

JavaScript (Part 1)

What is JavaScript?

- 객체 기반의 동적 스크립트 언어
- 타입을 명시할 필요가 없는 인터프리터 언어
 - 대부분의 웹브라우저는 자바스크립트 인터프리터를 내장
- 자바와는 연관성이 없음
- Node.js 통해 브라우저 밖에서도 사용가능하며 즉 서버측 프로그래밍에도 사용가능
- HTML로는 웹의 구조 및 내용을 작성하고, CSS로는 스타일링을 하고, 자바스크립트로는 웹과 사용자간의 상호작용을 구현

JavaScript 기능 Overview

1. HTML 내용 변경

```
<!DOCTYPE html>
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change HTML content.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML =
"Hello JavaScript!'">Click Me!</button>

</body>
</html>
```

What Can JavaScript Do?

Hello JavaScript!

Click Me!

What Can JavaScript Do?

JavaScript can change HTML content.

Click Me!

JavaScript 기능 Overview

2. HTML 속성 변경

```
<!DOCTYPE html>
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p>JavaScript can change HTML attribute values.</p>

<p>In this case JavaScript changes the value of the src (source) attribute of
an image.</p>

<button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn
on the light</button>



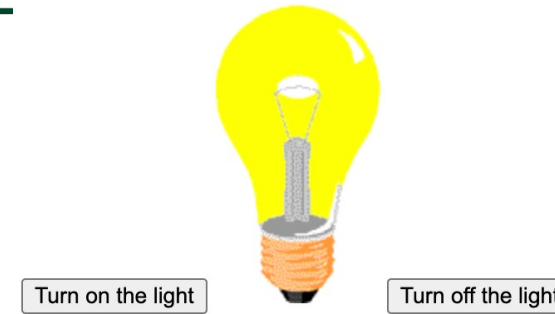
<button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn
off the light</button>

</body>
</html>
```

What Can JavaScript Do?

JavaScript can change HTML attribute values.

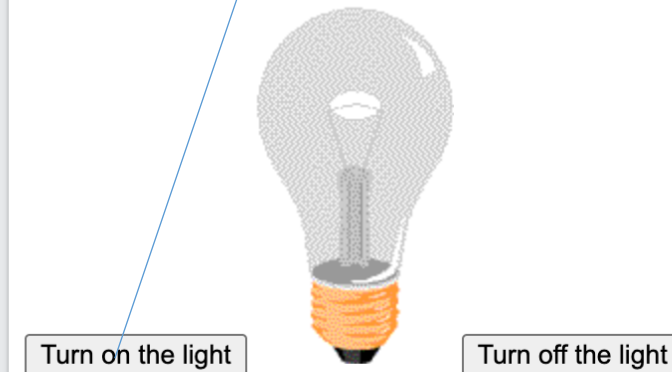
In this case JavaScript changes the value of the src (source) attribute of an image.



What Can JavaScript Do?

JavaScript can change HTML attribute values.

In this case JavaScript changes the value of the src (source) attribute of an image.



JavaScript 기능 Overview

3. HTML 스타일 변경

```
<!DOCTYPE html>
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change the style of an HTML element.</p>

<button type="button"
onclick="document.getElementById('demo').style.fontSize='35px'">Click Me!
</button>

</body>
</html>
```

What Can JavaScript Do?

JavaScript can change the style of an HTML element.

Click Me!

What Can JavaScript Do?

JavaScript can change the style of an HTML element.

Click Me!

JavaScript 문법 1

-- 튜토리얼 by w3schools & tcpschool.com

Table of contents

- Type
- Operator
- Function
- Object
- String methods
- Array methods
- Event

JavaScript 적용법

1. 내부 JavaScript : <head> <script>...</script> </head>

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>

<h2>Demo JavaScript in Head</h2>

<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

</body>
</html>
```


JavaScript 적용법

1. 내부 JavaScript: <body> <script>....</script> </body>

```
<!DOCTYPE html>
<html>
<body>

<h2>Demo JavaScript in Body</h2>

<p id="demo">A Paragraph</p>

<button type="button" onclick="myFunction()">Try it</button>

<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>

</body>
</html>
```

JavaScript 적용법

2. 외부 JavaScript: <script src="myjsfile.js"> </script>

```
<!DOCTYPE html>
<html>
<body>

<h2>Demo External JavaScript</h2>

<p id="demo">A Paragraph.</p>

<button type="button" onclick="myFunction()">Try it</button>

<p>This example links to "myScript.js".</p>
<p>(myFunction is stored in "myScript.js")</p>

<script src="myScript.js"></script>

</body>
</html>
```

HTML 페이지에 출력방법1: window.alert()

