

JavaScript & Jquery 02

by withsky

1 사용법

internal 방식

```
<> test.html x <> index.html x
1
2 <script>
3   alert("Hello World!");
4 </script>
```

external 방식

```
<> index.html x JS test.js
1
2 <script src="test.js"></script>
3
```

```
<> index.html JS test.js
1
2 alert("Hello World2!");
```

script 코드의 위치는 어디가 좋은가?

- 이론상으로는 화면 구현 후 js코드(소스)를 읽어 들일 수 있도록 문서 하단 위치
- 퍼포먼스가 높다고 하지만 최신 엔진의 고성능화로 사람이 느낄 수 있는 차이는 아님.

2 변수 Variable

데이터를 저장하는 장소 [데이터를 저장한 주소 위치를 저장한 이름]
데이터를 읽고 쓸수 있는 장소
지속적으로 사용할 수 있도록 메모리에 일시적으로 저장하는 역할
다른 언어와는 다르게 변수 선언은 var로 통일

변수 선언 방법

```
var 변수명 = 값;
```

```
var 변수명;  
변수명 = 값;
```

변수명 = 값; // 초기화 없이 바로 쓸 수는 있으나 조심

```
<> index.html <> 변수.html x  
1  <script>  
2  //변수 선언 후 사용  
3  var string;  
4  string = '안녕하세요.';  
5  document.write(string);  
6  
7  // 변수 초기화 수행  
8  var string2 = '저는 서민구입니다.';  
9  document.write(string2);  
10  
11 // 숫자 문자 구분 없이 사용 가능  
12 var number =2;  
13 document.write(number);  
14 </script>
```

변수값 변경

```
var 변수명 = 값;  
변수명 = 새로운값;
```

저장 할 수 있는 데이터

숫자(정수, 실수)
문자열
함수
객체(인스턴스)

3 자료형

기본형

- 숫자형(Number)
- 문자형(String) : "대한민국" / '대한민국'
- 논리형(Boolean) : true / false 둘중 하나
- undefined = 없는값 (값이 미정의된 값)
- null = 없는값 (값이 미정의된 값)

```
<> 기본형.html x
1  <script>
2
3  // 숫자형
4  var num = 1;
5  document.write(typeof(1));
6
7  // 문자형
8  var str = "문자";
9  document.write(typeof(str));
10
11 // 논리형
12 var bool = false;
13 document.write(typeof(bool));
14
15 // 미정의된 값
16 var non;
17 document.write(non);
18
19 </script>
```

참조형

- 배열(Array) : 데이터 집합
- 함수(Function) - 구문의 집합
- 객체(Object) - 데이터 + 함수 집합

3 자료형

배열 (Array)

데이터의 집합으로, 복수의 값을 보관할 수 있다.

인덱스(index)로 접근 가능

인덱스는 0부터 시작

문자, 숫자 혼용 가능

```
<> 배열.html x
1  <script>
2  // 배열(문자)
3  var arr = ["apple", "grape", "banana"];
4  document.write(arr[0]);
5
6  // 배열(숫자)
7  var arr = [1, 2, 3];
8  document.write(arr[0]);
9
10 // 문자, 숫자 혼용
11 var arr1 = ["apple", 1, "banana"];
12 var arr2 = ["melon", 2, "grape"];
13 document.write(arr1[1]+arr2[1]);
14
15 </script>
```

3 자료형

함수 (Function)

변수와 구문의 집합
일련의 처리 절차(기능)을 하는 집합

```
<> 함수.html x
1 <script>
2 // 함수
3 function sum(a, b) {
4     var s = a+b;
5     return s;
6 }
7
8 document.write(sum(1,2));
9 </script>
```

객체 (Object)

데이터와 함수의 집합
키값으로 접근

```
<> 객체.html x
1 <script>
2 // 객체 Object
3 var obj = {a:1, b:2, c:3};
4 document.write(obj.a);
5 document.write(obj['a']);
6 </script>
```

4 변수의 종류

지역변수 (Local)

함수내에서만 쓰이는 변수
함수안에서 만들어지고 함수 안에서만 사용
함수가 종료되면 변수도 소멸

```
<> 지역변수.html x
1  <script>
2      // 지역변수
3      function test() {
4          var a = 1;
5      }
6      document.write(a);
7
8  </script>
```

DevTools - file:///D:/javascript/ex1/%EC%A7%80%EC%97%AD%EB%B3%8

Elements Console Sources Network Performance

top Filter

✖ Uncaught ReferenceError: a is not defined
at 지역변수.html:6

전역변수 (Global)

함수 외부에 선언된 변수
함수 내/외 어디서나 쓸 수 있는 변수

```
<> 전역변수.html x
1  <script>
2      // 전역변수
3      var b = 2;
4      function test2() {
5          b = 3;
6      }
7      test2();
8      document.write(b);
9  </script>
```

4 변수의 종류

매개변수 (Parameter)

선언된 함수 내부로 값을 전달해주는 변수
매개변수도 지역변수이므로 함수가 종료되면 소멸

멤버변수 (Member)

클래스 내부에 선언된 변수
클래스가 인스턴스(객체)화 되어야 접근 가능

변수 선언 규칙

숫자로 시작할 수 없음.
첫단어는 (영문, _, \$)
대/소문자 구분
예약어 사용 금지
공백 불가

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

5 변수 명명규칙

이름만 보고도 쉽게 유추할 수 있어야 하며,
누구나 알아보기 쉽게 지어져야 함.
전체 파일에 통일된 방법으로 작성.
영단어로 작성할 것.

함수명

camelCase 기법
예) myFunction

클래스

Pascal 기법
예) MyClass

변수

underscore 기법
예) my_variable

6 문자열 변수 선언

문자열 변수 선언시

"(큰따옴표), '(작은 따옴표) 모두 사용 가능

```
<> 기타.html ●  
1 <script>  
2   var a = "테스트";  
3   var b = '테스트';  
4  
5 </script>
```

문자열 값에 ' 가 들어가야 하는 경우

외부에는 "로 감싸면 안에 '를 쓸 수 있음.
escape 문자를 사용한다.

```
6   var c = "I'm a boy";  
7   var d = 'I\'m a boy';  
8
```