KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

문서 객체 모델

[KB] IT's Your Life

KB 국민은행

1 문서 객체 모델의 기본 용어와 개념

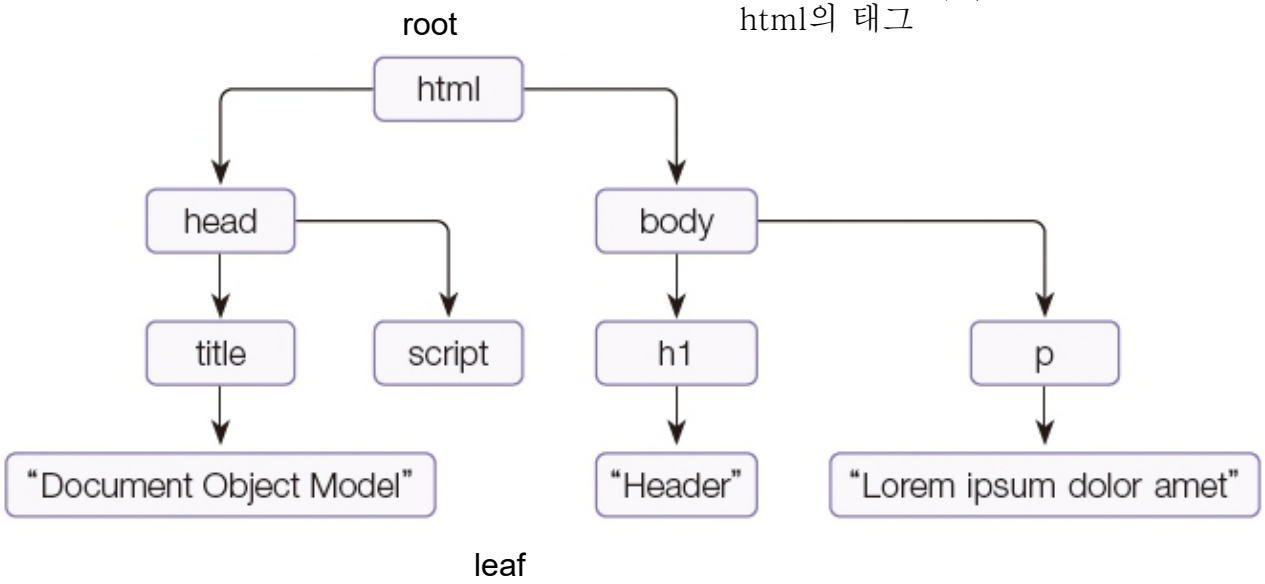
문서 객체 모델 기본 용어

- 문서 객체(Document Object)
  - HTML 태그를 자바스크립트에서 사용할 수 있는 객체로 만든 것.

- HTML 문서의 요소는 부모-자식 관계를 가짐 --> DOM Tree로 표현
  - 요소노드와 텍스트 노드

```
<!DOCTYPE html>
<html>
<head>
      <title>Document Object Model</title>
      <script>

      </script>
</head>
<body>
      <h1>Header</h1>
      <p>Lorem ipsum dolor amet</p>
</body>
</html>
```

DOM TREE에서 NODE는 html의 태그



1 문서 객체 모델의 기본 용어와 개념

문서 객체 모델 기본 용어

- 텍스트 노드를 갖지 않는 태그

```
<br>
<hr>
<img src="rintiantta.png">
```

- 정적 생성: 웹페이지를 처음 실행할 때 HTML 태그로 적힌 문서 객체를 생성.
- 동적 생성: 페이지를 실행 중에 자바스크립트를 사용해 문서 객체를 생성. js가 동적으로 필요에 따라 태그를 생성함
- 객체 모델: 웹 브라우저가 HTML 파일을 분석하고 출력하는 방식

