

JavaScript 기초과정

-1-

곽정애
jeongae716@gmail.com
ver 1.2

사용할 프로그램

- **Aptana**
 - <http://www.apтана.com/products/studio3/download.html>
 - 본인이 원하는 에디터를 사용하셔도 무관합니다.
- **Web Browser**
 - Chrome (필수)
 - Internet Explorer



HTML, CSS, JavaScript의 관계

HTML

보여줄 내용

CSS

스타일 설정
(디자인)

JavaScript

동작, 컨트롤

프로그래밍 언어

인터프리터 방식 언어

클라이언트 측 스크립트

Javascript 버전 및 크로스브라우징

JavaScript와 웹 접근성

JavaScript vs JAVA

태그

- `<script>`
 `// 코드`
 `</script>`

삽입 위치

- `<head> </head>` 내부
- `<body> </body>` 내부
- 외부 스크립트 링크 시 : `<script src="myScript.js"> </script>`

그 외 (이러지 맙시다!! >_<)

- `<button type="button" onclick="javascript:myFn();">Click Me!</button>`



JavaScript 삽입 (실무 예제)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, width=device-width, user-scalable=no">
  <meta name="format-detection" content="telephone=no">
  <title>sample</title>
  <link rel="stylesheet" type="text/css" href="../common/css/common.css">
  <link rel="stylesheet" type="text/css" href="../common/css/content.css">
  <script type="text/javascript" src="../common/js/libs/jquery-1.9.1.min.js"></script>
  <script type="text/javascript" src="../common/js/libs/jquery.hammer.min.js"></script>
  <script type="text/javascript" src="../common/js/libs/jquery.easing-1.3.min.js"></script>
  <script type="text/javascript" src="../common/js/libs/iscroll.js"></script>
  <script type="text/javascript" src="../common/js/common.js"></script>
</head>
<body>
  <div id="wrapper">
    <!-- contents -->
    <section id="container">
      <div id="contents">

      </div>
    </section>
    <!-- //contents -->
  </div>

  <script type="text/javascript" src="../common/js/page/test.js"></script>
  <script>
    (function(){
      })();
  </script>
</body>
</html>
```

→ 공용 스크립트 선언

→ 해당 페이지용 스크립트 선언

대소문자 구분

```
var test = "Hello world";  
alert(Test);
```

세미콜론

```
var test = "Hello world";  
alert(test);
```

한 줄 주석

```
// 한 줄 주석입니다.
```

여러 줄 주석

```
/*  
    이 주석은  
    여러 줄로 작성할 때  
    사용합니다.  
*/
```

```
<!doctype html>
<html lang="ko">
<head>
  <meta charset="utf-8">
  <title>Hello world</title>
</head>

<body>
  <script type="text/javascript">
    alert("Hello world! :: alert");
    // document.write("Hello world! :: document");
    // console.log("Hello world! :: console");
  </script>
</body>
</html>
```

여기서 잠깐!

console.log()

w3schools

- Tutorial : <http://www.w3schools.com/js/default.asp>
- Reference : <http://www.w3schools.com/jsref/default.asp>

생활코딩

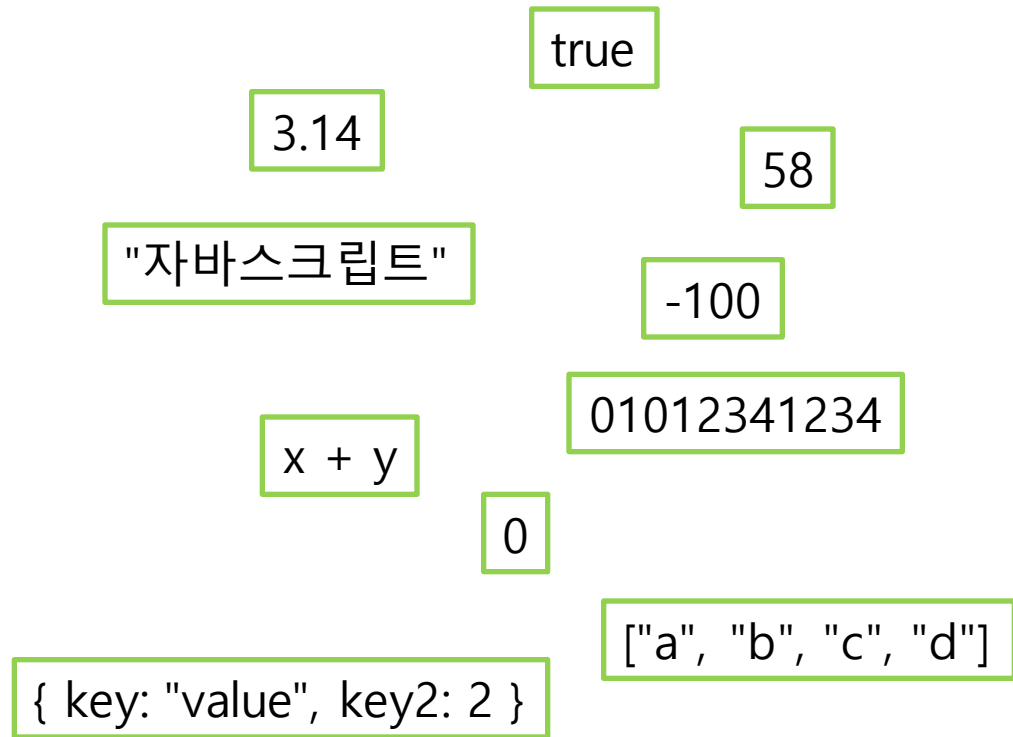
- <https://opentutorials.org/course/743>
- <https://opentutorials.org/course/1375>

코드스쿨

- <https://www.codeschool.com/>

변수(Variable)

데이터를 담는 그릇



변수(variable)

선언 방법

- 선언과 동시에 값 대입
var 변수명 = 값;

