

Javascript 문법
굵은 글씨 내용 재조정

Javascript란?

A large, bold, black 'JS' logo is centered on a solid yellow rectangular background.

객체 기반의 스크립트 프로그래밍 언어이다.
웹 브라우저 내에서 주로 사용하며 다른 응용 프로그램
의 내장 객체에도 접근할 수 있는 기능을 가지고 있다.

이름 변화 : 모카 > 라이브스크립트 > 자바스크립트

자바스크립트는 정적(브라우저가 한번 HTML,CSS 읽은
후 변하지 않음)으로 움직이는 HTML&CSS와는 달리 정
적인 데이터를 실시간으로 변경하고 계산,사용자 입력처
리, 경고메시지 등 동적인 내용을 사용자에게 바로 보이
게 한다.

정적 VS 동적

- 정적 타입 (ex, C언어 , java ...)

컴파일 시 타입 결정

`int num = 1;` (변수 선언 시 num 변수가 int 타입으로 선언됨)

- 동적 타입 (ex. JS, 파이썬 ...)

런타임 (코드 실행 시)에 타입 결정

`var num = 1;`

`Num="number1"`

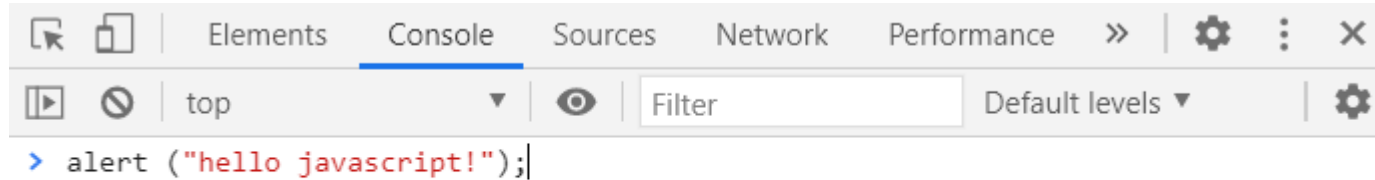
(따로 int 타입 선언 해주지 않고 num이 '1'이란 숫자로 선언)

Javascript로 할 수 있는 일은?

- **컨텐츠 바꾸기**
- 멀티미디어 다루기
- 움직이는 이미지 , 경고창, drag&drop 등등
- Html,css 정적으로 만들어진 웹페이지를 동적으로 변경

단순하지만 중요한것들만 가지고 있고 그러면서도 다양한 곳에서 사용되고 있는 '세상에서 제일 많이 쓰이는 단어'

JS 맛보기



구글 크롬 F12 콘솔창
`alert ("hello javascript!");` 입력

new-tab-page에 삽입된 페이지 내용:

hello javascript!

확인

경고창(alert를 사용했기 때문에)
으로 hello javascript! 가 출력됨

JS 코드는 어느 부분에 넣나요?

- html 문서 내 `<script>~</script>` 사이에 JS 코드 넣기
예시

외부 파일(*.js) 형식으로 작성하고 html 문서에 연결
`<script src="외부파일명.js"></script>`
예시

기초 문법- 문장,주석

```
> var a,b,c;  
< undefined  
> a=10; //a에다 10이라는 숫자값을 넣는다  
< 10  
> b=20; //b에다 20이라는 숫자값을 넣는다  
< 20  
> c=a+b; //c에다 a값과 b값을 더한 값을 넣는다  
< 30  
>
```

- 문장과 주석

문장 = 한 줄 한줄에 의미가 있는 것 (프로그램이 수행할 명령 지정)

서로 다른 의미를 가지고 있으므로 각각의 문장은
; 세미콜론으로 구분한다.

주석 = 그 코드를 설명하기 위한 문장
(코딩 시에는 보이지 않는다)

변수와 값

- 변수 : 값을 저장하기 위한 공간

> 값이 물이라면 그릇은 변수

변수에는 정수, 실수, 논리형(Boolean) 등 다양한 변수 저장 가능

변수 선언 : `var a,b,c;` (예시)

변수 초기화 : `var a=2;` (변수 선언과 동시에 2라는 값으로 사용한
다는 것을 약속한다) 2 말고는 다른 값 못 옴 !

```
> var a,b,c;  
< undefined  
  
> a=10; //a에다 10이라는 숫자값을 넣는다  
< 10  
  
> b=20; //b에다 20이라는 숫자값을 넣는다  
< 20  
  
> c=a+b; //c에다 a값과 b값을 더한 값을 넣는다  
< 30  
>
```


식별자 (대상을 식별하기 위한 이름 만들기)

- 이름을 붙일 때 사용하는 단어
어디에 사용? 변수 이름이나 함수 이름을 구성할 때 사용

문자, 숫자, `_`, `$` 기호로 구성

숫자로 시작하면 안됨 ex) 4do

공백 입력하면 안됨 ex) h ello

키워드 사용 안됨 ex) true, while, if 와 같이 이미 문법으로 정의된 이름 같은 것들

식별자 주의점 알려주기?

