

# Day0509 - 자바스크립트 기초

## ▼ Dom (Document Object Model)

- 문서 객체 모델
- 문서내의 엘리먼트들을 정의하고 접근하는 방법을 제공

## ▼ function

- 함수
- method / procedure
- 특정 동작을 수행하는 일정 코드의 부분

## ▼ 네이티브 자바스크립트

- 다른 라이브러리 혹은 프레임워크를 사용하지 않은 순수 자바스크립트
  - 우리가 배울거

## ▼ 로그

프로그램의 흐름을 확인하기 위해서 남기는 기록  
개발하기 전 가장 먼저 설계하는 목표

## ▼ 컴파일

- 소스코드를 기계가 이해할 수 있도록 기계어로 번역하는 과정

## ▼ 인터프리트 언어

- 실시간으로 실행되는 언어
  - ex) Javascript, PHP, ASP, Python
    - 실행해봐야 오류가 나는 지 암
- 반) 컴파일 언어
  - ex ) C, C++, JAVA, Go

## ▼ 자바스크립트

### ▼ 목적

- 웹페이지를 동적으로 작동하게 만들기 위해 고안된 언어.

### ▼ 구조

#### 1. inline

- 태그 내부에 사용

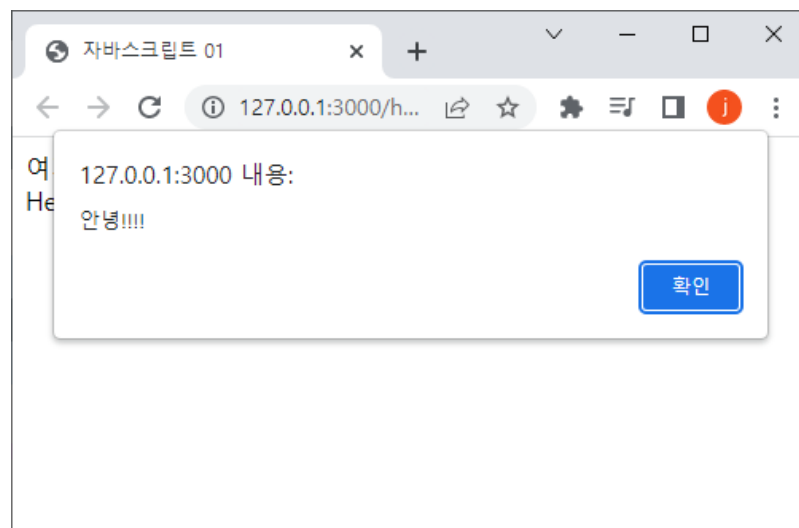


```
<div onclick="javascript:alert('click')">
  여기는 버튼이다
</div>

<div 이벤트="javascript:자바스크립트함수();">
```

## 2. embedded

- html 내부 어딘가에 삽입
- ex) body 처음



```
<script>
function hello(){
  alert("안녕!!!!");
}
```

```

</script>
<div onclick="javascript:hello()">
    Hello
</div>

```

### 3. external

- 외부 링크 삽입
- 자바스크립트의 확장자 : .js
- 로직이 분리되기 때문에 가장 좋은 방법
  - html에는 뼈대만
  - css에는 디자인만
  - js에는 기능만

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

```

function hi() {
    alert("Hi!!!!!!");
}

```

### ▼ 변수 선언

JAVA와 다르게 자료형이 없음

- **var**

- 중복 선언 가능

```

> var i = 20;
< undefined
> i
< 20
> var i = 10;
< undefined
> i
< 10

```

- 중복 선언으로 형변환 가능

```

> var i = "hello";
< undefined
> i
< 'hello'

```

- 현업에서 잘 사용 하지 않음

- **let**

- 중복 선언 불가

- **const**

- 반드시 선언과 동시에 초기화
- 선언만 하고 초기화 하지 않으면 에러!

```

> const text;
❌ Uncaught SyntaxError: Missing initializer in const declaration

```

- 변경할 내용이 없다면 보통 사용
- Java의 `final` 과 동일

#### ▼ 주석

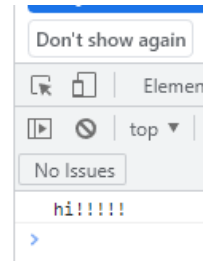
- 여러줄 주석 `/* */`
- 한줄 주석 `//`

#### ▼ 로그처리

- `console.log` → 콘솔창에서 확인하는 `print`

```
function hi() {
  console.log("hi!!!!!!");
}
```

hi



#### ▼ 연산

- 곱하기 `*`
- 나누기 `/`
- 나머지 `%`
- 더하기 `+`
  - 문자열 병합 가능
- 문자열 병합
  - `+`
  - `alert(`${변수명}병합할 문자열`);`