dom_fault.html 실행 순서에 따른 문서 객체 사용 오류

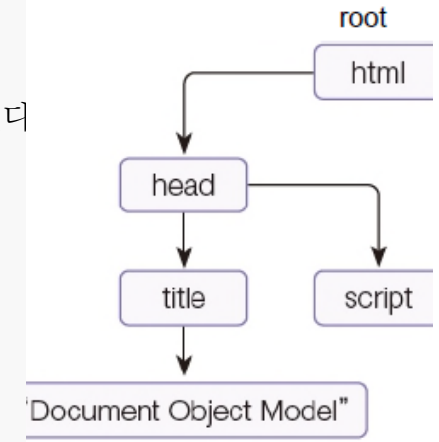
웹브가 html를 dom분석
위에서 아래로 순서대로 분석하고 실행

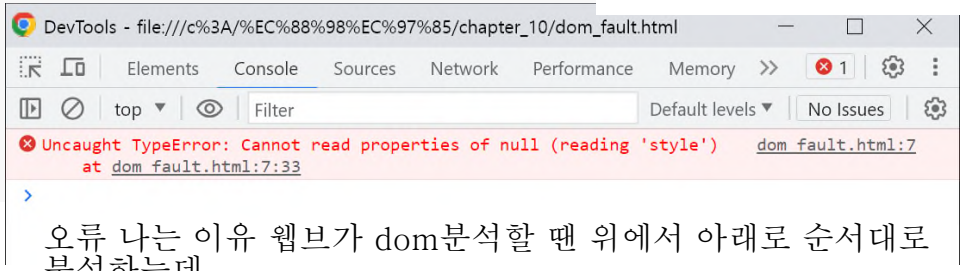
```
<!DOCTYPE html>
<html>
<head>
    <title>Document Object Model</title>
    <script>
        // h1 태그의 배경 색상을 변경합니다.
        document.querySelector('h1').style.backgroundColor = 'red';
        // h2 태그의 글자 색상을 변경합니다.
        document.querySelector('h2').style.color = 'red';
    </script>
</head>
<body>
    <h1>Process - 1</h1>
    <h2>Process - 2</h2>
</body>
</html>
```

css 선택자 : h1 노드들을 찾아서 변하를 가하겠다

h1태그 선택

이 시점
만들어진 dom tree 현황





오류 나는 이유 웹브가 dom분석할 땐 위에서 아래로 순서대로 분석하는데

body에 있는 h1을 읽기전에
head의 script 태그 안에서 body의 h1 에 접근하려 하니까

그럼 어덯게 해야돼?

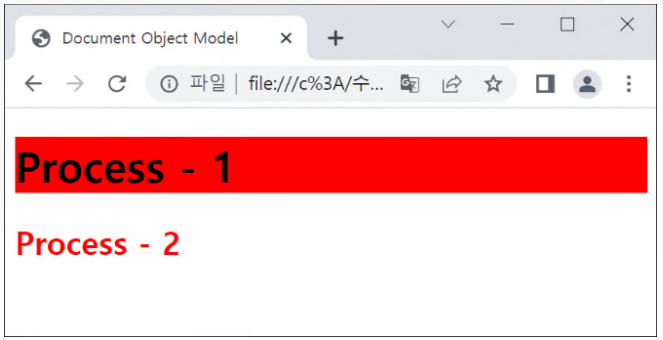
DOM TREE가 현 상황을 잘 반영하게 끔
다 만들어진 다음에 script로 접근하는 방법이 있다

dom_correct.html 해결방법1: script 태그 위치 옮기기

```
<!DOCTYPE html>
<html>
<head>
    <title>Document Object Model</title>
</head>
<body>
    <h1>Process - 1</h1>
    <h2>Process - 2</h2>

    <script>
        // h1 태그의 배경 색상을 변경합니다.
        document.querySelector('h1').style.backgroundColor = 'red';
        // h2 태그의 글자 색상을 변경합니다.
        document.querySelector('h2').style.color = 'red';
    </script>
</body>
</html>
```

DOM TREE에 접근하는
요소들이 로드 되어 있으니
잘 적요됨

하지만 더 간편한 방법이
없을까?



dom_event.html 해결 방법 2: 이벤트 활용하기

시스템에서는
window.onload(event객체)로 호출함

```
<!DOCTYPE html>
<html>
<head>
    <title>Document Object Model</title>
    <script>    제공되는 window전역객체의 onload이벤트 핸들러
        window.onload = function () {
            // h1 태그의 배경 색상을 변경합니다.
            document.querySelector('h1').style.backgroundColor = 'red';
            // h2 태그의 글자 색상을 변경합니다.
            document.querySelector('h2').style.color = 'red';
        };
    </script>
</head>
<body>
    <h1>Process - 1</h1>
    <h2>Process - 2</h2>
</body>
</html>
```

돔 트리에 로드 됐을 때 반영 되게끔 하는 함수를
지원해주는 이벤트 핸들러에 등록 시키기

window전역객체의 onload이벤트 핸들러는
html을 다 분석하여 dom tree를 완성시킨 이후에 실행된다.

해당 이벤트 핸들러 내용은
돔트리에 있는 노드들에 대한 제어가 들어가 있다.