```
<body>
  <h1>Window 객체의 alert() 메소드</h1>
  <button onclick="alertDialogBox()">alert 대화 상자</button>
  <script>
    function alertDialogBox() {
      alert("확인을 누를 때까지 다른 작업을 할 수 없어요!");
    }
  </script>
</body>
```

확인을 누를 때까지 다른 작업을 할 수 없어요!

OK

HTML 페이지에 출력방법2: innerHTML

```
<body>
```

```
  <h1>HTML DOM 요소를 이용한 innerHTML 프로퍼티</h1>
```

```
  <p id="text">이 문장을 바꿀 것입니다!</p>
```

```
  <script>
```

```
    var str = document.getElementById("text");
```

```
    str.innerHTML = "이 문장으로 바뀌었습니다!";
```

```
  </script>
```

```
</body>
```

HTML DOM 요소를 이용한 innerHTML 프로퍼티

이 문장으로 바뀌었습니다!

HTML 페이지에 출력방법3: document.write()

<body>

<h1>Document 객체의 write() 메소드</h1>

<button onclick="document.write(4 * 5)">버튼을 눌러
보세요!</button>

</body>

Document 객체의 write() 메소드

버튼을 눌러보세요!

20

변수선언

- **var:** 변수 재선언, 재할당 모두 가능

```
var a = 'osp';  
var a = 'osp2';  
a = 'osp3';
```

- **let:** 변수 재할당만 가능

```
let a = 'osp';  
let a = 'osp2'; //Syntax Error  
a = 'osp3';
```

- **const:** 변수 재선언, 재할당 모두 불가능

```
const a = 'osp';  
const a = 'osp2'; //Syntax Error  
a = 'osp3'; //Type Error
```

- **Using nothing**

```
x = 5;  
y = 6;  
z = x + y;
```

Data type

- Number (e.g., `var num = 10;`)
- String (e.g., `var className = "오소풀";`) '오소풀' 사용가능
- Boolean
- Undefined (e.g., `var str;`) 정의되지 않은 변수의 값
- Object (e.g., `var dog={name:"happy", age:3};`)

typeof

- `typeof "문자열" // string`
- `typeof 10 // number`
- `typeof NaN // number`
- `typeof false // boolean`
- `typeof undefined // undefined`
- `typeof new Date() // object`
- `typeof null // object`

Input	Output
숫자, NaN	"number"
문자열	"string"
true, false	"boolean"
null	"object"
undefined	"undefined"
함수	"function"
함수가 아닌 객체	"object"

Object type usage

```
var dog = { name: "해피", age: 3 }; // 객체의 생성
```

```
document.getElementById("result").innerHTML =  
    "강아지의 이름은 " + dog.name + "이고, 나이는 " + dog.age + "  
    살 입니다.";
```

Type conversion

1. Implicit type conversion: 자동 타입변환 가능

- 예시

```
var num = 20; // Number 타입의 20
```

```
num = "이십"; // String 타입의 "이십"
```

```
var month; // month라는 이름의 변수 선언
```

```
date = 25; // date라는 이름의 변수를 묵시적으로 선언
```

Type conversion

2. Explicit type conversion: 아래함수제공

문자열을 파싱(parsing)하여, 문자열에 포함된 숫자 부분을 실수 형태로 반환

- Number(), String(), Boolean(), Object(), parseInt(), parseFloat(), toString()

- 예시

문자열을 파싱하여, 문자열에 포함된 숫자 부분을 정수 형태로 반환

```
Number("10"); // 숫자 10
```

```
String(true); // 문자열 "true"
```

```
Boolean(0); // 불리언 false
```

```
Object(3); // new Number(3)와 동일한 결과로 숫자 3
```

```
false.toString(); // 문자열 "false"
```

```
Number(true); // 숫자 1
```

```
Number(false); // 숫자 0
```

산술연산자

```
var x = 10, y = 4;
```

```
document.write(x + y + "<br>"); // 14
```

```
document.write(x - y + "<br>"); // 6
```

```
document.write(x * y + "<br>"); // 40
```

```
document.write(x / y + "<br>"); // 2.5
```

```
document.write(x % y); // 2
```

비교연산자

```
var x = 3, y = 5;
```

```
var a = "abc", b = "bcd";
```

```
document.write((x > y) + "<br>"); // y의 값이 x의 값보다 크므로 false
```

```
document.write((a <= b) + "<br>"); // 알파벳 순서상 'a'가 'b'보다 먼저  
나오므로 'a'가 'b'보다 작으므로 true.
```