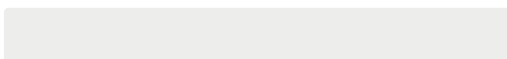
#### ▼ 타입확인

- `typeof`
  - `console.log(typeof 변수);`

```
> typeof 1
< 'number'
> typeof 'hello'
< 'string'
> typeof true
< 'boolean'
> typeof null
< 'object'
```

#### ▼ 함수

- 선언



```
// parameter 0
function sum(x, y){
    let result = x + y;
    console.log(result);
}

// return 0
function total(x, y) {
    return x + y;
}
```

```
> sum(10,20)
30
< undefined
> total(10,20)
< 30
> var result = total(10,20);
< undefined
> result
< 30
> total('hi', '지현')
< 'hi지현'
> total(true, false);
< 1
```

- 자바스크립트에는 자료형이 없어서 함수명만으로는 리턴타입과, 매개변수의 타입을 알 수 없음

#### ▼ 자주 사용하는 함수

##### ▼ 메시지 창 띄우기

- `alert('메시지')` - 경고창 실행
- `confirm('메시지');` - 확인 취소 창
- `prompt('메시지');` - 입력창

##### ▼ 제어문

- `if()`
  - `===` 으로 사용
  - `==` 을 제외한 나머지는 java랑 동일
- `isNaN()` - 숫자가 아닌지 확인

#### ▼ 셀렉터

- `document.getElementById('아이디명')`
  - html에 해당 id를 가진 객체 짜란

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, height=device-height">
    <title>Document</title>
</head>
<body>
    <div id="div1_id"></div>

    <div class="div1_class" name="div1_name">1</div>
    <div class="div1_class">2</div>
    <div class="div1_class">3</div>

    <input type="text" name="name">
</body>
</html>
```

```
> document.getElementById('div1_id')
< div id="div1_id"></div>
> let div1 = document.getElementById('div1_id')
< undefined
> div1
< div id="div1_id"></div>
> document.getElementsByClassName('div1_class')
< HTMLCollection(3) [div.div1_class, div.div1_class,
  div.div1_class, div1_name: div.div1_class]
  ▶ 0: div.div1_class
  ▶ 1: div.div1_class
  ▶ 2: div.div1_class
  ▶ div1_name: div.div1_class
  length: 3
  ▶ [[Prototype]]: HTMLCollection
> let nums = document.getElementsByClassName('div1_class')
< undefined
> nums[0]
< div class="div1_class" name="div1_name">1</div>
> nums[1]
< div class="div1_class">2</div>
>
```

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <style>
      .big-text {
        font-size: xx-large;
      }

      .blue-color {
        color: blue;
      }
    </style>
  </head>
  <body>
    <div id="div1_id"></div>

    <div class="div1_class" name="div1_name">1</div>
    <div class="div1_class">2</div>
    <div class="div1_class">3</div>

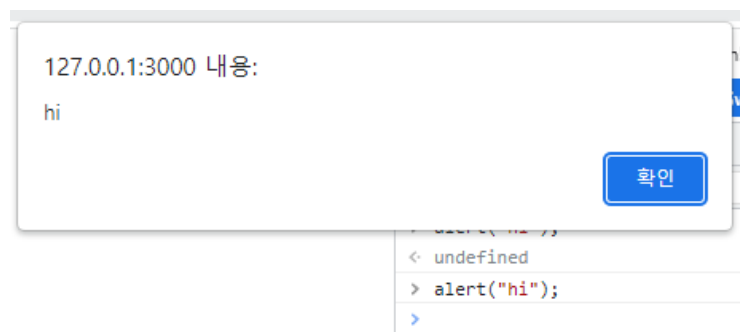
    <input type="text" name="name" />
  </body>
</html>

```



#### ▼ 이벤트

- onclick - 클릭했을 때
- onmouseover - 마우스를 올렸을 때
- onmouseleave - 마우스가 떠났을 때
- onload - 페이지가 모두 나오고 나서 실행
  - 자바스크립트는 먼저 실행되기 때문에 나중에 실행됐으면 할 때 body에 사용...?
- 
- 크롬 → 개발자도구(f12) → **console**
  - 간단한 문법 연습 가능



- 변수 치고 엔터 치면 자동 **print**

## ▼ 실습

### ▼ 실습1 - 이름입력받기

- `function` 과 `변수` 의 이름이 동일하면 오류

127.0.0.1:3000 내용:

이름을 입력해 주세요

확인

취소

이름 입력 창

127.0.0.1:3000 내용:

다시입력해 주세요.