하지만 더 많이 쓰이는 것은 js를 분리하여 <script defer src="..." > </script> 를 많이 사용한다

분리된 js 내용에는 이것만

2 문서 객체 선택

문서 객체 선택

- HTML 태그를 자바스크립트에서 문서 객체로 변환
- 문서 객체를 선택하는 메서드

완성된 돔트리에서
요소들을 제어하기 위해서는
우선 특정 원하는 요소들을 식별하고
골라야 한다.

구분	메서드	설명
1개 선택	document.getElementById(아이디)	아이디로 1개 선택
	document.querySelector(선택자)	선택자로 1개 선택
여러 개 선택	document.getElementsByName(이름)	name 속성으로 여러 개 선택
	document.getElementsByClassName(클래스)	class 속성으로 여러 개 선택
	document.querySelectorAll(선택자)	선택자로 여러 개 선택

아이디 이름

1개만 특정하는 메소드

만약 돔트리 노드들 중의 아이디들이 중복되면 첫 아이디의 노드만 선택됨

css의 선택자 그대로를 매개변수로.
단 발견된 태그가 여러개면 그 중 첫번째만 반환

태그의 이름을 통해서

여러개 반환 형식은?

배열과 비슷한 형식.

클래스의 이름을 통해서

css 선택자

select_id.html　　getElementById( ) 메서드를 사용해 문서 객체 1개 선택하기

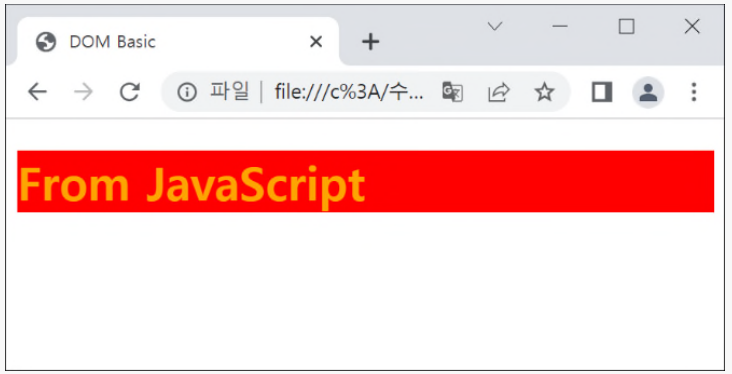
```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            let header = document.getElementById('header');
            // 문서 객체를 조작합니다.
            header.style.color = 'orange';
            header.style.background = 'red';
            header.innerHTML = 'From JavaScript';
        };
    </script>
</head>
<body>
    <h1 id="header">Header</h1>
</body>
</html>
```

붙이지 말고 아이디 내용만

항상 반환값이 null인지 확인해야함.

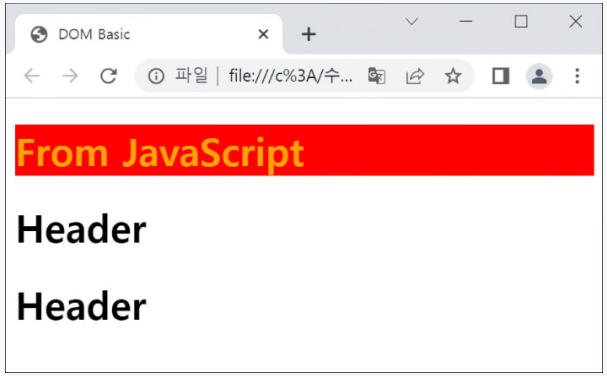
style뿐만 아니라

내용 제어도 가능



select_query.html querySelector( ) 메서드를 사용해 문서 객체 1개 선택하기

```
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.      단수 선택 메소드
            let header = document.querySelector('h1');

            // 문서 객체를 조작합니다.
            header.style.color = 'orange';
            header.style.background = 'red';
            header.innerHTML = 'From JavaScript';
        };
    </script>
</head>
<body>                          여러개의 해당된다면 첫번째 것만
    <h1>Header</h1>
    <h1>Header</h1>
    <h1>Header</h1>
</body>
```



select_all.html querySelectorAll( ) 메서드를 사용해 문서 객체 여러 개 선택하기

```
<script>
    // 이벤트를 연결합니다.
    window.onload = function () {
        // 문서 객체를 선택합니다.
        let headers = document.querySelectorAll('h1');

        for (let i = 0; i < headers.length; i++) {
            // 변수를 선언합니다.
            let header = headers[i];
            // 문서 객체를 조작합니다.
            header.style.color = 'orange';
            header.style.background = 'red';
            header.innerHTML = 'From JavaScript';
        }
    };
</script>
</head>
<body>
    <h1>Header</h1>
    <h1>Header</h1>
    <h1>Header</h1>
</body>
```

해당 반환객체는 키가 숫자 인덱스문자열. + 특정 속성들 + 메소드들
{ '0':값, '1':값, ... 'length':3 ...,}
기본적인 배열이 가지는 몇몇메소드를 가지고 있지 않다.