```
var num = 123;
```

- 먼저 선언. 그 후 값 대입
var 변수명;
변수명 = 값;

```
var num;  
num = 123;
```

여러 변수를 동시에 선언하려면

```
var num = 123,  
    str = "javascript & jquery",  
    bool = true,  
    myName;
```

```
var num, str, bool, myName;
```

여기서 잠깐!

var 없이 변수를 선언하고
값을 주면 어떻게 될까요?

대소문자 구분

영문, 숫자 및 특수문자(언더바(_)와 달러 기호(\$))만 사용 가능

첫 문자에는 숫자 사용 불가

예약어 사용 불가

- 자바스크립트에서 사용하고 있는 혹은, 사용될 예정인 이름

```
abstract boolean break byte case catch char class const continue debugger default delete do  
double else enum export extends false final finally float for function goto if implements import in  
instanceof int interface long native new null package private protected public return short static  
super switch synchronized this throw throws transient try typeof var volatile void while with
```

여기서 잠깐!

javascript의 일반적인 변수 선언 방식은?

데이터 타입(Data Type)

Number

```
var num = 23;
```

String

```
var str = "Hello~!!";
```

Boolean

```
var bool = false;
```

Array

```
var arr = ["java", "JavaScript", "c", "c++"];
```

Object

```
var sherlock = {  
  fullName: "Sherlock Holmes",  
  birthYear: 1854  
}
```

undefined

undefined vs null

여기서 잠깐!

데이터 타입을 비교하려면 `typeof` 를 사용합니다.
예: `typeof num;`

암시적(묵시적) 변환

```
var num = 23;
var str = "Hello~!!";
var str1 = "7";
var bool = true;

var numNum = num + num;
var numStr = num + str;
var numBool = num + bool;
var boolStr = bool + str;
var boolBool = bool + bool;

var numStrBool = num + str + bool;
var numBoolStr = num + bool + str;

var numStr1Multi = num * str1;
var numBoolMulti = num * bool;
var numStrMulti = num * str;
```

```
console.log(numNum, typeof numNum);
console.log(numStr, typeof numStr);
console.log(numBool, typeof numBool);
console.log(boolStr, typeof boolStr);
console.log(boolBool, typeof boolBool);

console.log(numStrBool, typeof numStrBool);
console.log(numBoolStr, typeof numBoolStr);

console.log(numStr1Multi, typeof numStr1Multi);
console.log(numBoolMulti, typeof numBoolMulti);
console.log(numStrMulti, typeof numStrMulti);
```


명시적 변환

```
var num = 23;  
var str = "7";  
var bool = true;
```

```
var numToStr = String(num);  
var numToStr1 = "" + num;  
var numToStr2 = num.toString();  
  
var boolToNum = Number(bool);
```

```
var strToNum = Number(str);  
var strToNum1 = parseInt(str);
```

```
console.log(numToStr, typeof numToStr);  
console.log(numToStr1, typeof numToStr1);  
console.log(numToStr2, typeof numToStr2);
```

```
console.log(boolToNum, typeof boolToNum);
```

```
console.log(strToNum, typeof strToNum);  
console.log(strToNum1, typeof strToNum1);
```

숫자(Number)

```
var intNum = 23;
var intNum1 = 2;
var floatNum = 0.05;
var floatNum1 = 0.01;
var numMax = Number.MAX_VALUE;

var sum = intNum + floatNum;
var sum1 = intNum + floatNum * intNum1;
var sum2 = (intNum + floatNum) * intNum1;
var sum3 = floatNum + floatNum1;

console.log(sum, typeof sum);
console.log(sum1);
console.log(sum2);
console.log(sum3);
console.log(numMax);
```

여기서 잠깐!

소수점 연산 오류를 주의하세요.



진수 표기 (10진수, 8진수, 16진수)

10진수 : 숫자만 쓴다.

8진수 : 숫자 앞에 숫자 0을 붙인다.

16진수 : 숫자 앞에 숫자 0과 영어 소문자 x를 붙인다.