```
document.write(x < a); // x의 값은 숫자이고 a의 값은 문자열이  
므로 비교할 수 없음으로 false.
```

문자열 결합연산자

var x = 3 + 4; // 피연산자가 둘 다 숫자이면 덧셈 연산을 수행함.

*var y = "좋은 " + "하루 되세요!" // 피연산자가 둘 다 문자열이면
문자열 결합 연산을 수행함.*

*var z = 12 + "월" // 피연산자가 하나라도 문자열이면 문자열 결
합 연산을 수행함.*

Delete 연산자

`var arr = [1, 2, 3];` *// 배열 생성*

`delete arr[2];` *// 배열의 원소 중 인덱스가 2 인 요소(즉 3)를 삭제함.*

`document.write(arr + "
");` *// 1, 2, 출력됨*

`document.write(arr[2] + "
");` *// 배열에 빈자리가 생긴 것으로 undefined 값이 됨*

`document.write(arr.length);` *// 3 이 출력됨. 배열의 요소를 삭제하는 것이지 배열의 길이까지 줄이는 것은 아님.*

조건문 (if, if/else, if/else if/else)

if (표현식1) {

표현식1의 결과가 참일 때 실행하고자 하는 실행문;

} else if (표현식2) {

표현식2의 결과가 참일 때 실행하고자 하는 실행문;

} else {

표현식1의 결과도 거짓이고, 표현식2의 결과도 거짓일 때 실행하고자 하는 실행문;

}

조건문 (switch)

```
switch (조건 값) {  
    case 값1:  
        조건 값이 값1일 때 실행하고자 하는 실행문;  
        break;  
    case 값2:  
        조건 값이 값2일 때 실행하고자 하는 실행문;  
        break;  
    ...  
    default:  
        조건 값이 어떠한 case 절에도 해당하지 않을 때 실행하고자 하는 실행문;  
        break;  
}
```

조건문 예제

```
<!DOCTYPE html>
<html lang="ko">

<head>
  <meta charset="UTF-8">
  <title>JavaScript Conditional Statement</title>
</head>

<body>

  <h1>switch 문</h1>

  <script>
    var x = "문자열";

    switch (typeof x) {
    case "number":
      document.write("변수 x의 타입은 숫자입니다.<br>");
      break;
    case "string":
      document.write("변수 x의 타입은 문자열입니다.<br>");
      break;
    case "object":
      document.write("변수 x의 타입은 객체입니다.<br>");
      break;
    default:
      document.write("변수 x의 타입을 잘 모르겠네요...<br>");
      break;
    }
  </script>

```

switch 문

변수 x의 타입은 문자열입니다.

반복문

- While문

`while` (표현식) {

표현식의 결과가 참인 동안 반복적으로 실행하고자 하는 실행문;

}

- For문

`for` (초기식; 표현식; 증감식) {

표현식의 결과가 참인 동안 반복적으로 실행하고자 하는 실행문;

}

For문 예제 1

```
var arr = [3, 4, 5];
```

```
for (var i = 0; i < arr.length; i++) { // 배열 arr의 모든 요소의 인덱스(index)  
    )를 출력함.
```

```
    document.write(i + " ");           0 1 2  
}
```

```
for (var i in arr) { // 위와 같은 동작을 하는 for / in 문
```

```
    document.write(i + " ");           0 1 2  
}
```

For문 예제 2

```
var arr = [3, 4, 5];
```

```
for (var i = 0; i < arr.length; i++) { // 배열 arr의 모든 요소의 인덱스(index)  
    )를 출력함.
```

```
    document.write(arr[i] + " ");          3 4 5  
}
```

```
for (var value of arr) { // 위와 같은 동작을 하는 for / of 문
```

```
    document.write(value + " ");          3 4 5  
}
```

함수

```
function 함수이름(매개변수1, 매개변수2,...) {  
    함수가 호출되었을 때 실행하고자 하는 실행문;  
}
```

함수예제

<body>

<h1>함수예제 </h1>

<script>

function sqr(x) { // 제공의 값을 구하는 함수 sqr를 정의함.

return x * x;

}

var sqrNum = sqr; // 변수 sqrNum에 함수 sqr을 할당함.

document.write(sqr(4) + "
"); // 함수 sqr을 호출함.

document.write(sqrNum(4)); // 변수 sqrNum를 함수처럼 호출함.

</script>

</body>

16

16

객체란?

- 예시: 고양이



프로퍼티(property)

- cat.name = "나비"
- cat.family = "코리안 숏 헤어"
- cat.age = 0.1
- cat.weight = 300

메소드(method)

- cat.mew()
- cat.eat()
- cat.sleep()
- cat.play()

객체: Property 참조예시