기본 자료형 - 숫자 Number

- 정수 (int형)

소수점이 없는 수 (1,2,3...)

음수(-) , 양수(+)

ex) a=1 ; / b=-6 ;

- 실수 (float, double형)

소수점이 있는 수 (2.3 , 4.555555)

>float형까지 알려 줄 필요가 있을까?

문자열(String)

- 문자의 집합

➤ `console.log("BTS"+"는"+"방탄소년단");`

> `console.log("BTS"+"는"+"방탄소년단");`

BTS는방탄소년단

웹 페이지 안에 문자나 문장을 넣고 싶다면 큰따옴표, 작은따옴표를 사용

문자열 연결 연산자는 +

불린형 Boolean

- 참(true)과 거짓(false)의 싸움

```
> var a=5,b=10;
```

```
< undefined
```

```
> console.log(a>b);
```

```
false
```

```
< undefined
```

```
> console.log(a<b);
```

```
true
```

결과 출력 방법

- 콘솔에 출력 `console.log()`
- 팝업창에 출력 `window.alert()`
- 브라우저에 출력 `document.write()`
- **문서객체모델(DOM)**에 출력 : `innerHTML`

변수 및 연산자 활용 문제

- 숫자 2개를 입력받아서 덧셈과 뺄셈 결과 출력해보기

```
<script>
var num1=prompt("숫자 2개로 덧셈\wn 첫번째 숫자 입력하기");
var num2=prompt("두번째 숫자 입력");

document.write(num1+" "+num2+"="+ (num1+num2)+"<br>");
</script>
```

그렇다면 뺄셈은 어떻게 해야 할까요?

기초 문법 - 조건문

- If , if else, if else if, switch

- If문 (만약에~)

If(조건식){

코드블록 // 조건식이 참 일때 실행되는 부분

}

조건식이 참(true)이면 코드 블록 실행 후 if 문 탈출

If(조건식)에서 거짓이면 바로 if문 벗어남

예제 : 홀짝 구분

- If~else문

If(조건식) {

 코드블록 1 // 조건식이 참일때 실행되는 부분

}

Else {

 코드블록 2 // 조건식이 거짓일 때 실행되는 부분

}

If 조건식이 참 = 코드블록 1 실행

If 조건식이 거짓 = 코드블록 2 실행 > 기준은 첫번째 if다!

If문 예시 - 홀/짝 구분

```
<script>
Function showResult() {
    var num1=document.getElementById("num1");
    if(num1%2==0){
        alert(num1.value+"는 짝수");
    }
    else { alert(num1.value+"는 홀수"); }
}
</script>
```

- If else if 문

```
if(조건식1) {  
    // 조건식 1이 참일 때 실행  
    코드블록 1  
}  
Else if (조건식 2) {  
    // 조건식 2가 참일 때 실행 (조건식1 거짓, 조건식 2 참)  
    코드블록 2  
}  
Else {  
    // 조건식 1,2 거짓  
    코드블록 3  
}
```

예제 : 등급에 따른 성적 프로그램

If~else if~else 예시 : 성적 출력 프로그램

- 방탄이의 수학 성적을 입력받고 if~else if~else에 따라 처리

<script>

```
var score=Number(prompt("점수 입력"));
```

```
if(score>=90){  
    document.write("A등급 입니다"); }
```

```
else if(score>=80){  
    document.write("B등급 입니다");}
```

```
else if(score>=70){  
    document.write("C등급 입니다");}
```

```
else { document.write("D등급 입니다");}
```

>> 방탄이가 입력한 수학 성적에 따라 그에 걸맞는 if문 코드 블록 수행.