```
var num = 12;  
var numOct = 012;  
var numHex = 0x12;  
  
console.log(num);  
console.log(numOct);  
console.log(numHex);  
  
var phoneNumber = 01011112222;  
console.log(phoneNumber);
```

숫자 객체 전용

소수점 자릿수 지정할 때 사용하며, 문자열을 반환

```
var floatNum = 0.05;
var floatNum1 = 0.01;

var sum = floatNum + floatNum1;
var sumToFixed = sum.toFixed(2);

console.log(sum);
console.log(sumToFixed, typeof sumToFixed);
```

변수 sumToFixed의 값을 숫자(Number) 타입으로 만들어 보세요.



parseInt(), parseFloat()

문자열을 숫자로 변환

```
var str = "3.14";  
var num = parseInt(str);  
var num1 = parseFloat(str);  
  
console.log(num);  
console.log(num1);  
  
// 숫자가 아닌 문자열이 있을 경우  
var str1 = "3.14 PIE";  
console.log(parseInt(str1), parseFloat(str1));  
  
// 숫자가 뒤쪽에 있을 경우  
console.log(parseInt("I like 7"));
```

수학(Math)

수학 연산에 관련된 객체 (http://www.w3schools.com/jsref/jsref_obj_math.asp)

```
var num = 3.1;  
var num1 = 3.6;  
  
// 반올림, 버림, 올림  
console.log(Math.round(num));  
console.log(Math.round(num1));  
console.log(Math.floor(num1));  
console.log(Math.ceil(num));
```


수학 연산에 관련된 객체 (http://www.w3schools.com/jsref/jsref_obj_math.asp)

```
// 랜덤 (0~1 사이의 난수 값 : 0 포함, 1 미포함)
var randomNumber = Math.random();
console.log(randomNumber);

// 랜덤 (0~100 사이의 난수 값)
console.log(Math.floor(randomNumber * 100));
```

주사위 눈처럼 1 ~ 6 사이의 랜덤한 숫자를 출력해 보세요.

문자열(String)

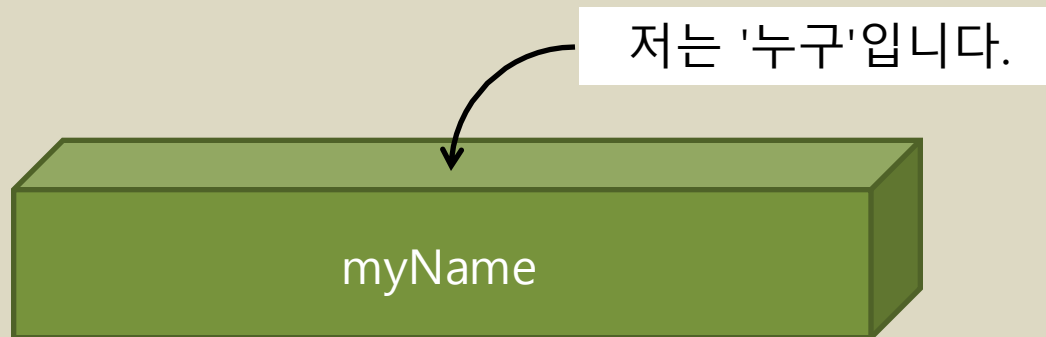
"문자열을 표현합니다."

'이것도 문자열을 표현합니다.'

"저는 '누구'입니다."

"'Javascript 기초과정'에 참석해주셔서 감사합니다.'

```
var myName = "저는 '누구'입니다.";
```



이중 인용부호 사용, 또는 이스케이프 처리 시 문자 앞에 역슬래시(\) 추가

이스케이프 시퀀스	기능
\t	탭
\n	줄 바꿈
\b	백스페이스
\"	" 표시
\'	' 표시
\\	\ 표시
\uNNNN(N은 0~9까지의 수치 또는 A(a)~F(f))	16진수로 된 유니코드 문자

```
var str = "Front-end Developer\t\t\"WebPublisher\"\nDesigner";
```

```
var str1 = 'It\'s alright!';
```



문자열 객체의 인스턴스 생성

new 연산자 사용하여 생성

```
// strObj라는 이름의 인스턴스 생성  
// 인스턴스를 통해 String 객체의 속성 또는 메서드 사용 가능  
var strObj = new String("JavaScript");  
  
// javascript라는 문자열이 newStrObj에 대입  
var newStrObj = strObj.toLowerCase();
```

new 연산자를 사용하지 않고 생성

- 문자열 리터럴이 생성된다.
- 문자열 리터럴은 String 객체의 인스턴스와 마찬가지로 String 객체의 속성 또는 메서드를 사용할 수 있다.

