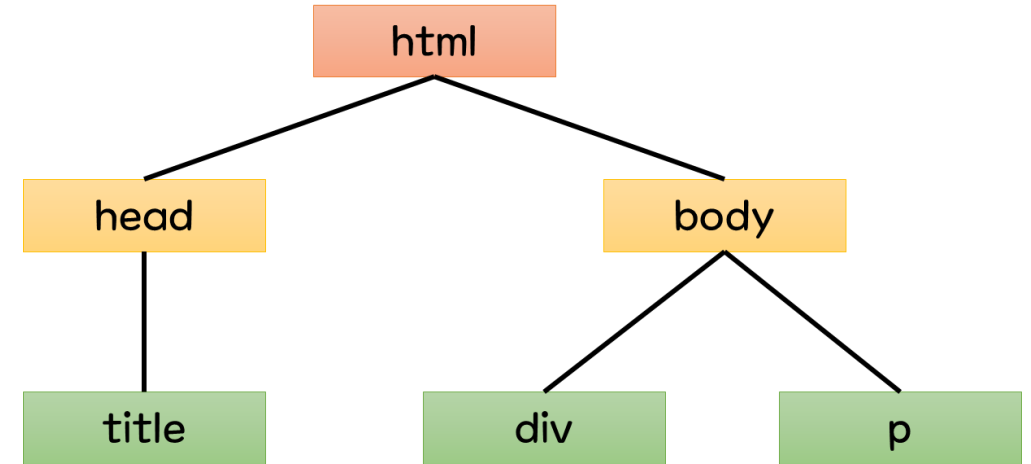
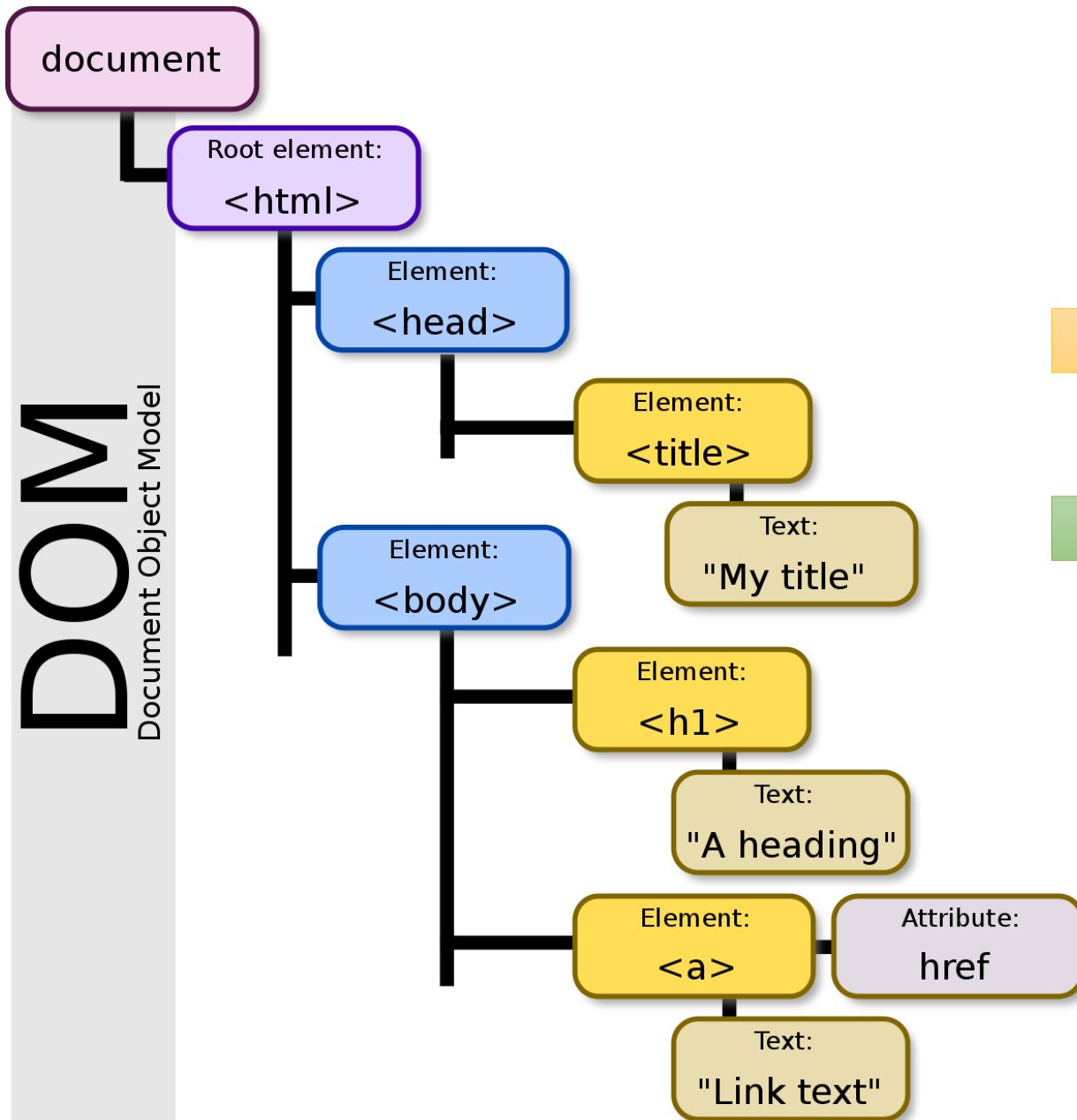
- Math.min()
- Math.max()
- Math.random()
- Math.round()
- Math.floor()
- Math.ceil()