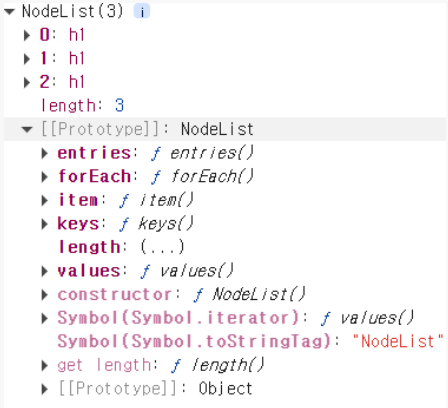
모든 일치 요소들

배열과 비슷한 형식의 객체로 반환받음. 해당 객체를 유사배열객체라고함.

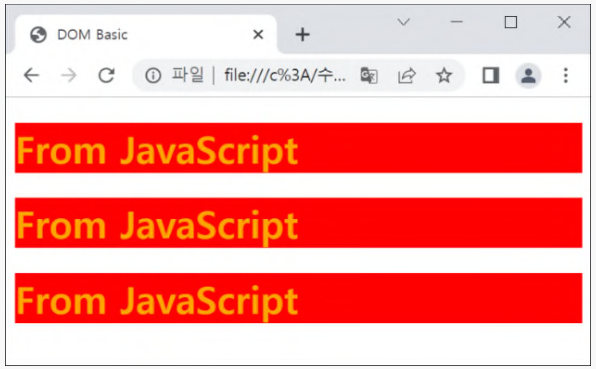
순회하며 접근

많은 것들 중 하나

해당 유사 배열 객체는 NodeList라는 객체다.
인덱스 키 : 값
length 속성
+
메소드들
이 있다



해당 NodeList 의 요소에는
html 노드가 있는데
각 노드도
수많은 속성과
메소드(이벤트 핸들러)
가 있다.



앞에서 말한
각 노드의
속성과 메소드들

```
scrollHeight: 45
scrollLeft: 0
scrollTop: 0
scrollWidth: 151
shadowRoot: null
slot: ""
spellcheck: true
▶style: CSSStyleDeclaration {0: 'cc
tabIndex: -1
tagName: "H1"
textContent: "heellllooo"
title: ""
translate: true
virtualKeyboardPolicy: ""
```

3 문서 객체 조작

글자 조작

글자 속성

속성	설명
textContent	문서 객체 내부 글자를 순수 텍스트 형식으로 가져오도록 변경
innerHTML	문서 객체 내부 글자의 HTML 태그를 반영해 가져오도록 변경