확인

빈칸이거나 취소를 눌렀을 시 다시 입력

127.0.0.1:3000 내용:

홍길동님 안녕하세요.

확인

이름 입력 시 이름이 콘솔창과 경고창에 출력

홍길동



```
function start() {  
  const name = prompt("이름을 입력해 주세요");  
  // 지금 코드에서 바뀔 일이 없기 때문에 const가 더 좋음  
  checkName(name);  
}  
  
function checkName(name) {  
  if (name === null || name === "") {  
    alert("다시입력해 주세요.");  
    start();  
  } else {  
    console.log(name);  
    alert(name + "님 안녕하세요.");  
  }  
}
```

<button onclick="javascript:start()">클릭클릭</button>

### ▼ 실습2 - isNaN, for문 연습

127.0.0.1:3000 내용:

숫자를 입력해주세요.

확인

취소

숫자가 아닌 값 입력 시 다시 입력

127.0.0.1:3000 내용:

1이상 100 이하를 넣어주세요

확인

1~100 범위 밖의 값을 입력 시

127.0.0.1:3000 내용:

완료되었습니다.

확인

1~100 범위의 값 입력 시

92

93

94

95

96

97

98

99

100

제대로된 값 입력 시  
콘솔창

```
<body>
  <button onclick="javascript:enterNum()">클릭클릭</button>
</body>
```

```
function enterNum() {
  const num = prompt("숫자를 입력해주세요.");
  checkNum(num);
}

function checkNum(num) {
  if (isNaN(num)) {
    alert("숫자를 입력해 주세요");
  } else if (num < 0 || num > 100) {
    alert("1이상 100 이하를 넣어주세요");
  } else {
    printNum();
  }
}

function printNum() {
  for (let i = 1; i <= 100; i++) {
    console.log(i);
  }
  alert("완료되었습니다.");
}
```

### ▼ 실습3 - 텍스트 입력 후 컬러변경 버튼 만들기

127.0.0.1:3000 내용:

텍스트를 입력해 주세요.

확인

취소

텍스트 입력창

## 텍스트

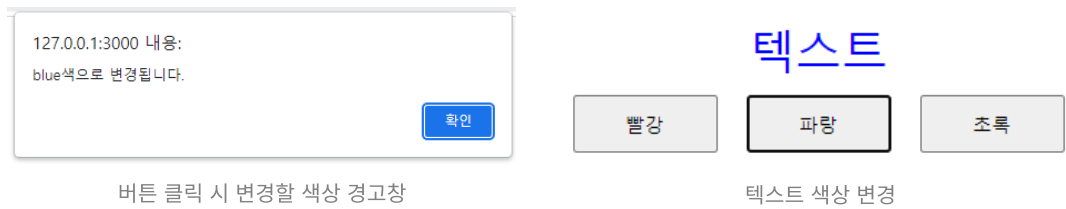
빨강

파랑

초록

텍스트 입력 후 버튼 만들어 짐





```
<div class="container">
  <div id="text"></div>
  <div id="btn">
    </div>
</div>
```

```
.container {
  margin: 0 auto;
  padding: 25%;
  text-align: center;
}

#text {
  font-size: xx-large;
}

#btn{
  margin: 0 auto;
}

#btn > button {
  margin: 10px;
  width: 100px;
  height: 40px;
}
```

```
function start(){
  let text;
  do{
    text = prompt('텍스트를 입력해 주세요. ');
  }while(text === null || text.length === 0);
  document.getElementById("text").innerHTML = text;
  document.getElementById("btn").innerHTML = "<button onclick = changecolor('red')>빨강</button>"
  + "<button onclick = changecolor('blue')>파랑</button>"
  + "<button onclick = changecolor('green')>초록</button>";
}

function changecolor(color){
  let div = document.getElementById("text");
  alert(color + '색으로 변경됩니다.')
  div.style.color = color;
}
```

#### ▼ 실습4 - 메뉴만들기(클릭 시 색상변경)

<https://codepen.io/HeyJihyun/pen/QWZmbvY>

▼ 실습5 - 메뉴반들기(hover 시 색상변경)

<https://codepen.io/HeyJihyun/pen/NWOYjGe>