```
var person = {  
  name: "홍길동",    // 이름 프로퍼티를 정의함.  
  birthday: "030219", // 생년월일 프로퍼티를 정의함.  
  pld: "1234567",    // 개인id 프로퍼티를 정의함.  
  fullId: function() { // 생년월일과 개인id를 합쳐서 주민등록번호를 반환함.  
    return this.birthday + this.pld;  
  }  
};  
  
person.name // 홍길동  
person["name"] // 홍길동
```

객체: Method 참조예시

```
var person = {  
  name: "홍길동",  
  birthday: "030219",  
  pld: "1234567",  
  fullId: function() {  
    return this.birthday + this.pld;  
  }  
};  
  
person.fullId() // 0302191234567  
person.fullId; // function () { return this.birthday + this.pld; }
```

객체 생성

```
var kitty = {  
  name: "나비",  
  family: "코리안 숏 헤어",  
  age: 1,  
  weight: 0.1  
};  
  
document.write("우리 집 새끼 고양이의 이름은 " + kitty.name + "이고,  
종은 " + kitty.family + "입니다.");
```

프로토타입

```
function Dog(color, name, age) {  
    this.color = color;  
    this.name = name;  
    this.age = age;  
}  
  
var myDog = new Dog("흰색", "마루", 1);  
myDog.family = "시베리안 허스키"; // 품종에 관한 프로퍼티를 추가함.  
myDog.breed = function() {    // 털색을 포함한 품종을 반환해 주는 메소드를 추가함.  
    return this.color + " " + this.family;  
}  
  
document.write("우리 집 강아지는 " + myDog.breed() + "입니다.");
```

우리 집 강아지는 흰색 시베리안 허스키입니다.

EWHA,
THE FUTURE
WE CREATE

String 메소드

메소드	설명
indexOf()	String 인스턴스에서 특정 문자나 문자열이 처음으로 등장하는 위치의 인덱스를 반환함.
charAt()	String 인스턴스에서 전달받은 인덱스에 위치한 문자를 반환함.
substring()	String 인스턴스에서 전달받은 시작 인덱스부터 종료 인덱스 바로 앞까지의 문자열을 추출한 새 문자열을 반환함.
split()	String 인스턴스에서 구분자(separator)를 기준으로 나눈 후, 나뉜 문자열을 하나의 배열로 반환함.
concat()	String 인스턴스에 전달받은 문자열을 결합한 새로운 문자열을 반환함.
toUpperCase()	String 인스턴스의 모든 문자를 대문자로 변환한 새로운 문자열을 반환함.
toLowerCase()	String 인스턴스의 모든 문자를 소문자로 변환한 새로운 문자열을 반환함.
trim()	String 인스턴스의 양 끝에 존재하는 공백과 모든 줄 바꿈 문자(LF, CR 등)를 제거한 새로운 문자열을 반환함.

String 메소드

메소드	설명
replace()	인수로 전달받은 패턴에 맞는 문자열을 대체 문자열로 변환한 새 문자열을 반환함.
match()	인수로 전달받은 정규 표현식에 맞는 문자열을 찾아서 하나의 배열로 반환함.
includes()	인수로 전달받은 문자나 문자열이 포함되어 있는지를 검사한 후 그 결과를 불리언 값으로 반환함.
startsWith()	인수로 전달받은 문자나 문자열로 시작되는지를 검사한 후 그 결과를 불리언 값으로 반환함.
endsWith()	인수로 전달받은 문자나 문자열로 끝나는지를 검사한 후 그 결과를 불리언 값으로 반환함.

String 메소드 예제

```
var str = "abcDEFabc";
```

```
str.indexOf('abc'); //0
```

```
str.charAt(0); // a
```

`str.charAt(10);` // 빈 문자열 -> 전달받은 인덱스가 문자열의 길이보다 클 경우에는 빈 문자열을 반환함.