```
var strLiteral = "String Literal";  
// var strLiteral = String("String Literal");  
var newStrLiteral = strLiteral.toLowerCase();
```

여기서 잠깐!

JavaScript에서는 주로 리터럴 형태로 선언합니다.

더하기(+) 연산자 사용

- 휴대폰 번호를 하나의 변수에 담아보자.

```
var phoneNum1 = 휴대폰 번호 처음 세자리,  
    phoneNum2 = 휴대폰 번호 중간 네자리,  
    phoneNum3 = 휴대폰 번호 마지막 네자리,  
    myPhoneNumber = phoneNum1 + phoneNum2 + phoneNum3;  
  
console.log(myPhoneNumber); // 브라우저 콘솔창에 myPhoneNumber값 출력
```

더하기(+) 연산자 사용

- 문자열과 문자열이 아닌 데이터 타입을 더해보자.

```
// 문자열 + 숫자
var grade = 100;
var gradeTotal = "성적:" + grade;
console.log(gradeTotal, typeof gradeTotal);

// 숫자 + 문자열
var value = 10.0;
var valueTotal = value + "점";
console.log(valueTotal, typeof valueTotal);

// 문자열 + 논리
var bool = true;
var boolTotal = "값:" + bool;
console.log(boolTotal, typeof boolTotal);
```


문자열 길이, 인덱스에 해당하는 문자열 찾기

- length
- charAt(*index*)

```
var str = "Hello world!!";  
  
console.log(str.length);  
console.log(str.charAt(0), str.charAt(6));
```

검색하려는 문자열이 존재 하지 않을 경우 -1을 반환

- `indexOf(searchValue)`
- `indexOf(searchValue, start)`
- `lastIndexOf(searchValue)`
- `lastIndexOf(searchValue, start)`

```
var str = "I like apple, banana, pineapple, mangosteen~!";  
  
console.log(str.indexOf("apple"));  
console.log(str.indexOf("apple", 13));  
console.log(str.lastIndexOf("apple"));  
console.log(str.lastIndexOf("apple", 6));
```

I	l	i	k	e	,	a	p	p	l	e	,	b	a	n	a	n	a	,	p	i	n	e	a	p	p	l	e	,	m	a	n	g	o	s	t	e	e	n	~	!
0		5		10		15		20		25		30		35		40		44																						



문자열 일부분 추출하기

- `substr(start, length)`
 - http://www.w3schools.com/jsref/jsref_substr.asp
- `substring(start, end)`
 - http://www.w3schools.com/jsref/jsref_substring.asp

```
var str = "I like apple, banana, pineapple, mangosteen~!";

console.log(str.substr(7));
console.log(str.substring(7));
console.log(str.substr(7, 13));
console.log(str.substring(7, 13));
```

I like apple, banana, pineapple, mangosteen~!

7 12 19

- toUpperCase()
- toLowerCase()

```
var str = "I like apple.";

console.log(str.toUpperCase());
console.log(str.toLowerCase());
```

구분자를 기준으로 문자열을 분할하여 배열에 담아 반환

- `split(separator, limit)`

```
var str = "I like apple, banana, pineapple, mangosteen~!";  
var strArr = str.split(",");  
var strArr1 = str.split(",", 1);  
  
console.log(strArr);  
console.log(strArr1);  
console.log(str);
```

정규표현식(RegExp)



정규표현식(RegExp)

참고!

이런 것도 있어요. 가볍게 읽고 넘어가세요.

특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용하는 형식 언어 (regular expression)

- http://www.w3schools.com/js/js_regexp.asp
- http://www.w3schools.com/jsref/jsref_obj_regexp.asp

```
var email = "admin@monoraum.com";  
var regExp = /^([\w-]+(?:\.[\w-]+)*)@((?![\w-]+\.)*\w[\w-]{0,66})\.([a-z]{  
2,6}(?:\.[a-z]{2})?)$/;  
  
console.log(regExp.test(email));
```

```
var userAgent = "Mozilla/5.0 (iPhone; CPU iPhone OS 8_0 like Mac OS X)  
AppleWebKit/600.1.3 (KHTML, like Gecko) Version/8.0 Mobile/12A4345d Safari/600.1.4";  
var regExp = /iPhone|iPad|iPod/i;  
var isIOS = regExp.test(userAgent);  
  
console.log(isIOS);
```