훈련교사 : 전 은 석
email : euns_jun@naver.com

JavaScript

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript

- 크로스-플랫폼, 객체지향 스크립트 언어
- 작고 가벼운 언어
- 웹브라우저(호스트 환경) 내에서 주로 사용 - JavaScript는 프로그램 제어를 제공
- 다른 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고 있다.
- Node.js와 같은 런타임 환경과 같이 서버 사이드 네트워크 프로그래밍에도 사용

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript

- 넷스케이프 커뮤니케이션즈 코퍼레이션의 브렌던 아이크(Brendan Eich)가 처음에는 모카(Mocha)라는 이름으로, 나중에는 라이브스크립트(LiveScript)라는 이름으로 개발하였으며, 최종적으로 자바스크립트(자바와 닮았다해서...)가 되었다.
  자바스크립트가 썬 마이크로시스템즈의 **자바와 구문(syntax)이 유사한 점도 있지만**, 이는 사실 **두 언어 모두 C 언어의 기본 구문을 바탕**했기 때문이고,
  자바와 자바스크립트는 직접적인 관련성이 없다.

❖ Java와 비교한 Javascript

JavaScript	Java
객체 지향. 객체의 형 간에 차이 없음. 프로토타입 메커니즘을 통한 상속, 속성과 메서드는 어떤 객체든 동적으로 추가될 수 있음.	클래스 기반. 객체는 클래스 계층구조를 통한 모든 상속과 함께 클래스와 인스턴스로 나뉨. 클래스와 인스턴스는 동적으로 추가된 속성이나 메소드를 가질 수 없음.
변수 자료형이 선언되지 않음 (동적 형지정, dynamic typing).	변수 자료형은 반드시 선언되어야 함 (정적 형지정, static typing).
하드 디스크에 자동으로 작성 불가.	하드 디스크에 자동으로 작성 가능.

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ **최근 버전 :** ECMAScript(자바스크립트의 표준화된 버전) 2016 (2016년 6월 17일)

❖ **브라우저마다 지원되는 버전이 다르며, 가장 범용적으로 지원되는 버전은 1.5이다.**

❖ **상표는 오라클 소유 /**
넷스케이프 커뮤니케이션스가 발명, 구현한 기술 및
모질라 재단과 같은 독립 기관의 라이선스 하에 사용된다.

❖ **유용한 링크**

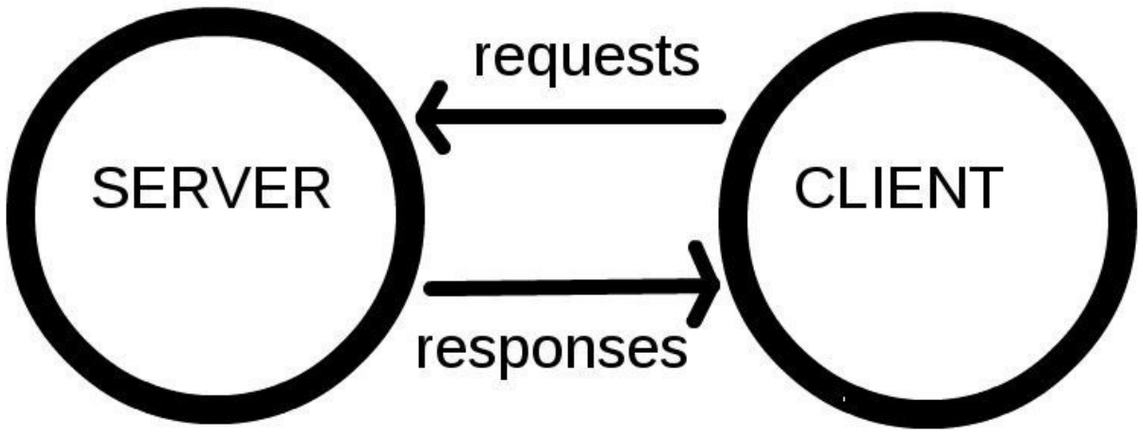
- **https://developer.mozilla.org/ko/docs/Web/JavaScript**
- **https://www.w3schools.com/js/default.asp**