textContent를 통해서 문자열을 대입하면 문자열 그대로

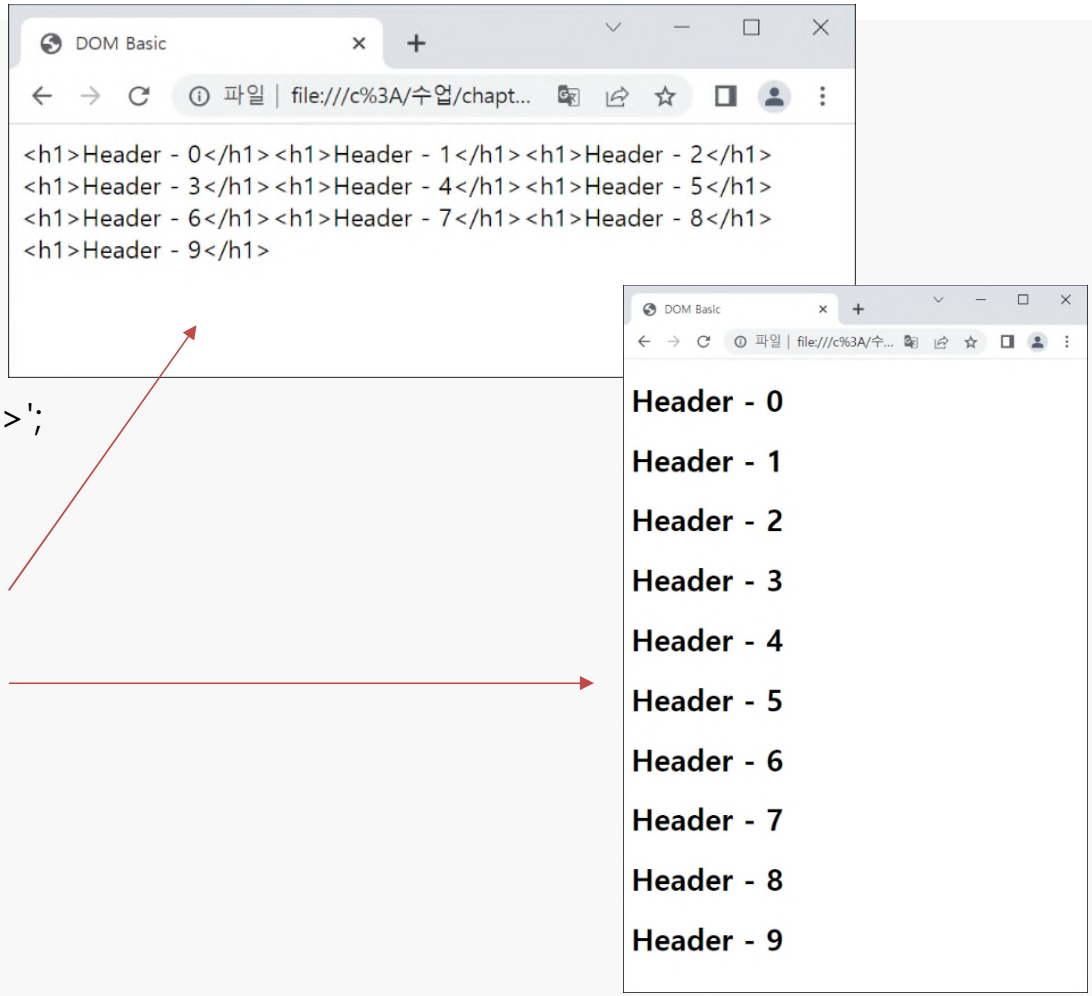
innerHTML을 통해서 문자열을 대입하면 해당 문자열을 html의 내용이 되는 것.
해당 내용이 태그라면 html의 태그 내용이 되는 것이기에 태그로써 적용함.

정리
textContent => '<h1> dfsdfasd </h1>'
innerHTML => <h1> afdasfasd </h1> => 진짜로 html 에 적용됨.

innerHTML => <script> ... </script> 를 대입하면 진짜 script태그로 작동함

control_text.html 내부 글자 변경

```
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 변수를 선언합니다.
            let output = '';
            for (let i = 0; i < 10; i++) {
                output += '<h1>Header - ' + i + '</h1>';
            }

            // 문서 객체 내부의 글자를 변경합니다.
            document.body.textContent = output;
            // innerHTML 속성을 사용합니다.
            // document.body.innerHTML = output;
        };
    </script>
</head>
<body>

</body>
```



스타일 조작

자바스크립트로 CSS 속성 값을 추가•제거•변경

```
① let header = document.getElementById('header');
  header.style.backgroundColor-color = 'red';

② let header = document.getElementById('header');
  header.style.backgroundColor = 'red';
```

스타일시트의 스타일 속성	자바스크립트의 스타일 식별자
background-image	backgroundImage
background-color	backgroundColor
box-sizing	boxSizing
list-style	listStyle

control_style.html 스타일 조작

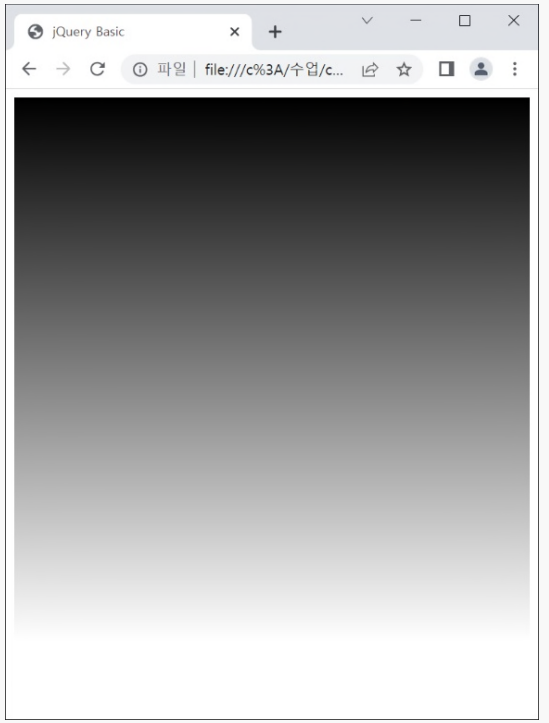
```
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 추가합니다.
            let output = '';
            for (let i = 0; i < 256; i++) {
                output += '<div></div>';
            }
            document.body.innerHTML = output;

            // 문서 객체를 선택합니다.
            let divs = document.querySelectorAll('div');
            for (let i = 0; i < divs.length; i++) {
                // 변수를 선언합니다.
                let div = divs[i];

                // 스타일을 적용합니다.
                div.style.height = '2px';
                div.style.background = 'rgb(' + i + ',' + i + ',' + i + ')';
            }

        };
    </script>
</head>
<body></body>
```