```
str.substring(2, 6); // cDEF
```

String 메소드 예제

```
var str = "자바스크립트는 너무 멋져요! 그리고 유용해요.";
str.split(); // 구분자를 명시하지 않으면 아무런 동작도 하지 않음.
str.split(""); // 한 문자("")씩 나눔. 자,바,스,크,립,트,는,,너,무,,멋,져,요,!,,,그,리,고,,유,용,해,요,.
str.split(" "); // 띄어쓰기(" ")를 기준으로 나눔. 자바스크립트는 너무,멋져요!,그리고,유용해요.
str.split("!"); // 느낌표("!")를 기준으로 나눔. 자바스크립트는 너무 멋져요, 그리고 유용해요.
str.concat("는 너무 멋져요!"); // 자바스크립트는 너무 멋져요! 그리고 유용해요.는 너무 멋져요!
str.concat("는 너무 멋져요!", " 그리고 유용해요."); // 자바스크립트는 너무 멋져요! 그리고 유용해요.
                             는 너무 멋져요! 그리고 유용해요.
var str = "   JavaScript   ";
str.trim(); //JavaScript
```


Array 메소드

메소드	설명
push()	하나 이상의 요소를 배열의 가장 마지막에 추가하고, 배열의 총 길이를 반환함.
pop()	배열의 가장 마지막 요소를 제거하고, 그 제거된 요소를 반환함.
reverse()	배열 요소의 순서를 전부 반대로 교체함.
sort()	해당 배열의 배열 요소들을 알파벳 순서에 따라 정렬함.
join()	배열의 모든 요소를 하나의 문자열로 반환함.
slice()	전달받은 시작 인덱스부터 종료 인덱스 바로 앞까지의 모든 배열 요소를 추출하여 만든 새로운 배열을 반환함.
concat()	해당 배열의 뒤에 인수로 전달받은 배열을 합쳐서 만든 새로운 배열을 반환함.

Array 메소드 예제

```
var arr = [1, true, "JavaScript"];  
arr.length;           // 3  
arr.push("자바스크립트");  
arr.length;           // 4  
arr;                   // [1,true,JavaScript, 자바스크립트]  
arr.pop(); // 자바스크립트 삭제 (popped)  
arr.reverse(); // [JavaScript,true,1]
```

Array 메소드 예제

var strArr = ["로마", "나라", "감자", "다람쥐"]; // 한글은 ㄱ, ㄴ, ㄷ
순으로 정렬됨.

var numArr = [**10**, **21**, **1**, **2**, **3**]; // 숫자는 각 자릿수 별로
비교된 후 정렬됨.

strArr.sort(); // [감자, 나라, 다람쥐, 로마]

numArr.sort(); // [1, 10, 2, 21, 3]

Array 메소드 예제

```
var arr = [1, true, "JavaScript"];  
arr.join();    // 1,true,JavaScript 문자열 반환  
arr.join(' + '); // 1 + true + JavaScript 문자열 반환  
arr.join(' ');  // 1 true JavaScript 문자열 반환  
arr.join("");   // 1trueJavaScript 문자열 반환  
arr.slice(1, 3); // [true,JavaScript]  
arr.concat("다섯", [6, 7]); // [1,true,JavaScript, 다섯,6,7] -> 값과 배열  
도 한 번에 합칠 수 있음.
```

Event 란?

- 키보드, 마우스, HTML DOM, Window 객체 등과 관련한 이벤트 발생 시 이벤트에 반응하여 특정 동작을 수행
- 예시

```
<p onclick="changeText(this)">이 문자열을 클릭해 보세요!</p>
```

이벤트

...

```
<script>
```

```
function changeText(element) {  
    element.innerHTML = "문자열의 내용이 바뀌었습니다!";  
}
```

```
</script>
```

이벤트 타입

이 문자열을 클릭해 보세요!



이벤트 타입

문자열의 내용이 바뀌었습니다!

Event listener

- 이벤트 발생시 이벤트를 처리하는 함수
- 리스너가 호출되기 위해서는 등록 되어야 함
 - 객체나 요소 (e.g., id)에 프로퍼티로 등록
 - 객체나 요소의 메소드에 리스너 전달

Event listener 등록1: 객체나 요소에 프로퍼티로 등록

- 프로퍼티로 등록

```
window.onload = function()  
{  
    // 이 함수는 HTML 문서가  
    로드될 때 실행됨.
```

```
    var text = document.getElementById("  
text"); // 아이디가 "text" 인 요소를 선택함.
```

```
    text.innerHTML = "HTML 문서가 로드  
되었습니다.";
```

```
}
```

(-) 하나의 리스너만 등록가능

- 태그에 속성으로 등록

```
<p onclick="alert('문자열을 클릭  
했어요!')">이 문자열을 클릭해  
보세요!</p>
```

(-) 유지보수, 가독성

Event listener 등록1: 객체나 요소에 프로퍼티로 등록

- Input 태그 이벤트 핸들러 요소에 등록 (함수 연결)

```
<script type="text/javascript">
function func1() {
  alert("제출버튼 클릭완료.");
}
</script>
```

- `<input type = "button" value="Submit" onClick="func1()">`

Event type	Event handler	Description
Mouse event	onMouseOver	When the mouse enters a particular object
	onMouseDown	When a mouse button is clicked
	onMouseUp	When a mouse button is released
	onClick	When a mouse button is clicked and released within a particular object
	onDbClick	When a mouse is clicked twice rapidly
Keyboard event	onKeyDown	When a keypad is pressed
	onKeyUp	When a keyboard is pressed and released
Other event	onFocus	When an input field in a form element is focused
	onSubmit	When the entered input in a form element is submitted
	onReset	When the entered input in a form element is reset

Event listener 등록2: 객체나 요소의 메소드에 리스너 전달

- 대상객체.addEventListener(이벤트명, 실행할이벤트리스너)

- 예시

