

웹 표준에 맞는 HTML5 프로그래밍 강의 노트

제 15회차
JavaScript 기초

■ 학습목표

- JavaScript의 기본 용어와 개념을 설명할 수 있다.
- JavaScript의 기본 문법을 설명할 수 있다.

■ 학습내용

- JavaScript의 개요
- JavaScript의 기본 문법

1. JavaScript의 개요

1) JavaScript의 개요

- 웹 브라우저에서 많이 사용하는 프로그래밍 언어

- HTML 문서를 처리할 수 있는 응용 프로그램
- Script를 해석할 수 있는 엔진을 가짐

- 넷스케이프의 Brendan Eich에 의해 '모카'라는 이름으로 만들어짐

- 썬 마이크로시스템과 공동 개발하게 되면서 JavaScript라는 명칭으로 변경됨(Javascript와 Java는 다른 언어임)

2) JavaScript의 특징 및 구성요소

- JavaScript의 특징

- HTML 문서와 함께 처리되므로, 일반 프로그래밍 언어보다는 기능이 제한적임
- 객체 기반의 언어
- 이벤트 중심의 프로그래밍 언어
- 인터프리터 언어로서 클라이언트의 웹 브라우저에 의해 해석되고 실행
- HTML 문서 내에 기술되고 HTML 문서와 함께 수행
- HTML에 연산, 제어 등 프로그래밍적인 요소를 구축하고 클라이언트 자원을 활용

- JavaScript의 구성요소 : 객체, 문법, 이벤트와 이벤트 핸들러

- 객체

- 속성과 메소드를 가짐
- JavaScript 객체의 종류
- 사용자 정의 객체
- 내장 객체
- 브라우저 지원을 이용하는 브라우저 객체

1. JavaScript의 개요

2) JavaScript의 특징 및 구성요소

- 문법 : 자료형, 변수, 제어문, 함수
- 이벤트와 이벤트 핸들러

- 이벤트 핸들러와 함께 JavaScript를 사용하면 상호 작용이 있는 웹 페이지를 만들 수 있음

- JavaScript를 활용한 실습 예제

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript 실습하기!!</title>

<script>

                alert('Hello World!!!');

</script>

</head>

<body>

</body>
</html>
```

2. JavaScript의 기본 문법

1) 기본 용어

- 표현식 : 값을 만들어내는 간단한 코드

```
253
50 + 47 - 23 * 5
'JavaScript'
```

- 문장 : 하나 이상의 표현식이 모여 '문장'이 됨

- 문장 끝에는 세미콜론을 찍어 문장의 종결을 알려줌

```
253;
50 + 47 - 23 * 5;
var string = 'Java' + 'Script';
Alert('Hello world');
```

- 키워드 : 자바스크립트가 처음 만들어질 때 정해진 특별한 의미가 있는 단어

- break, case, catch, continue, default, delete, do, else, false, finally, for, function, if, in, instanceof, new, null, return, switch, this, throw, true, try 등
- W3C는 자바스크립트 프로그램 작성 시 이 키워드를 사용하지 않기를 권고함

- 식별자 : 자바 스크립트에서 이름을 붙일 때 사용하는 단어

- 식별자를 만들 때 반드시 지켜야 하는 규칙
 - 키워드를 사용하면 안됨
 - 숫자로 시작하면 안됨
 - 특수 문자는 _와 \$만 허용함
 - 공백 문자를 포함할 수 없음
- 대부분의 자바스크립트 개발자가 지키는 관례
 - 생성자 함수의 이름은 항상 대문자로 시작함
 - 변수와 인스턴스, 함수, 메소드의 이름은 항상 소문자로 시작함
 - 여러 단어로 이루어진 식별자는 각 단어의 첫 글자를 대문자로 함

2. JavaScript의 기본 문법

2) 변수

- 변수 : 값을 저장할 때 사용하는 식별자

- 변수의 선언 후 변수의 값을 할당해야 함