태그 관련 정보의 문자열 만들기

태그로 해석을 하는 innerHTML에 대입

같은 이벤트 핸들러에서 추가된게 바로 반영이 되나?

문자열로 줘야함!!!!



속성 조작

문서 객체의 속성 조작 메서드

메서드	설명
setAttribute(속성 이름, 속성 값)	속성 지정
getAttribute(속성 이름)	속성 추출

표준 정으된 속성들은 그냥 문서객체.표준속성=값; 하면되지만.
img,src='afasd.png';

표준 속성이 아닌 경우
속성 조작 메서드를 사용함.
imgObj.setAttribute('data-kkh','12321d');

- 웹 브라우저 제조사가 구현하지 않았다면 접근하지 못할 수 있음.
- 'data-'로 시작하는 사용자 지정 속성은 반드시 속성 조작 메서드로 접근.

```
<body>
    <div data-role="page">
        <div data-role="header"></div>
        <div data-role="content"></div>
        </div>
        <div data-role="footer"></div>
    </div>
</body>
```

사용자 정의 속성은 data-로 시작하는게 관례.

돔 트리에서
한 노드의 사용자 지정 속성들은
dataset 속성에 모아놓아져 있다

control_attribute.html　　img 태그 속성 조작하기

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 변수를 선언합니다.
            let image = document.getElementById('image');

            // 속성을 변경합니다.
            image.src = 'http://via.placeholder.com/300x200';
            image.width = 300;
            image.height = 200;
        };
    </script>
</head>
<body>
    <img id="image">
</body>
</html>
```

먼저 해당 이미지를 식별하여 선택.

표준 속성들을 제어하므로
imgObj.표준속성=...; 형식으로 접근

정적 생성으로는 너비와 높이가 배정 안되어
정상적으로 나오지 않음

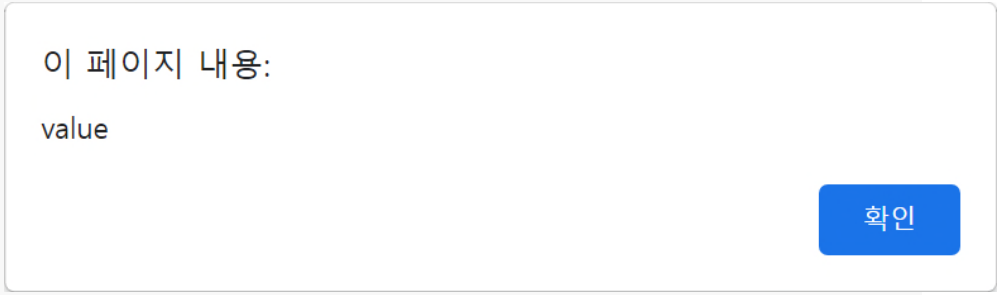
control_body.html body 태그 속성 조작하기

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 속성을 지정합니다.
            document.body.setAttribute('data-custom', 'value');

            // 속성을 추출합니다.
            let dataCustom = document.body.getAttribute('data-custom');
            alert(dataCustom);
        };
    </script>
</head>
<body>

</body>
</html>
```

body태그 노드는
식별과정 필요 없이 바로 접근 가능.

body태그 노드의 data-custom 속성의 값을 제어

추출



dom_clock.html 문서 객체 속성 조작

```
<html>
<head>
    <title>Clock</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            let clock = document.getElementById('clock');

            // 1초마다 함수를 실행합니다.
            setInterval(function () {
                let now = new Date();
                clock.innerHTML = now.toString();
            }, 1000);
        };
    </script>
</head>
<body>
    <h1 id="clock"></h1>
</body>
</html>
```

clock ID 태그 노드를 식별 선택.