❖ 웹의 동작 원리

- ◆ 요청(클라이언트)과 응답(서버)의 과정

❖ 클라이언트(사용자) : 서버에 요청하는 쪽

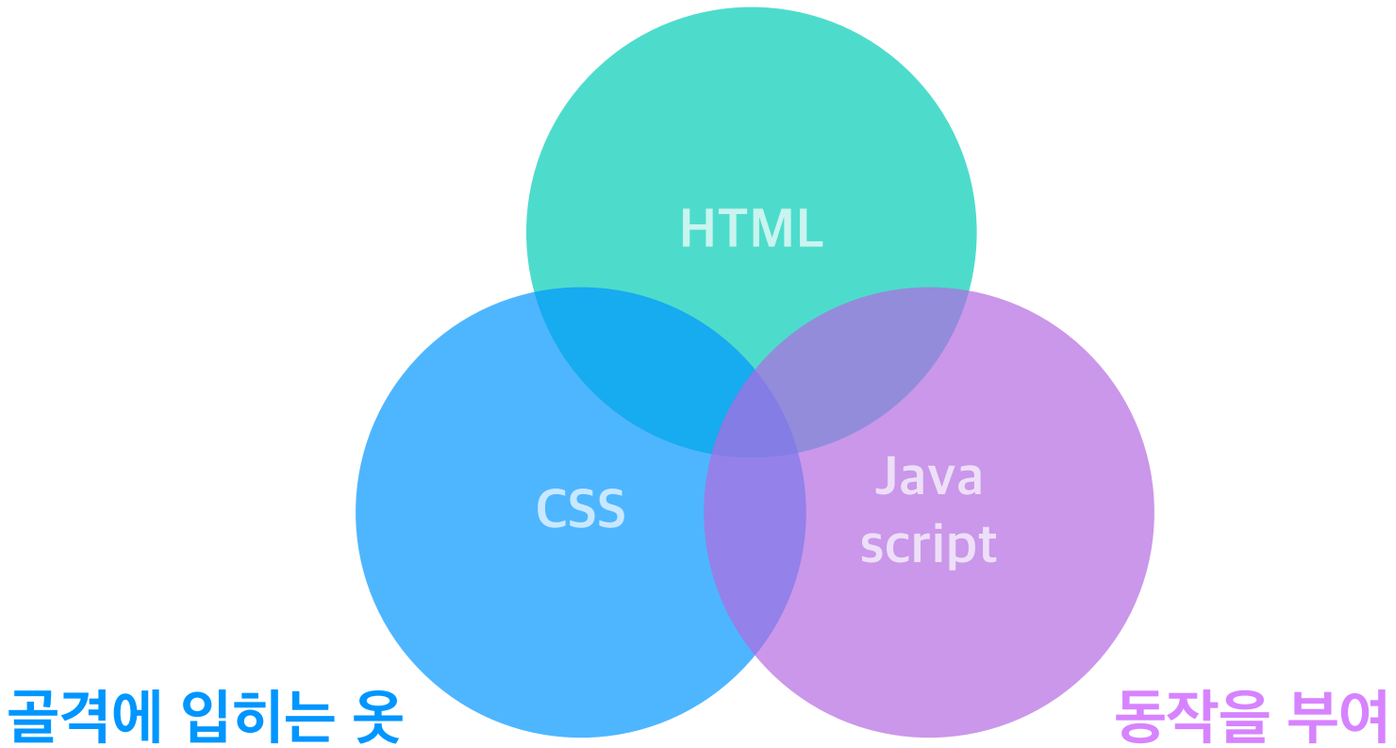
❖ 서 버(제공자) : 요청에 응답하는 쪽



훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 자바스크립트의 역할

* 웹 문서를 동적으로 제어하기 위해 고안된 프로그래밍 언어



훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 자바스크립트의 역할

* 요소의 추가 및 삭제
* CSS 및 HTML 요소의 스타일 변경
* 사용자와의 상호작용
* 폼의 유효성 검증
* 마우스와 키보드 이벤트에 대한 스크립트 실행
* 웹 브라우저 제어 및 쿠키 등의 설정과 조회
* AJAX 기술을 이용한 웹 서버와의 통신

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript 특징

* 인터프리터 언어 - 에러가 발생하면 에러가 발생한 다음 줄 부터는 구문 분석 않는다.
* 클라이언트 스크립트 언어 - 서버의 부하를 줄일 수 있다.
* 객체 기반 언어
* 공개된 언어
* 다양한 라이브러리 - 예] jQuery, NODE.js