```
var btn = document.getElementById("btn");    // 아이디가 "btn"인 요소를 선택함.  
btn.addEventListener("click", clickBtn);      // 선택한 요소에 click 이벤트 리스너를 등록함.  
btn.addEventListener("mouseover", mouseoverBtn); // 선택한 요소에 mouseover 이벤트 리스너를 등록함.  
btn.addEventListener("mouseout", mouseoutBtn); // 선택한 요소에 mouseout 이벤트 리스너를 등록함.  
  
function clickBtn() {  
    document.getElementById("text").innerHTML = "버튼이 클릭됐어요!";  
}  
  
function mouseoverBtn() {  
    document.getElementById("text").innerHTML = "버튼 위에 마우스가 있네요!";  
}  
  
function mouseoutBtn() {  
    document.getElementById("text").innerHTML = "버튼 밖으로 마우스가 나갔어요!";  
}
```

여러개의 리스너 등록가능

Event listener: 리스너 삭제

```
function clickBtn() {  
    btn.removeEventListener("mouseover", mouseoverBtn);  
    btn.removeEventListener("mouseout", mouseoutBtn);  
    document.getElementById("text").innerHTML = "이벤트 리스너  
가 삭제되었습니다!";  
}
```

대화상자로 사용자 입력값 반환

- prompt(): `window.prompt("간단한 메시지" + "입력란의 기본 메시지")`;

- 예시

```
function promptDialogBox() {  
    var inputStr = prompt("당신의 이름을 입력해 주세요 :", "홍길동");  
    if (inputStr != null) {  
        document.getElementById("text").innerHTML = "당신의 이름은 " + inputStr + "입니  
다.";  
    }  
}
```



Javascript 실습

실습시간 공지

- 실습시간: 5-6:15pm
 - 개인실습과제 3까지 마쳤을경우 6시부터는 이른 퇴장 가능합니다
- TO-DO
 - 개인실습과제1: 구구단 자바스크립트 작성하기
 - 개인실습과제2: 아이스크림 자판기 만들기
 - 그룹실습과제3: 팀프로젝트 화면설계

개인실습1: 반복문, 조건문 연습

• 개인실습과제1: 구구단 출력 스크립트 작성해보기

• 요구사항

- 사용자에게 원하는 단을 입력 받는다 (예시: 9단) – 그림1
- 입력 받은 단의 구구단을 옆과 같이 출력한다
- 입력받은 단이 2보다 작거나 9보다 크면 "2-9사이의 숫자를 입력하세요" 대화상자를 띄우고 (그림2) 구구단 출력안함

구구단

9x1=9
9x2=18
9x3=27
9x4=36
9x5=45
9x6=54
9x7=63
9x8=72
9x9=81

힌트: OR = ||

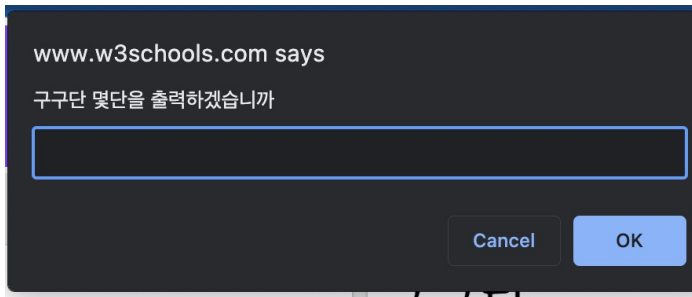


그림1

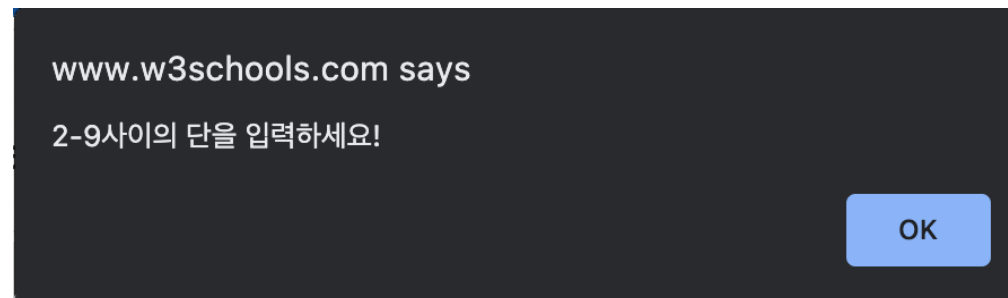


그림2

개인실습과제1: 구구단 출력 스크립트 작성해보기

- 파일명: practice1_학번.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>구구단</h1>
```

```
<p>아래버튼을 클릭하여 원하는 단을 입력하세요</p>
```

```
<button onclick="myFunction()">Click</button>
```

```
<script>
```

```
function myFunction() {
```

```
    //프롬프트 사용하여 사용자 인풋을 받음 "구구단 몇단을 출력하겠습니까 " 문구 보여줌
```

```
    //입력받은 값이 2보다 작거나 9보다 크면 "2-9사이의 단을 입력하세요!" alert 띄움
```

```
    //입력받은 값이 2-9 사이이면 구구단 출력
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

구구단

아래버튼을 클릭하여 원하는 단을 입력하세요

Click

개인실습2: 리스너등록 연습

• 개인실습과제2: 아이스크림 자판기

• 요구사항

- 금액충전기능
- 아이스크림 선택기능
- 선택한 아이스크림 표시
- 현재 잔액 Balance에 표시
- Balance에 따라 선택가능한 아이스크림 버튼은 파란색표시, 불가능한 아이스크림 빨간색표시

Icecream Vending Machine

아이스크림의 구매를 위해 아래 버튼을 눌러 금액을 충전해주세요

1000원

500원

100원

원하는 아이스크림의 버튼을 선택하여 주세요

돼지바(700원)

월드콘(1000원)

누크바(500원)

Choice : 아이스크림을 선택해주세요

Balance : 0원

개인실습2 step-by-step instruction

Step1. 사이버캠퍼스에서 practice2.html 다운로드

Step2. html 파일 경로에 practice2.js 파일 생성

Step3. practice2.js 파일에 작업 시작

- 초기화면 셋업
 - Balance 에 사용할 변수선언
 - 초기문구 보여주기