DOM

(Document Object Model)

DOM(Document Object Model)

- HTML 문서 요소의 집합!
- HTML 문서는 각각의 node 와 object 의 집합으로 문서를 표현
- 따라서 각각 node 또는 object 에 접근하여 문서 구조 / 스타일 / 내용 등을 변경 할 수 있도록 하는 것!



Document

- 웹 페이지에 존재하는 HTML 요소에 접근하여 행동을 하고자 할 때 사용하는 객체

DOM

1. 새로운 HTML 요소나 속성 추가
2. 존재하는 HTML 요소나 속성 제거
3. HTML 문서의 모든 HTML 요소 변경
4. HTML 문서의 모든 HTML 속성 변경
5. HTML 문서의 모든 CSS 스타일 변경
6. HTML 문서에 새로운 HTML 이벤트 추가
7. HTML 문서의 모든 HTML 이벤트에 반응

DOM API

Document Object Model, Application Programming Interface

요소 선택

Document - 요소 선택

- document.getElementById(아이디 속성값)
 - document.getElementsByClassName(클래스 속성값)
 - document.getElementsByTagName(태그 이름)
 - document.getElementsByName(name 속성값)
-
- document.querySelector(CSS 선택자)
 - document.querySelectorAll(CSS 선택자)

querySelector(“요소 선택자”)

- 요소 선택자를 사용해서 자신이 가져오고 싶어하는 요소를 가져오는 메소드
- 문서에서 만나는 **제일 첫번째 요소**를 반환 합니다!

```
let boxEl = document.querySelector(".box");  
console.log(boxEl);
```


querySelectorAll(“요소 선택자”)

- 문서에 존재하는 모든 요소를 찾아주는 메소드
- 모든 요소를 가져와서 배열(같은) 데이터로 만들어 줍니다!

```
<body>
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
  <div class="box">7</div>
</body>
```

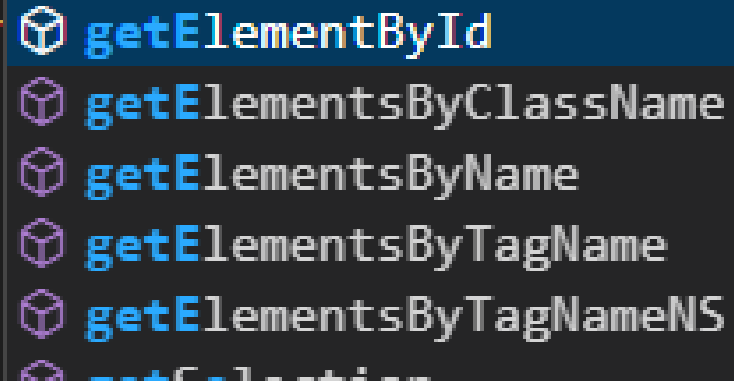
```
let boxEls = document.querySelectorAll(".box");
console.log(boxEls);
```

```
▼ NodeList(7) [div.box, div.box, div.box, div.box, div.box, div.box, div.box]
  ▶ 0: div.box
  ▶ 1: div.box
  ▶ 2: div.box
  ▶ 3: div.box
  ▶ 4: div.box
  ▶ 5: div.box
  ▶ 6: div.box
    length: 7
  ▶ [[Prototype]]: NodeList
```

getElementById(“ID이름”)

