LIKE LION UNIV.
×
SANGMYUNG UNIV.

프론트엔드 6st 세미나

JavaScript

발표자 : 박서연

LIKE LION UNIV.
×
SANGMYUNG UNIV.

목 차

JS 기초 03

Document Object 12

querySelector 14

Form 17

Event 20

실습 26

HTTP method 27

객체 기반의 스크립트 프로그래밍 언어

1. 변수에 값 저장
2. 배열, 객체 등의 자료구조 존재
3. 웹 페이지에서 발생하는 특정 이벤트에 대한 응답
4. Application Programming Interface(API) 사용

JS 기초

변수

데이터를 저장할 수 있는 메모리 공간
숫자, 문자 등 저장 가능

```
let message = "Hello";
var sum = 0;
const name = "박서연";
```

	스코프 (유효 범위)	재선언	재할당
var	함수 레벨	가능	가능
let	블록 레벨	불가능	가능
const	블록 레벨	불가능	불가능 (객체는 가능)

주의 필요

배열

연관된 데이터를 관리하기 위한 데이터 타입

```
const fruits = ["배", "사과", "바나나"];
console.log(fruits.length);
console.log(fruits[1]);
fruits.push("체리");
console.log(fruits);
```

배열 접근

```
for (let i = 0; i < fruits.length; i++) {
    console.log(fruits[i]);
}
```

1. index로 접근

```
for (let fruit of fruits) {
    console.log(fruit);
}
```

2. for...of로 접근

```
fruits.forEach((fruit) => {
    console.log(fruit);
});
```

3. forEach로 접근

객체

키(key)와 값(value)로 구성된 프로퍼티(property)의 집합

키- 문자형

값- 모든 자료형

```
const people = {
    name: "lion",
    age: "23",
};
console.log(people.name);
people.address = "종로구";
console.log(people);
```

점 표기법

```
console.log(people["name"]);
console.log(people["age"]);
```

대괄호 표기법

객체 접근

for ... in으로 접근

```
for (key in people) {
    let value = people[key];
    console.log(`${key} : ${value}`);
}
```

조건문

특정 조건 만족 시 실행하는 명령의 집합

```
var score = 96;

if (score >= 90) {
    console.log("90점 이상");
} else {
    console.log("90점 미만");
}
```

반복문

명령을 반복적으로 실행해야 할 때 사용

```for
for (let i = 0; i < 3; i++) {
    console.log(i);
}
```

```while
let i = 0;
while (i < 3) {
    console.log(i);
    i++;
}
```

JS 기초

함수

```
function funcDeclarations() {
  return "A function declaration";
}
```

1. 함수 선언식

```
var funcExpression = function () {
  return "A function expression";
};
```