```
var coin = 0;
document.getElementById("current").innerHTML = coin
+ "원";
document.getElementById("selection").innerHTML = "아
이스크림을 선택해주세요";
```

Icecream Vending Machine

아이스크림의 구매를 위해 아래 버튼을 눌러 금액을 충전해주세요

1000원 500원 100원

원하는 아이스크림의 버튼을 선택하여 주세요

돼지바(700원) 월드콘(1000원) 누크바(500원)

Choice : 아이스크림을 선택해주세요

Balance : 0원

개인실습2 step-by-step instruction

Step3. practice2.js 파일에 작업 시작

- 각 버튼에 연결시킬 리스너 함수 생성

1000원

500원

100원

돼지바(700원)

월드콘(1000원)

누크바(500원)

```
//1000원 버튼 클릭시 Balance에 1000원 추가되어야 함
function click_btn1() {
  coin = coin + 1000;
  color_change();
  alert("1000원 넣음");
  document.getElementById("current").innerHTML = coin + "
원";
}
```

개인실습2 step-by-step instruction

- 각 버튼에 연결시킬 리스너 함수 생성

1000원

500원

100원

돼지바(700원)

월드콘(1000원)

누크바(500원)

//돼지바(700원) 버튼 클릭시 Balance에 -700원 되고 Choice에 돼지바 나타나야함

```
function click_icecream1() {  
    coin -= 700;  
    document.getElementById("selection").innerHTML = "돼지바";  
    color_change();  
    document.getElementById("current").innerHTML = coin + "원";  
}
```

단, 돈이 부족할 경우
"돈이 부족합니다"
대화상자 띄움
(e.g.,
alert("돈이 부족합니다
");)

개인실습2 step-by-step instruction

- Balance에 따라 아이스크림 선택버튼 폰트 색 변경을 위한 함수생성 (e.g., `color_change()`)

돼지바(700원) 월드콘(1000원) 누크바(500원)

Choice : 아이스크림을 선택해주세요

Balance : 0원

최초화면 Balance 0원이므로 모든 버튼 빨간색 폰트

돼지바(700원) 월드콘(1000원) 누크바(500원)

Choice : 아이스크림을 선택해주세요

Balance : 1000원

Balance 1000원이므로 모든 버튼 파란색 폰트

돼지바(700원) 월드콘(1000원) 누크바(500원)

Choice : 돼지바

Balance : 800원

Balance 800원이므로 월드콘 버튼 빨간색 폰트

```
document.getElementById("icecream1").style.color = "blue";  
document.getElementById("icecream2").style.color = "red";  
document.getElementById("icecream3").style.color = "blue";
```

개인실습2 step-by-step instruction

- 리스너 등록 (리스너 함수-버튼 연결)



1000원 버튼에 해당하는 리스너를 slide6에서 생성한 함수를 사용하여 등록해보자