❖ Javascript 사용 이유

* 서버의 부하를 줄일 수 있다.
* 동적인 사이트 구현
* 인터렉티브한 사이트 구현
* 다양한 라이브러리 언어 활용
* HTML5 API 기반 언어

❖ Javascript 작성 방법

- 대소문자 구분하여 작성
- 문장은 세미콜론(;)으로 구분

바른 예	var age=25 document.write("당신의 나이는 " + age + "입니다.")
	var age=25; document.write("당신의 나이는 " + age + "입니다.");
	var age=25; document.write("당신의 나이는 " + age + "입니다.");
잘못된 예	var age=25 document.write("당신의 나이는 " + age + "입니다.")

- 큰따옴표(" ")와 작은따옴표(' ')를 구분하여 사용

바른 예	document.write(" 자바스크립트 학습 ");
	document.write(' 자바스크립트 학습 ');
잘못된 예	document.write(" 자바스크립트 학습 ")

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript 포함 방법

- **HTML 문서 내부에 코드를 작성하는 방법**

✤ <head> 태그 또는 <body> 태그 내에 코드 작성

```
<head>
    <meta charset="utf-8"/>
    <title>자바스크립트 예제</title>
    <script>
        // 자바스크립트 코드 작성
    </script>
</head>
<body>
    <script>
        // 자바스크립트 코드 작성
    </script>
</body>
```

✤ HTML 태그 안에 속성값으로 정의

```
<button type="button" onclick="alert('자바스크립트')">버튼 클릭</button>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript 포함 방법

◆ 자바스크립트 코드의 실행 순서 살펴보기

js_test01.html

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8"/>
   <title>자바스크립트 예제</title>
   <script>
      var num=0;
      document.write("head 태그 내 실행 순서 : " + num + "<br/>");
   </script>
   <script>
      var num=1;
      document.write("head 태그 내 실행 순서 : " + num + "<br/>");
   </script>
</head>
<body>
   <script>
      var num=2;
      document.write("body 태그 내 실행 순서 : " + num + "<br/>");
   </script>
   <script>
      var num = 3;
      document.write("body 태그 내 실행 순서 : " + num + "<br/>");
   </script>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript 포함 방법

• 별도로 작성한 후 HTML 문서에서 참조하는 방법

- 외부 자바스크립트 파일을 만든 후 HTML 문서의 <script> 태그에 src 속성을 추가하여 참조

위치	src 속성값
HTML 문서와 같은 디렉터리에 있는 경우	〈script src="myscript.js"〉〈/script〉
HTML 문서와 다른 디렉터리에 있는 경우	〈script src="./ejs/myscript.js"〉〈/script〉
HTML 문서와 다른 서버 디렉터리에 있는 경우	〈script src="http://www.hanbit.co.kr/jsfile/myscript.js"〉〈/script〉

✻ **자바스크립트 파일을 외부에서 작성했을 때의 장점**

- 자바스크립트 파일을 HTML 문서와 분리하여 관리할 수 있음
- 자바스크립트 코드를 관리, 유지보수, 디버깅하기 쉬움
- 자바스크립트 코드의 보안성과 안전성을 높일 수 있음

❖ Javascript 포함 방법

◆ 외부 자바스크립트 문서 작성 후 참조하기

./js/test_js01.js

```
var age=23;
/* 문자에 스타일 속성 적용 */
document.write("<div style='color: red; font-size: 24px;'>외부 자바스크립트 파일</div>");
document.write("당신의 나이는 " + age + "입니다.");
```

js_test02.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
    <script src="./js/test_js01.js"> </script>
</head>
<body>
    <p>
    <!-- 버튼을 클릭하면 메시지 창 출력 -->
    <button type="button" onclick="alert('외부 자바스크립트 파일')">버튼 클릭</button>
    </p>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript 포함 방법

◆ 내포 관계인 자바스크립트 파일 참조하기

./js/test_js02.js

```
document.write(“test_js02.js");
document.write("<div style='color: red; font-size: 24px;'>외부 자바스크립트 파일</div>");
document.write("<script src=‘./js/test_js3.js'> </script>");
```

./js/test_js03.js