- Switch문

값에 따라 서로 다른 코드블록을 실행 할 때 사용

```
Switch(표현식) {  
    case 값1 : 코드블록 1; break;  
    case 값2 : 코드블록 2; break;  
    ....  
    default : 코드블록 n; (생략 가능)  
}
```

예제 : 요일 출력

Switch문 예제 - 요일 출력

- <p id="output"> </p>

```
<script>
    var day;
    Switch(new Date().getDay()) {
        case 0; day = "월요일"; break;
        case 1; day = "화요일"; break;
        case 2; day = "수요일"; break;
        case 3; day = "목요일"; break;
        case 4; day = "금요일"; break;
        case 5; day = "토요일"; break;
        case 6; day = "일요일"; break;
    }
    Document.getElementById("output").innerHTML+="오늘은"+day+"입니다";
</script>
```

반복문 for

- 동일한 작업을 반복 처리 해야 할 경우 > 같은 코드를 여러 줄 쓰게 된다면 비효율적.
>> 같은 코드를 원하는 횟수만큼 반복처리 해서 간편하게 구현할 수 있도록 하는 문법

반복문 사용 안 한 경우 : `Console.log("1"); ~ console.log("5");`

반복문 사용 한 경우 : `for (var i=1; i<=5; i++) {
 console.log(i);
}`

While, do while도 알려줘야하나?

For문 예제 – 10명의 손님이 왔다면?

```
for (var i=1;i<=10;i++){  
    console.log(i+"번째 손님");}
```

1번째 손님

2번째 손님

3번째 손님

4번째 손님

5번째 손님

6번째 손님

7번째 손님

8번째 손님

9번째 손님

10번째 손님

반복문 – while문

- while(조건식){
 반복할 코드 블록
}

조건식을 먼저 검사하고 값이 참(true)인 동안 코드 블록을 반복 실행
반복 횟수가 미리 정해지지 않은 경우 주로 사용

Ex) var i=1;
While (i<=10){
 console.log(i+"번째 손님");
 i++;
}

```
var i=1;  
while (i<=10){  
    console.log(i+"번째 손님");  
    i++;  
}
```

1번째 손님

2번째 손님

3번째 손님

4번째 손님

5번째 손님

6번째 손님

7번째 손님

8번째 손님

9번째 손님

10번째 손님

For문과 while문의 차이

- 공통점은 둘 다 같은 반복문!
- For문 – 반복 횟수가 정해져 있을 경우에 사용
Ex) 10번 반복해라, 배열-길이가 정해져 있으므로 사용 유리

While문 – 반복 횟수가 정해지지 않았을 경우에 사용
Ex) 무한 반복, 특정 조건에 만족할 때까지 반복해야 하는 경우

배열 Array

- 여러 값을 표현하기 위한 방법
- 배열 생성은 [] ([] 안에 데이터를 넣는다)

Array() 객체 생성자 사용

BTS 멤버 : RM, 슈가, 제이홉, 진, 지민, 뷔, 정국

➤ BTS 멤버를 배열에 넣고 싶다면?

```
var BTS = ["RM","슈가","제이홉","진","지민","뷔","정국"]
```

비슷한 것들을 하나로 묶는다. 즉 이들은 관련 있는 데이터들

> 각각이 모두 'BTS 멤버' 라는 특징을 가지고 있어 하나의 배열로 묶을 수 있음.

배열의 인덱스

- var BTS = ["RM","슈가","제이홉","진","지민","뷔","정국"]
 0 1 2 3 4 5 6

우리는 인덱스 넘버가 1부터라고 생각하기 쉽다.

그러나 배열의 인덱스 넘버는 0부터 시작 !

Ex) BTS[0] = ?

BTS[3] = ?

BTS[6] = ?

getElementById

- 자바스크립트에서 중요한 함수
- **get(가져오다)+Element(요소)+ById(Id로부터)**

```
<body>
<input type="button" id="btn" value="button"/>
<script>
    var btn=document.getElementById("btn");
    alert(btn); //변수 btn에 id가 btn인 요소의 정보를 담고 alert 대화상자에 정보 넣어주기
</script>
</body>
```

Document.get~ 인 이유는 getElementById가 document 객체의 내장함수여서 . 을 붙인다.

document.write

- html에서 텍스트를 써주는 것과 같은 기능
- 사용법 : document.write("화면에 출력하고 싶은 말");

```
<script>
```

```
    document.write("Do you like BTS?");
```

```
</script>
```

여러 문장을 쓸 경우 자동으로 줄바꿈이 되지 않아
띄어쓰기 태그인
 태그를 반드시 써줄 것!



**SULLIVAN
PROJECT**