동적으로 내용을 바꾸는 코드

1000ms

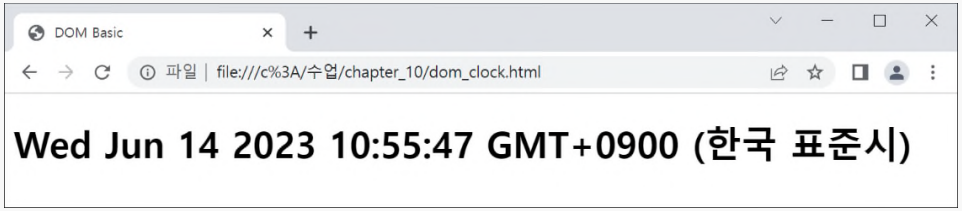
객체 내용을 문자열화 시키는 toString() 메소드

초 단위로 등록함수를 반복 실행하는 setInterval 이벤트 핸들러(f,시간);

초단위로 등록함수를 한번 실행하는 setTimeout이벤트 핸들러(f,시간);

반환형식은 timer_id라는 각 인벤트 핸들러 관련 타이머 식별자.

각 이벤트 핸들럴르 멈추고 싶으면 clearInterval(해당timer_id); clearTimeout(해당timer_id);



언제 처리하는 것에 대한 제어
=>
이벤트

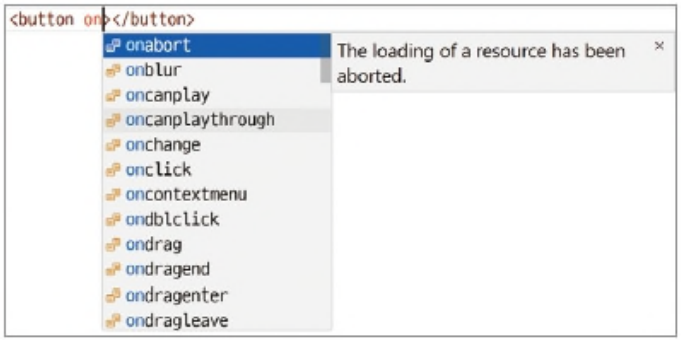
언제 실행할지를 제어.
키보드를 누를 때 실행시켜줘, 클릭할 때 실행시켜줘.
미리 이벤트 핸들러에 등록시켜야 함.

이벤트

- 키보드 누르기, 마우스 클릭 등 어떤 현상이 프로그램에 영향을 미침.

```
window.onload = function () { };
```

```
onblur  onfocus  onfocusin  onfocusout
onload  onresize  onscroll  onunload
onclick  ondbclick  onmousedown
onmouseup  onmousemove  onmouseover
onmouseout  onmouseenter
onmouseleave  onchange  onselect
onsubmit  onkeydown  onkeypress
onkeyup  onerror
```



이벤트 연결

- 이벤트 모델: 문서 객체에 이벤트를 연결하는 방식.

2가지 방법 있다.

구분	종류
DOM 레벨 0	• 인라인 이벤트 모델 • 고전 이벤트 모델
DOM 레벨 2	• 마이크로소프트 인터넷 익스플로러 이벤트 모델 • 표준 이벤트 모델

태그 안에다가 어떤 이벤트인지 js코드를 넣는 방법

이벤트 연결

인라인 이벤트 모델

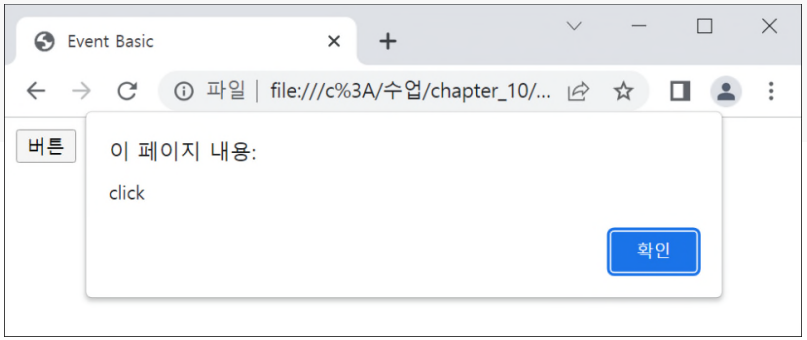
- HTML 태그 내부에 자바스크립트 코드를 넣어 이벤트를 연결하는 방식.

```
<태그 on이벤트="자바스크립트코드">...</태그>
```

주로 함수 호출문을 지정

event_inline.html 인라인 이벤트 모델 사용하기

```
<!DOCTYPE html>
<html>
<head>
    <title>Event Basic</title>
</head>
<body>
    <button onclick="alert('click')">버튼</button>
</body>
</html>
```



event_inlineWithScript.html script 태그에 인라인 이벤트 모델 사용하기

```
<!DOCTYPE html>
<html>
<head>
    <title>Inline Event</title>
    <script>
        function buttonClick() {
            alert('click');
        }
    </script>
</head>
<body>
    <button onclick="buttonClick()">버튼</button>
</body>
</html>
```