```
document.write(“test_js03.js는 test_js02.js에 포함");
document.write("<div style='color: blue; font-size: 20px;'>외부 자바스크립트 파일</div>");
document.write("<script src=‘./ejs/test_js04.js'> </script>");
```

./js/test_js04.js

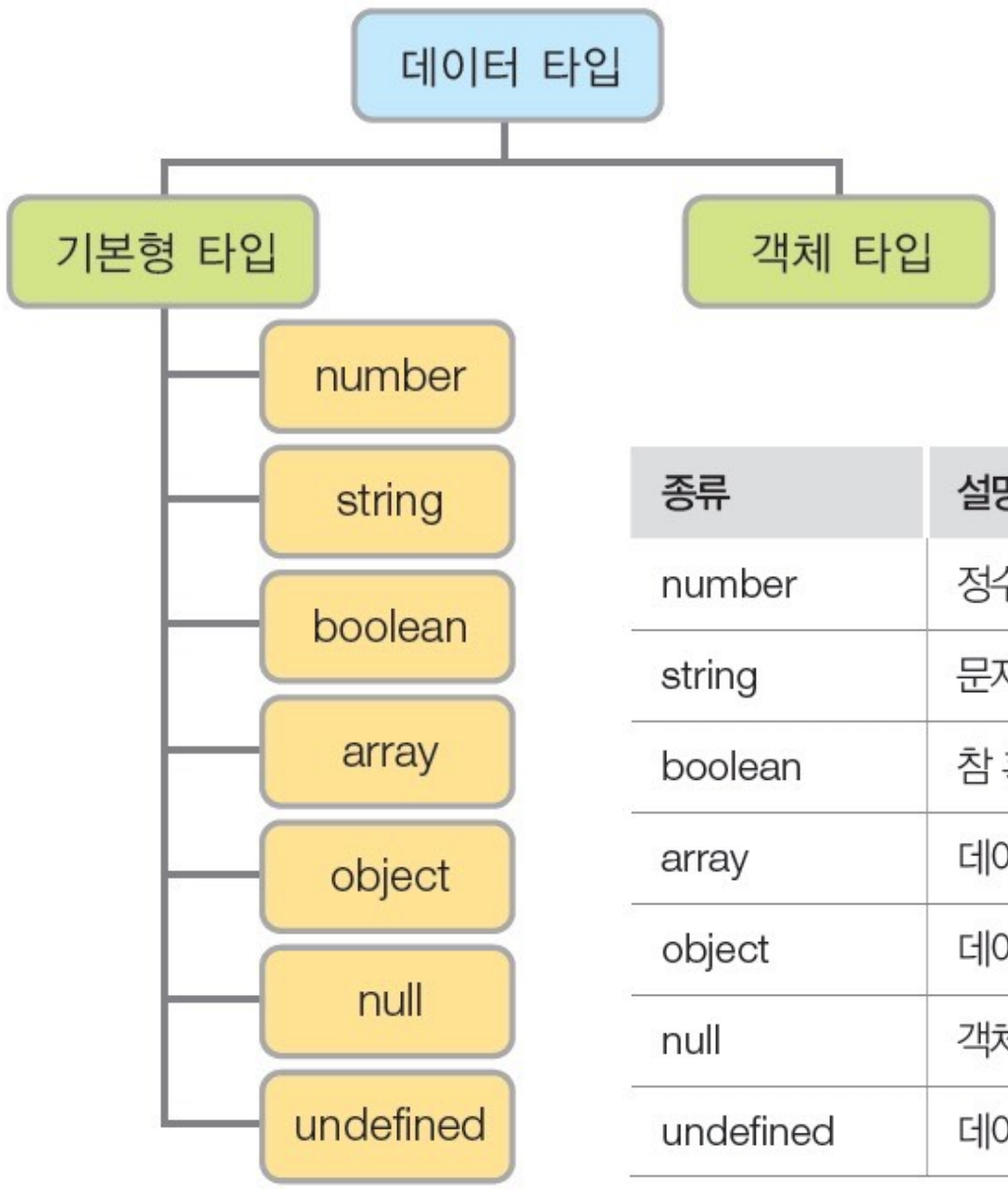
```
document.write(“test_js04.js는 test_js03.js에 포함");
document.write("<div style='color: green; font-size: 16px;'>외부 자바스크립트 파일</div>");
alert('Nested Script File');
```

js_test03.html

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8"/>
</head>
<body>
   <script src="./ejs/njs1.js"> </script>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript 데이터 타입



기본형 타입의 종류

종류	설명	사용 예
number	정수 혹은 실수	100, 10.5, 10e+3
string	문자 혹은 문자열	“홍길동”, ‘홍길동’
boolean	참 혹은 거짓	true, false
array	데이터의 집합(배열, 객체로 취급)	[“서울”, “부산”, “인천”]
object	데이터 속성과 값으로 이루어진 집합	{name: ‘홍길동’, age: 25}
null	객체 값이 없음	null
undefined	데이터 값이 정해지지 않음	undefined

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript 포함 방법

◆ typeof 연산자를 사용하여 데이터 타입 확인하기

js_datatype.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset=“utf-8"/>
    <title>Javascript Data Type</title>
</head>
<body>
    <script>
        var num;        // 변숫 값이 없음
        var obj=null;   // 객체 변숫 값이 없음
        document.write(typeof 100+"<br>");
        document.write(typeof 10.5+"<br>");
        document.write(typeof "홍길동"+"<br>");
        document.write(typeof true+"<br>");
        document.write(typeof [1,2,3]+"<br>");
        document.write(typeof {name:'홍길동', age:25}+"<br>");
        document.write(typeof num+"<br>");
        document.write(typeof obj+"<br>");
    </script>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript 변수명 규칙

- **변수명 작성 규칙**
  - 문자, 밑줄(_), 달러 기호($)로 시작
  - 대소문자 구별('변수 A'와 '변수 a'는 서로 다른 변수)
  - 한글은 사용 가능하나 영문자 사용 권장
  - 자바스크립트에서 정한 예약어는 변수명으로 사용 불가능