```
const btn1 = document.getElementById("btn1");  
btn1.addEventListener("click", click_btn1);
```

개인실습2 step-by-step instruction

Step4. Step1-3의 구현이 끝났다면 practice2.html 브라우저에서 열어보고 아래의 시나리오로 테스트해보자

- 체크포인트
 - 초기화면 체크
 - 금액충전시 balance에 반영 및 아이스크림 버튼 폰트색 변화 체크
 - 아이스크림 선택시 balance에 반영 및 아이스크림 버튼 폰트색 변화 체크
- 테스트 시나리오

1. 1500원 충전후 돼지바 구매

돼지바(700원) 월드콘(1000원) 누크바(500원)

Choice : 돼지바

Balance : 800원

2. 500원 충전

돼지바(700원) 월드콘(1000원) 누크바(500원)

Choice : 아이스크림을 선택해주세요

Balance : 500원

3. 500원 충전후 누크바 구매

돼지바(700원) 월드콘(1000원) 누크바(500원)

Choice : 누크바

Balance : 0원



팀프로젝트 공지

이화 마켓 웹어플리케이션

- 요구되는 웹페이지 화면리스트

1. 상품 등록 화면
2. 상품 전체조회 화면
3. 상품 세부화면
4. 상품 리뷰 작성 화면
5. 상품 리뷰 전체조회 화면
6. 상품 리뷰 상세 화면
7. 로그인/로그아웃 화면
8. 회원가입
9. 그외에 차별성을 가진 우리 팀만의 화면(들)

1. 상품등록화면

- 아래 정보를 입력받는다
 - 판매자 아이디
 - 상품명
 - 가격
 - 지역
 - 상태
 - 설명
 - 외부 혹은 내부 대표사진
 - 이외 원하는정보 입력가능

2. 상품 전체조회 화면

- 등록된 상품들 리스팅
 - 사진, 상품명으로 리스팅
- 페이지네이션 사용 (웹페이지 하단에 페이지 번호 추가)
- 각 상품 클릭시 상품 세부화면으로 이동

3. 상품 세부화면

- 상품 등록화면에서 등록한 내용 보여줌
- ~~상품 리뷰조회 화면 링크연결(기능삭제)~~

4. 상품 리뷰 작성화면

- 아래 정보를 입력받는다
 - 리뷰자 아이디
 - 상품 이름
 - 리뷰제목
 - 리뷰사진
 - 리뷰글
 - 별점
 - 이외에 원하는 정보 입력가능

5. 상품 리뷰 전체조회 화면

- 등록된 리뷰들이 리스트업
 - 리뷰사진, 리뷰제목 으로 리스트업
- 페이지네이션 사용 (웹페이지 하단에 페이지 번호 추가)
- 리뷰 클릭시 상품 리뷰 상세화면으로 이동

6. 상품 리뷰 상세 화면

- 상품 리뷰 작성화면에서 등록된 내용 보여줌

7. 회원가입

- 아래 정보를 입력받는다
 - (필수) 아이디
 - (필수) 비밀번호
 - (필수) 이메일 주소
 - (선택) 핸드폰 번호
 - 이외 원하는 정보 입력가능
- 아이디 중복체크 기능
- 비밀번호는 Hash 값으로 저장

8. 로그인/로그아웃

- 아이디, 비밀번호로 로그인
- 비밀번호는 해시값 생성하여 DB의 비밀번호 Hash 와 비교
- 로그인시 로그아웃 버튼 활성화 되게함

그룹실습1: 팀프로젝트 화면설계 (Part1)

- Hand-writing 혹은 원하는 그래픽 툴을 사용하여 이화마켓 웹 어플리케이션 화면설계
 - 지난시간까지 배운 html, css 를 생각하며 화면 그려보기
 - 그래픽 툴을 사용 할 경우 인터랙션 적용할 필요 없음
 - 데드라인: 10/18 WED 23:59 (Note that 10/19은 프로포절 발표!)

그룹실습2: 팀프로젝트 화면설계 (Part2)

- 설계한 화면을 바탕으로 HTML 및 CSS 생성
 - 데드라인: 11/9 THU 23:59

Coming up

- Summary of TO-DO
 - 개인실습과제 1,2 사이버캠퍼스 제출
 - 팀프로젝트 화면설계 (Part1) 사이버캠퍼스 제출 (10/18)
 - 팀프로젝트 화면설계 (Part2) 사이버캠퍼스 제출 (11/9)
- 다음주 JavaScript (Part2) 수업합니다