번외
<a href="javascript : js 코드 "> ...가능

뒤로가기 할때 많이 사용됨

근데 가능하면 인라인 방법은 비추

event_tradition.html 고전 이벤트 모델

```
<!DOCTYPE html>
<html>
<head>
    <title>Traditional Event</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            let button = document.getElementById('button');

            // 이벤트를 연결합니다.
            button.onclick = function () {
                alert('click');
            };
        };
    </script>
</head>
<body>
    <button id="button">버튼</button>
</body>
</html>
```

버튼을 식별 선택하고
해당 식별 노드의 이벤트 핸들러에 등록.

하지만 이 고전 방법에 단점.
보통 이벤트 핸들러는 여러개의 콜백 함수를 등록 할 수 있지만

이방법은 하나만 등록할 수 있다

표준 이벤트 등록 방법(dom2)를 사용해라.

타겟태그노드객체.addEventListener(발동조건, 이벤트 콜백함수);

imgObj.addEventListener('click', f() {});

addEventHandler함수의 발동조건 매개변수에는
'keyup' 'keydown' 'change' 등 많다.
다양하고 태그마다 다르니
필요할 때마다 인터넷 검색 ㄱㄱ

이벤트 사용

- 모든 이벤트 핸들러에는 이벤트 정보를 담고 있는 이벤트 객체가 매개변수로 잔달됨
- 이벤트 객체를 사용하면 이벤트와 관련한 정보를 알아낼 수 있음.

event_object.html 이벤트 정보

```
<!DOCTYPE html>
<html>
<head>
    <title>Event Object</title>
    <script>
        window.onload = function (event) {
            alert(event);
            console.log(event);
        };
    </script>
</head>
<body>

</body>
</html>
```

앞서 말했듯이 window.onload(event객체); 형식으로 호출함.

모든 이벤트 핸들러의 첫번째 매개변수는 event객체 이다.

event객체

```
▼ Event i
    isTrusted: true
    bubbles: false
    cancelBubble: false
    cancelable: false
    composed: false
  ▶ currentTarget: Window {window: Window, self: Window, docume
    defaultPrevented: false
    eventPhase: 2
    returnValue: true
  ▶ srcElement: document
  ▶ target: document
    timeStamp: 29.899999998509884
    type: "load"
  ▶ [[Prototype]]: Event
```

현재 이벤트 타겟

이벤트 사용

기본 이벤트:

- 특정 태그가 가진 기본적인 이벤트.

대표적 예

- <a>의 기본 이벤트
  - href에 지정한 페이지로 이동
  - 이벤트 핸들러에서 false를 리턴하면 기본 이벤트 핸들러 호출되지 않음

form 태그의 기본 이벤트 : 클릭시 서버에 전송

가끔
form과 a의 기본 이벤트를 의도와 다르게 사용하고 싶을 때가 있다.

event_defaulthtml.html 기본 이벤트 제거

```
<!DOCTYPE html>
<html>
<head>
    <title>Traditional Event - Default Event</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            let button = document.getElementById('button');

            // 이벤트를 연결합니다.
            button.onclick = function () {
                // 기본 이벤트를 제거합니다.
                return false;
            };
        };
    </script>
</head>
<body>
    <a id="button" href="http://hanbit.co.kr">버튼</a>
</body>
</html>
```

가끔
form과 a의 기본 이벤트를 의도와 다르게 사용하고 싶을 때가 있다.

false반환시
기본 액션 실행 안됨.

반환 전까지는 본문은 실행함.

if문으로 조건 분기하여 특정 조건에 기본 이벤트 실행 유무를
결정할 수 잇다

번외 이벤트 핸들러 함수 안에서 this 키워드는 어떻게 해석되느냐.
이벤트 핸들러가 어떻게 정의되었는냐에 따라 달라진다. 나중에 ㄱㄱ

본문은 실행되고
반환이 false면

기본 액션 실행 안됨

혹은 event매개변수를 주고, event.defaultPrevented();해도됨

활동되는 거에 비해서 코드가 복잡해지고 량이 많고 관리가 힘듬.
=> 좀더 편하게 해주는 프레임워크 사용. => jquery, angular, react, vue
프레임워크를 사용하지 않는 기존 방법은 이벤트 하나하나에 일일이 등록하는 느낌
=>
프레임워크를 사용하면 데이터에 따른 자동적으로 이벤트를 매핑해주는 느김
후자가 훨씬 생산성 높다