- 해당 ID를 가지는 요소를 불러오는 메소드

```
const inputEl = document.getElementById("input");
```



getElementById
getElementsByClassName
getElementsByName
getElementsByTagName
getElementsByTagNameNS
getSelection

요소 다루기

Document - 요소 다루기

- document.createElement(html요소)
- document.write(텍스트)
- [].appendChild();
- [].removeChild();
- [].append();
- [].remove();
- [].innerText = 내용;
- [].className = 클래스 이름;

.textContent .innerText .innerHTML

- 태그 내에 들어갈 문자열을 지정

요소.textContent="hi";

선택된 요소에 내부의 문자열로 hi가 들어가게 됩니다.

classList.~

- 선택 요소에 class 를 더하거나, 빼거나, 클래스가 존재하는지 체크하는 메소드
- 해당 기능과 CSS 를 잘 활용하면 액티브한 웹페이지 구성이 가능
 - 요소.classList.add()
 - 요소.classList.remove()
 - 요소.classList.contains()
 - 요소.classList.toggle()

setAttribute, html 요소 속성 추가

- 선택한 요소의 속성 값을 직접 지정할 수 있는 메소드
- 요소.setAttribute(“속성명”, “지정할 속성“)

```
searchInputEl.setAttribute("placeholder", "통합검색");
```

다른 노드에 접근하기

- 특정 노드를 선택한 후, 그 노드의 형제, 부모, 자식 노드에 접근하는 방법
 - 요소.children / 요소.children[0]
 - 요소.parentNode
 - 요소.previousElementSiblings
 - 요소.nextElementSiblings

createElement('html 요소')

- html 의 특정 노드를 생성
- 괄호안에는 html의 요소인 태그명을 넣어주시면 됩니다!

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

요소를 만들었으면 추가해야겠죠?

- 요소.append() /요소.appendChild()
 - 선택된 요소의 맨 뒤의 자식 요소로 추가됨
- 요소.prepend()
 - 선택된 요소의 맨 앞쪽인 자식 요소로 추가됨
- 요소.before()
 - 선택된 요소의 앞에 있는 형제 요소로 추가됨
- 요소.after()
 - 선택된 요소의 바로 뒤인 형제 요소로 추가됨

요소 삭제, remove() removeChild()

- 요소.remove();
 - 선택된 요소가 삭제 됩니다.
- 요소.removeChild('요소의 자식요소');
 - 선택된 요소의 자식 요소가 삭제 됩니다.

이벤트

`addEventListener`

요소.addEventListener(이벤트, 명령)

- 선택 요소에 지정한 이벤트가 발생하면, 약속 된 명령어를 실행시키는 메소드

```
let boxEl = document.querySelector(".box");  
  
console.log(boxEl);  
  
boxEl.addEventListener("click", function() {  
    alert("click!");  
})
```

```
document.querySelector(".box").addEventListener("click", function() {  
    alert("click");  
})
```

addEventListener 이벤트의 종류

- Click : 클릭
- Mouse 계열
 - Mouseover : 요소에 커서를 올렸을 때
 - Mouseout : 마우스가 요소를 벗어날 때
 - Mousedown : 마우스 버튼을 누르고 있는 상태
 - Mouseup : 마우스 버튼을 떼는 순간
- Focus : 포커스가 갔을 때
- Blur : 포커스가 벗어나는 순간

addEventListener 이벤트의 종류

- Key 계열
 - Keypress : 키를 누르는 순간 + 누르고 있는 동안 계속 발생
 - Keydown : 키를 누르는 순간에만 발생
 - Keyup : 키를 눌렀다가 떼는 순간
- Load : 웹페이지에 필요한 모든 파일(html, css, js 등)의 다운로드가 완료 되었을 때
- Resize : 브라우저 창의 크기가 변경 될 때
- Scroll : 스크롤이 발생할 때
- Unload : 링크를 타고 이동하거나, 브라우저를 닫을 때
- Change : 폼 필드의 상태가 변경 되었을 때