```
<script>
  // 변수 선언
  var age;
  // 변수에 값 할당
  age = 26;
  // 출력
  alert(age);
</script>
}
```

- 변수를 활용한 실습 예제

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>변수 사용하기!!</title>

<script>

    var radius = 20;
    var pi = 3.14;
    alert(2 * pi * radius);

</script>

</head>

<body>

</body>
</html>
```

2. JavaScript의 기본 문법

3) 연산자

- 사칙 연산자

연산자	설명	연산자	설명
+	더하기 연산자	*	곱하기 연산자
-	빼기 연산자	/	나누기 연산자

- 비교 연산자 : 좌변과 우변을 비교하는 연산자

연산자	설명
>=	좌변이 우변보다 크거나 같음
<=	우변이 좌변보다 크거나 같음
>	좌변이 큼
<	우변이 큼
==	좌변과 우변이 같음
!=	좌변과 우변이 다름

- 논리 연산자

연산자	설명
!	논리 부정 연산자
&&	논리곱 연산자
	논리합 연산자

<논리곱 연산자>

<논리합 연산자>

좌변	우변	결과	좌변	우변	결과
true	true	true	true	true	true
true	false	false	true	false	true
false	true	false	false	true	true
false	false	false	false	false	false

2. JavaScript의 기본 문법

3) 연산자

- 복합 대입 연산자

연산자	설명
+=	기존 변수의 값에 값을 더함
-=	기존 변수의 값에 값을 뺌
*=	기존 변수의 값에 값을 곱함
/=	기존 변수의 값에 값을 나눔
%=	기존 변수의 값에 나머지를 구함

- 증감 연산자 : 전위, 후위에 따라 값이 다르니 증감연산자를 사용할 때는 항상 주의해야 함

연산자	설명
변수++	기존의 변수 값에 1을 더함(후위)
++변수	기존의 변수 값에 1을 더함(전위)
변수--	기존의 변수 값에 1을 뺌(후위)
--변수	기존의 변수 값에 1을 뺌(전위)

2. JavaScript의 기본 문법

3) 연산자

- 연산자를 활용한 실습 예제

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>연산자 실습하기!!</title>

<script>

    alert('사칙 연산자');
    alert(5+2*9);

    alert('비교 연산자');
    alert(4 < 2);

    alert('논리 연산자');
    alert(30<10 && 20<40);

    alert('복합 대입 연산자');
    var a = 10;
    a *= 3;
    alert(a);

    alert('증감 연산자');
    var b = 11;
    b++;
    alert(b);
    b--;
    alert(b);

</script>

</head>

<body>

</body>
</html>
```

2. JavaScript의 기본 문법

4) 자료형

- 자료형 : 변수에 사용되는 값
- 자료형의 종류 : 정수형, 부동소수점, Boolean, Null, 문자열
- 정수형 : 10진수, 16진수, 8진수의 3가지 형태

진수	예문	설명
10진수 10	10	일반적인 숫자형태
16진수 10	0xA	0x를 숫자앞에 붙여 0~9, A~F까지 문자로 표현
8진수 10	012	0~7까지의 숫자만 사용

```
var intNum = 200;
```

- 부동소수점 : 소수점을 가지고 있는 숫자(실수형)

실수	예문
소수형 실수	5.7893
지수형 실수	0.9876E+3

```
var dbNum = 56.42;
```

- Boolean : 크기가 1비트인 자료형으로 주로 함수의 리턴 값이나 관계 연산자의 결과값에 사용

진수	설명
True	참
False	거짓

```
var sw1 = true;(참)
var sw2 = false;(거짓)
```

- Null : 아무것도 없다는 것을 의미하며 주로 예외처리에서 사용

```
var novalue = null;
```

- 문자열 : 따옴표(" ", ')에 들어가는 모든 문자

```
var string = "안녕하세요";
var string2= 'HTML5 Javascript';
```

2. JavaScript의 기본 문법

5) 형 변환

- 형 변환 : 숫자를 문자로, 문자를 숫자로 변환하는 것
- 형 변환의 종류 : 암시적 형 변환, 명시적 형 변환
- 암시적 형 변환

- 명령어를 수행하는 과정 중에 자바스크립트 엔진이 자동으로 변환하는 것
- 자바스크립트 컴파일러에 의해서 자동으로 형 변환이 일어남