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

✤ **Javascript 예약어**

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 변수 사용 방법

● 변수 사용 예

- var x; // 변수 x 선언
- var y=10; // 변수 y 선언 및 초깃값 할당
- var x=y; // 변수 y의 값을 변수 x에 저장
- var a, b, c; // 변수 a, b, c 선언
- var a=10, b=11, c=12; // 변수 a, b, c 선언 및 각각 다른 초깃값 할당
- var a=b=c=10; // 변수 a, b, c 선언 및 같은 초깃값 할당
- var name=“홍길동”, age=25; // 변수 name, age 선언 및 각각 다른 초깃값 할당
- var total=a+b+c; // 변수 a, b, c 값을 더한 결과를 변수 total에 저장

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 변수 사용 방법

● 변수 사용 시 문법적으로 오류가 발생한 사례

- var 7num=100; // 숫자로 시작하는 변수명 잘못 사용
- var &num=100; // 특수 문자로 시작하는 변수명 잘못 사용
- var true=1; // 예약어를 변수명으로 잘못 사용
- var 10=x; // 좌변에 상수값 잘못 선언
- var a+b=20; // 좌변에 연산식 잘못 선언
- var “홍길동”=name; // 좌변에 문자열값 잘못 선언
- var get Number=100; // 변수명 사이에 공백(space) 잘못 선언
- var a, b, c=100; // 콤마로 구분한 변수명 잘못 선언

❖ Javascript 포함 방법

◆ 변수의 재선언후 데이터 타입

js_data_retype02.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
</head>
<body>
    <script>
        stdName="홍길동";    // var 키워드 생략
        comGrade=96;         // var 키워드 생략
        var stdName;         // 변수명 재선언
        var comGrade;        // 변수명 재선언
        document.write("학생 이름 : " + stdName + "<br>");
        document.write("컴퓨터 점수 : " + comGrade + "<br>");
    </script>
</body>
</html>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ Javascript 전역변수 & 지역변수

- **전역 변수**
  - 코드 내 어느 위치에서든 선언하여 전 영역에서 사용할 수 있는 변수
- **지역 변수**
  - 변수가 선언된 해당 블록에서 선언하여 범위 내에서만 유효하게 사용할 수 있는 변수

```
<script>
    var globValue1;     // 전역 변수 선언
        globValue2;     // 전역 변수 선언, var 생략

    function 함수() {
        var locValue;   // 지역 변수 선언
            globValue;  // 함수 내부에서 var 생략 시 자동 전역 변수로 선언
        locValue=10;    // 지역 변수 사용
    }

    globValue=10;       // 전역 변수 사용
</script>
```

전역 변수와 지역 변수