2. 함수 표현식

```
var arrowFunction = () => {
  return "A arrow function";
};
```

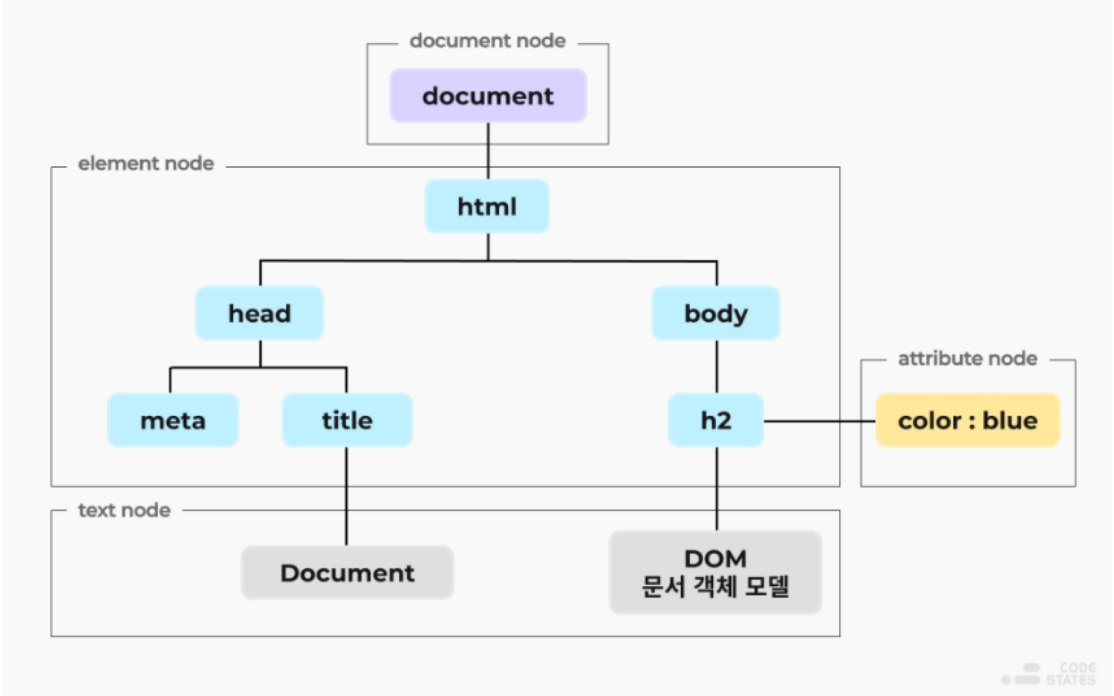
3. 화살표 함수

Document Object

DOM (Document Object Model)

HTML 문서의 프로그래밍 interface

프로그래밍 언어가 해당 문서에 접근하여 읽고 조작



Document Object

HTML 요소 선택

메소드	설명
getElementsByTagName(태그이름)	해당 태그 이름의 요소를 모두 선택
getElementById(아이디)	해당 아이디의 요소를 선택
getElementsByClassName(클래스이름)	해당 클래스와 일치하는 엘리먼트들의 HTMLCollection 반환

querySelector

querySelector

선택자와 일치하는 첫 번째 엘리먼트 반환

querySelectorAll

선택자와 일치하는 엘리먼트들의 NodeList 반환

연산 느리고 브라우저 지원 범위 좁음

HTMLCollection vs NodeList

HTMLCollection

- 유사배열
- 인덱스나 반복문을 통한 요소에 접근 가능, 배열 메서드는 사용 불가
- DOM의 변경사항 실시간으로 반영

NodeList

- 유사 배열
- 인덱스나 반복문을 통한 요소에 접근 가능, 배열 메서드 사용 불가
- forEach(), entires(), keys(), values() 사용 가능
- Dom의 변경사항 실시간으로 반영되지 않음

querySelector

HTML 요소 변경

```
<p class="p1">안녕하세요</p>
```

안녕하세요

```
document.querySelector(".p1").innerHTML = "Hello";
```

Hello

CSS 요소 변경

```
document.querySelector(".p1").style.color = "orange";
```

Hello

Form

form에 접근

```
<form action="" id="frm">
    <p>이름<input type="text" name="name" id="name" /></p>
    <p>나이<input type="text" name="age" id="age" /></p>
    <p>이메일<input type="text" name="email" id="email" /></p>
    <p>주소<input type="text" name="address" id="address" /></p>
</form>
```

```
const frm1 = document.getElementById("frm");
const frm2 = document.querySelector("#frm");
```

Form

elements에 접근

```
<form action="" id="frm">
    <p>이름<input type="text" name="name" id="name" /></p>
    <p>나이<input type="text" name="age" id="age" /></p>
    <p>이메일<input type="text" name="email" id="email" /></p>
    <p>주소<input type="text" name="address" id="address" /></p>
</form>
```

```
const name = frm1.name.value;
const age = frm1.elements[1].value;
const email = frm1.elements["email"].value;
const address = document.getElementById("address").value;
```