```
var example1 = 10 + "20"; // 숫자 + 문자 = 문자로 변환됨
var example2 = "10" - 20; // 문자 - 숫자 = 숫자로 변환됨
var example3 = "10" + false; // 문자 - Boolean = 문자로
```

- 명시적 형 변환

- 명령어로 값을 원하는 유형으로 변환하는 것
- 사용자가 직접 지정해주는 방식

<String()을 이용하는 방법>

```
var numstring = 20;
console.log(numstring.typeof(numstring));
var ns = String(numstring);
console.log(ns.typeof(ns));
```

<toString(진수)를 이용하는 방법>

```
var numstring = 20;
console.log(numstring.toString()); // 문자열로 변환
console.log(numstring.toString(8)); // 8진수 문자열로 변환
console.log(numstring.toString(10)); // 10진수 문자열로 변환
console.log(numstring.toString(16)); // 16진수 문자열로 변환
```

<Number()을 이용하는 방법>

```
var stringnum = "20";
Number(stringnum);
Number("20");
Number(true) or Number(false) // Boolean 값 변환
```

2. JavaScript의 기본 문법

5) 형 변환

〈parseInt()을 이용하는 방법〉

```
var stringnum = "20";
parseInt(stringnum);
parseInt("20");
parseInt("0xFF"); // 255 반환
parseInt("29.234"); // 29 반환
```

〈parseFloat()을 이용하는 방법〉

```
var stringnum = "12.345";
parseFloat(stringnum);
parseFloat("12.345");
```

6) 함수

- 함수 : 함수는 특정 기능을 하는 변수와 구문들의 집합

- 중복 구문을 함수로 묶어서 재사용이 가능

- 파라메타와 반환값이 없는 함수

```
function 함수이름() {
    ...
    ...
}
```

- 파라메타만 있고 반환값이 없는 함수

```
function 함수이름(파라메타 이름1, 파라메타이름2...) {
    ...
    ...
}
```

- 파라메타와 반환값이 있는 함수

```
function 함수이름(파라메타 이름1, 파라메타이름2...) {
    ...
    ...
    return 반환값;
}
```

2. JavaScript의 기본 문법

6) 함수

- function 키워드로 함수 만들기

- 가장 일반적인 방식

```
function test() {  
    console.log("test");  
}  
test();
```

- 리터럴 방식으로 함수 만들기

- 이 방식도 많이 사용됨

```
var test = function() {  
    console.log("test");  
}  
test();
```

- function 객체 가지고 만들기

- 거의 사용되지 않고 있음

```
var test = new function(undefined, 'console.log("test")");  
test();
```

■ 정리하기

1. JavaScript의 개요

- 웹 브라우저에서 많이 사용하는 프로그래밍 언어
- **JavaScript**는 브라우저 기반에서 실행되는 **객체 기반 언어**이고, **Java**는 썬 마이크로시스템에서 개발한 **객체 지향 언어**임
- HTML 문서와 함께 처리되므로, **일반 프로그래밍 언어보다는 기능이 제한적**임
- 구성요소 : **객체, 문법, 이벤트와 이벤트 핸들러**

■ 정리하기

2. JavaScript의 기본 문법

- **표현식** : 값을 만들어내는 간단한 코드
- **문장** : 하나 이상의 표현식
- **키워드** : JavaScript가 처음 만들어질 때 정해진 특별한 의미가 있는 단어
- **식별자** : 이름을 붙일 때 사용하는 단어
- **변수** : 값을 저장할 때 사용하는 식별자
- **연산자의 유형** : 사칙 연산자, 비교 연산자, 논리 연산자, 복합 대입 연산자, 증감 연산자
- **자료형** : 정수형, 부동소수점, Boolean, Null, 문자열
- **형 변환** : 숫자를 문자로, 문자를 숫자로 변환하는 것
- **함수** : 특정 기능을 하는 변수와 구문들의 집합
- **파라메타** : 외부 데이터를 함수 내부로 받아올 수 있는 역할을 함