

BOM (Browser Object Model)

- 웹 브라우저를 통해 실행될 때 Javascript가 갖게되는 기본 객체 구조.
- 모든 객체는 window 객체의 하위 객체로서 존재한다.

HTML 태그를 객체로 가져오기

태그 이름으로 가져오기

특정 태그 모두를 가져오기 때문에 반환되는 객체는 항상 배열 형식이다.

반환되는 객체의 수가 너무 많기 때문에 잘 사용하지 않는다.

```
const 객체 = document.getElementsByTagName("태그이름")
```

ID값으로 가져오기

ID이름에 "#"을 붙이지 않는다.

ID값은 HTML 문서 안에서 고유한 요소이므로 항상 단일 객체로 반환된다. ID값이 여러개일 경우 첫번째것을 가져옴

```
const 객체 = document.getElementById("ID이름");
```

CSS 클래스 이름으로 가져오기

클래스 이름에 점을 붙이지 않는다.

특정 클래스가 적용된 모두를 가져오기 때문에 반환되는 객체는 배열 형식이다.

HTMLCollection이라고 불림

```
const 객체 = document.getElementsByClassName("CLASS이름");
```

★ Elements는 무조건 반복돌림

CSS 선택자로 단일 객체 가져오기

CSS 선택자의 형태에 상관 없이 단일 객체로 반환된다.

만약 동일한 선택터를 적용받는 요소가 두 개 이상인 경우 가장 첫 번째 요소만 반환한다.

```
const 객체 = document.querySelector("CSS선택자");
```

CSS 선택자로 복수 객체 가져오기

CSS 선택자의 형태에 상관 없이 복수 객체를 배열로 반환한다.

```
const 객체 = document.querySelectorAll("CSS선택자");
```

HTML 태그안에 내용 넣기

Javascript 가져온 HTML 객체는 innerHTML 이라는 속성값을 갖는다.

이 값에 할당한 내용은 실제 HTML의 시작태그와 끝 태그 사이에 적용된다.

```
const 객체 = document.querySelector("#hello");
객체.innerHTML = "Hello World";
```

```
<div id="hello"> Hello World </div>
      -----
      innerHTML
```

Javascript 이벤트

이벤트의 정의

프로그램이 겪는 어떠한 사건. ~때 → 시점을 중요시 생각해야 함

- 사용자의 클릭, 마우스 오버
- 페이지 로딩 직후, 페이지 종료 직전

자바스크립트가 제공하는 브라우저 관련 기능 안에 이미 다양한 이벤트를 감지하는 기능이 구현되어 내장 기능으로 개발자에게 제공된다.

이벤트 리스너

어떤 이벤트가 발생하기를 기다리고 있다가 이벤트 발생시 연결된 함수를 호출해 주는 기능.

이벤트를 감지하는 것은 Javascript 안에 이미 내장되어 있지만 어떤 이벤트가 발생했을 수행할 동작은 웹 페이지마다 다를 수 밖에 없으므로 미리 구현해 놓을 수 없다. (페이지마다 개별 구현이 필요함)

이벤트를 감지하는 기능이 수행할 동작을 콜백함수로 요구한다.

```
객체.addEventListener('이벤트이름', 콜백함수);
```

이벤트 핸들러

HTML 태그의 속성 형태로 존재하는, 이벤트 발생시 실행할 자바스크립트 코드 영역

Javascript가 고도화 됨에 따라 HTML과 Javascript 코드를 분리하는 경향이 생기면서 이벤트 핸들러는 거의 사용되지 않는 추세.

```
<태그이름 on이벤트이름="...JS코드영역..."></태그이름>
```

Javascript 이벤트 종류

마우스관련

자주 사용되는 이벤트만 이번 단원에서 정리

이벤트	이벤트 핸들러	설 명
click	OnClick	대상을 클릭했을 경우
dblclick	OnDbClick	대상을 더블클릭했을 경우
mousedown	OnMouseDown	마우스 버튼을 누르고 있는 동안
mouseup	OnMouseUp	마우스 버튼을 누르고 있다가 땀 경우
mousemove	OnMouseMove	마우스를 움직였을 경우
mouseout	OnMouseOut	대상에서 마우스 포인터가 벗어났을 경우
mouseover	OnMouseOver	대상에 마우스 포인터가 위치했을 경우
dragdrop	OnDragDrop	대상을 클릭한 상태에서 이동했을 경우

키보드 관련

자주 사용되는 이벤트만 이번 단원에서 정리

이벤트	이벤트 핸들러	설 명
keydown	OnKeyDown	키가 눌러져 있는 동안 반복 실행된다.
keyup	OnKeyUp	키를 눌렀다가 놓았을 경우
keypress	OnKeyPress	화면에 출력되는 키가 눌러질 경우 (한글동작안함)

form 요소 관련

21단원-Form요소에서 정리함

이벤트	이벤트 핸들러	설 명
submit	OnSumit	입력 양식을 서버로 보냈을 경우 (submit버튼을 누른 경우)
change	OnChange	대상에 입력되어 있는 값이 바뀌었을 경우. 입력상자,체크박스,라디오,드롭다운에 적용
blur	OnBlur	대상에서 포커스가 빠져나간 경우
focus	OnFocus	대상에 포커스가 들어 왔을 경우
reset	OnReset	대상을 재시작 시켰을 경우(주로 form)

이벤트	이벤트 핸들러	설 명
select	OnSelect	입력 양식의 한 필드를 선택했을 경우

브라우저 관련

자주 사용되는 이벤트만 이번 단원에서 정리

이벤트	이벤트 핸들러	설 명
abort	OnAbort	이미지를 읽다가 중단했을 경우
error	OnError	에러가 발생했을 경우
load	OnLoad	대상을 열었을 경우 (주로 페이지 로딩 직후를 의미)
move	OnMove	윈도우나 프레임을 움직였을 경우
resize	OnResize	윈도우나 프레임의 크기가 변경했을 경우
unload	OnUnload	대상을 종료했을 경우

트랜지션 관련

이벤트	이벤트 핸들러	설 명
transitionrun	OnTransitionRun	트랜지션이 동작하는 동안
transitionstart	OnTransitionStart	트랜지션이 시작된 경우
transitioncancel	OnTransitionCancel	트랜지션이 취소된 경우
transitionend	OnTransitionEnd	트랜지션이 종료된 경우