Form

form 제출

```html
<form action="" id="frm">
  이름: <input type="text" name="name" id="name" /> <br />
  나이: <input type="text" name="age" id="age" /> <br />
  <input type="button" value="제출" id="submitBtn" />
</form>
<ul id="list"></ul>
```

```js
document.getElementById("submitBtn").onclick = function submit() {
  const list = document.getElementById("list");
  const content =
    document.getElementById("name").value +
    ", " +
    document.getElementById("age").value;

  const item = document.createElement("li");
  item.innerHTML = content;
  list.appendChild(item);
};
```

이름:
나이:
제출

Event

마우스 이벤트

이벤트	설명
click	요소에 마우스 클릭했을 때 이벤트 발생
dbclick	요소에 마우스를 더블클릭했을 때 이벤트가 발생
mouseover	요소에 마우스를 오버했을 때 이벤트가 발생
mouseout	요소에 마우스를 아웃했을 때 이벤트가 발생
mousedown	요소에 마우스를 눌렀을 때 이벤트가 발생
mouseup	요소에 마우스를 떼었을 때 이벤트가 발생
mousemove	요소에 마우스를 움직였을 때 이벤트가 발생
contextmenu	context menu가 나오기 전에 이벤트 발생

Event

키 이벤트

이벤트	설명
keydown	키를 눌렀을 때 이벤트가 발생
keyup	키를 떼었을 때 이벤트가 발생
keypress	키를 누른 상태에서 이벤트가 발생

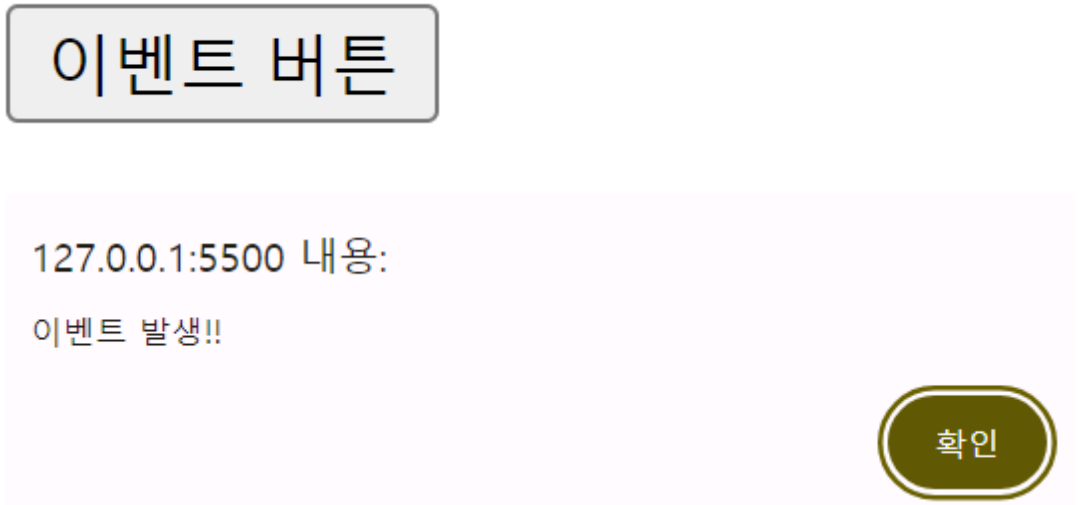
폼 이벤트

이벤트	설명
focus	요소에 포커스가 이동되었을 때 이벤트 발생
blur	요소에 포커스가 벗어났을 때 이벤트 발생
change	요소에 값이 변경 되었을 때 이벤트 발생
submit	submit 버튼을 눌렀을 때 이벤트 발생
reset	reset 버튼을 눌렀을 때 이벤트 발생
select	input이나 textarea 요소 안의 텍스트를 드래그하여 선택했을 때 이벤트 발생

Event

inline event

```
<input type="button" value="이벤트 버튼" id="btn" onclick="alert('이벤트 발생!!');" />
```



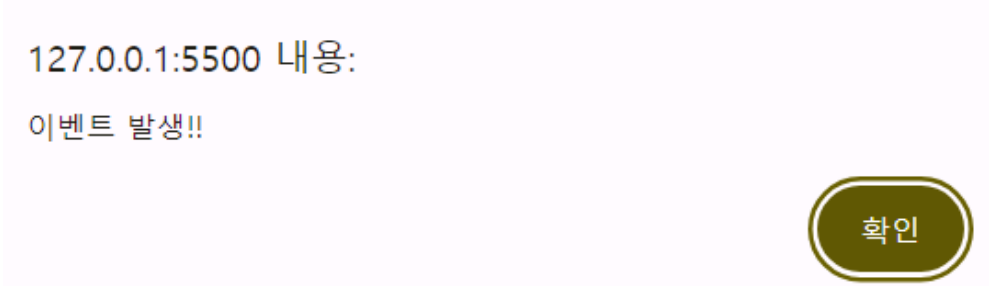
Event

property event

```
<input type="button" value="이벤트 버튼" id="btn" />
```

이벤트 버튼

```
var btn = document.getElementById("btn");
btn.onclick = function() {
    alert("이벤트 발생!!");
};
```



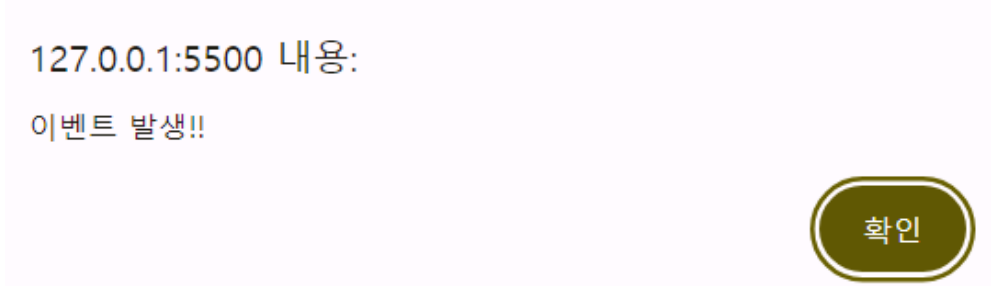
Event

addEventListener

```
<input type="button" value="이벤트 버튼" id="btn" />
```

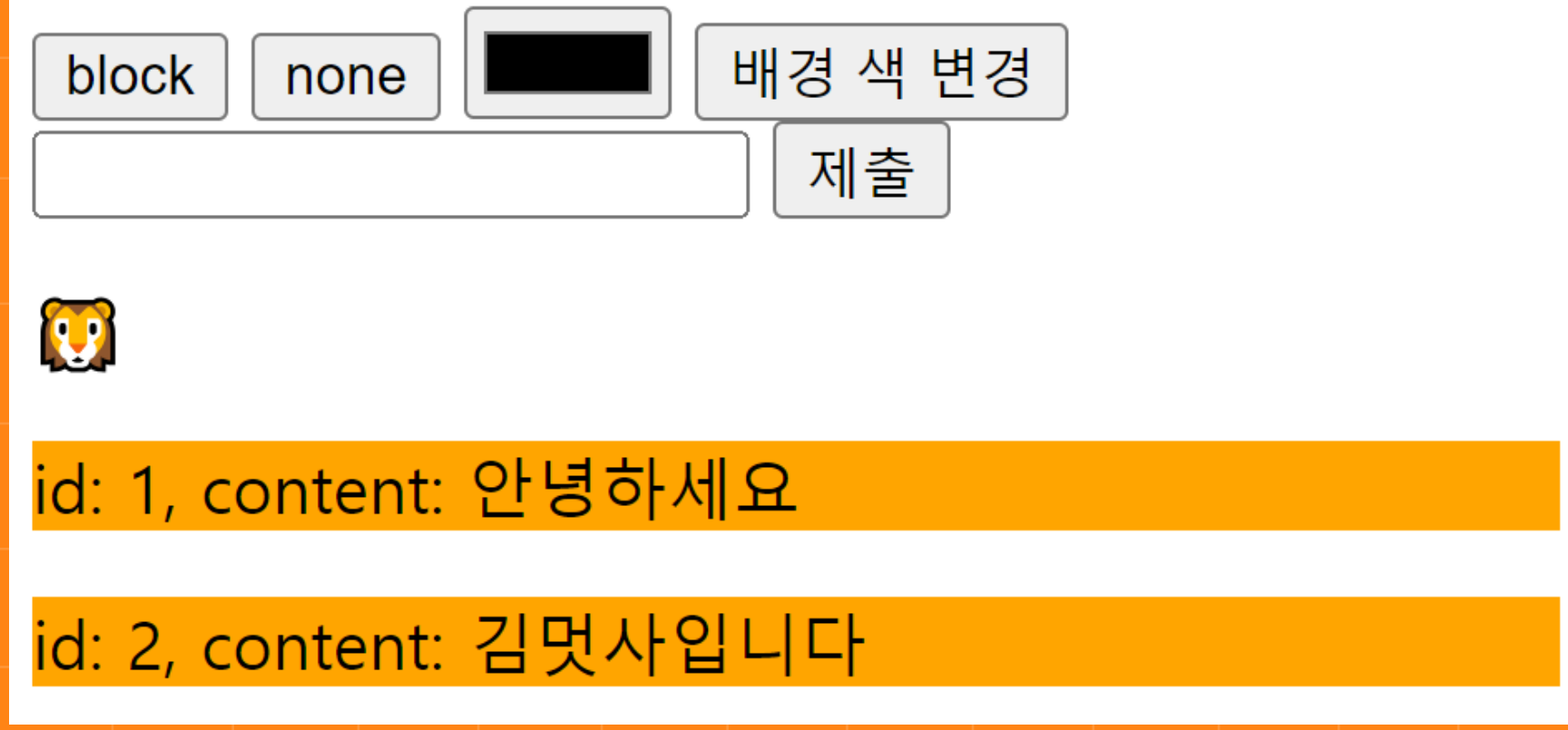
이벤트 버튼

```
var btn = document.getElementById("btn");
btn.addEventListener("click", function() {
    alert("event 발생!!");
});
```



LIKE LION UNIV.
×
SANGMYUNG UNIV.

실 습



hint
- `<input type="color" />`
- `document.body.style.backgroundColor`
- 증가 연산자 `++`

LIKE LION UNIV.
×
SANGMYUNG UNIV.

HTTP Method

- **GET** : 리소스 조회
- **POST** : 요청 데이터 처리, 주로 등록에 사용
- **PUT** : 리소스 대체, 없으면 생성
- **PATCH** : 리소스 부분 변경
- **DELETE** : 리소스 삭제
- POST와 PUT에만 body를 첨부 가능

HTTP Method

JSON (JavaScript Object Notation)

조화된 데이터를 표현하기 위한 문자 기반 표준 포맷

서버와 클라이언트 간 데이터를 전송할 때 사용

"키" : "값"

```
{
    "name": "Jang",
    "gender": "male",
    "age": 26,
    "alive": true
}
```

⚠ 큰 따옴표만 사용 가능

LIKE LION UNIV.
×
SANGMYUNG UNIV.

HTTP Method

API 명세서

💸 (가격 비교) 나라 선택

☑ 완료	☑
☰ http method	GET
☰ url	/api/compare/nation


```
{
    "code": 200,
    "httpStatus": "OK",
    "message": "요청에 성공하였습니다.",
    "data": [
        {
        "nationId": 1,
        "nationName": "가봉",
        "ImageUrl":
        },
        {
        "nationId": 2,
        "nationName": "가이아나",
        },
        {
        "nationId": 3,
        "nationName": "가이아나",
        },

        ... 생략

    ]
}
```

LIKE LION UNIV.
×
SANGMYUNG UNIV.

프론트엔드 6st 세미나

감사합니다