

# ABC3012: Advanced Web Programming

Donghyun Kang ([donghyun@changwon.ac.kr](mailto:donghyun@changwon.ac.kr))

NOSLab (<https://noslab.github.io>)

Changwon National University

# 마지막 과제

- Github에 본인의 홈페이지를 만드세요
  - 예: 개인 홈페이지, 상품 소개, 블로그
  - 반드시 HTML, Java Script, CSS가 별도의 파일로 관리되고 사용되어야 함
- 과제 기간 (6주 프로젝트)
  - 9주차 ~ 13주차
- 과제 발표
  - 14주차 실시간 강의 시간에 개인 홈페이지 공개 및 내용 소개
    - 6월 8일

# 자바스크립트 식별자

- 식별자
  - 자바스크립트 프로그램의 변수, 상수(리터럴), 함수의 이름
  - 식별자 만드는 규칙
    - 첫 번째 문자 : 알파벳(A-Z, a-z), 언더스코어(\_), \$ 문자만 사용 가능
    - 두 번째 이상 문자 : 알파벳, 언더스코어(\_), 0-9, \$ 사용 가능
    - 대소문자는 구분되어 다루어짐
      - myHome과 myhome은 다른 식별자
    - 자바스크립트 예약어 사용 불가
      - false, for, if, null 등 자바스크립트 예약어 사용 불가
  - 식별자 사용 사례

```
6variable;      // (x) 숫자로 시작할 수 없음
student_ID;    // (0)
_code;         // (0) 맞지만 권하지 않음
if;            // (x) 예약어 if 사용 불가
%calc          // (x) % 사용 불가
bar, Bar;      // (0) bar와 Bar는 서로 다른 식별자임에 주의
```

# 자바스크립트 문장

- 문장

- 자바스크립트 프로그램의 기본 단위는
- 문장과 문장을 구분하기 위해 세미콜론(;) 사용

```
i = i + 1           // (0) 한 줄에 한 문장만 있는 경우 세미콜론 생략 가능  
j = j + 1;         // (0)  
k = k + 1; m = m + 1; // (0) 한 줄에 여러 문장  
n = n + 1 p = p + 1; // (x) 첫 번째 문장 끝에 세미콜론이 필요함
```

- 주석문

```
// 한 라인 주석. 라인의 끝까지 주석 처리  
/*  
  여러 라인 주석  
*/
```

# 데이터 타입

- 자바스크립트 언어에서 다루는 데이터 종류
  - 숫자 타입 : 정수, 실수(예: 42, 3.14)
  - 논리 타입 : 참, 거짓(예: true, false)
  - 문자열 타입(예: '좋은 세상', "a", "365", "2+4")
  - 객체 레퍼런스 타입 : 객체를 가리킴. c 언어의 포인터와 유사
  - null : 값이 없음을 표시하는 특수 키워드. Null, NULL과는 다름
- 특징
  - 자바스크립트에는 문자 타입 없음. 문자열로 표현

# 변수

- 변수 : 자바스크립트 데이터 저장 공간
  - 변수 선언 : 변수 이름을 정하고, 저장 공간 할당

```
var score;           // 변수 score 선언
var year, month, day; // year, month, day의 3 개의 변수 선언
var address = "서울시"; // address 변수를 선언하고 "서울시"로 초기화
```

- var 없이 선언

```
age = 21;           // var 없이, 변수 age를 선언하고 21로 초기화
```

- age가 이미 선언된 변수이면, 존재하는 age에 21 저장

- 자바스크립트에는 변수 타입 없음
  - 변수 타입 선언하지 않음

```
var score; // 정상적인 변수 선언
int score; // 오류. 변수 타입 int 없음
```

- 변수에 저장되는 값에 대한 제약 없음

```
score = 66.8; // 실수도 저장 가능
score = "high"; // 문자열로 저장 가능
```

# 지역변수와 전역변수

지역변수	전역변수
함수 내에 var 키워드로 선언	함수 밖에 선언되거나, 함수 내에 var 키워드 없이 선언된 변수
선언된 함수 내에서만 사용	프로그램 전역에서 사용

```
var x;    // 전역변수 x
function f() {
  var y;  // 지역변수 y 선언
  x = 10; // 전역 변수 x에 10 저장
  y = 10; // 지역 변수 y에 10 저장
  z = 10; // 새로운 전역변수 z 선언. 10으로 초기화
}
```

# this로 전역변수 접근

- 지역 변수와 전역 변수의 이름을 같을 때
  - 전역 변수에 접근하고자 할 때 : **this.전역변수**

```
var x;    // 전역변수
function f() {
  var x;   // 지역변수
  x = 1;   // 지역변수 x에 1 저장
  this.x = 100; // 전역변수 x에 100 저장
}
```



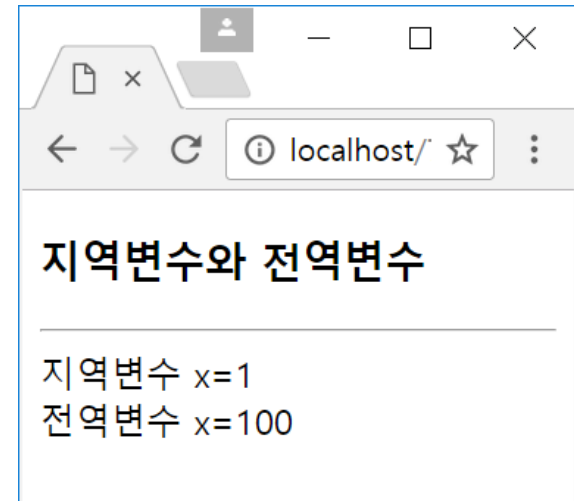
# 예제 6-6 지역변수와 전역변수

```
<!DOCTYPE html>
<html>
<head>
<title>지역변수와 전역변수</title></head>
<body>
<h3>지역변수와 전역변수</h3>
<hr>
<script>
var x=100; // 전역변수 x

function f() { // 함수 f() 선언
    var x=1; // 지역변수 x

    document.write("지역변수 x=" + x);
    document.write("<br>");
    document.write("전역변수 x=" + this.x);
}

f(); // 함수 f() 호출
</script>
</body>
</html>
```



# 자바스크립트의 상수

- 상수(literal)
  - 데이터 값 그 자체
- 상수 종류


상수의 종류		특징	예
정수	8진수	0으로 시작	var n = 015; // 8진수 15. 10진수로 13
	10진수		var n = 15; // 10진수 15
	16진수	0x로 시작	var n = 0x15; // 16진수 15. 10진수로 21
실수	소수형		var height = 0.1234;
	지수형		var height = 1234E-4; // $1234 \times 10^{-4} = 0.1234$
논리	참	true	var condition = true;
	거짓	false	var condition = false;
문자열		""로 묶음	var hello = "안녕하세요";
		''로 묶음	var name = 'kitae';
기타	null	값이 없음을 뜻함	var ret = null;
	NaN	수가 아님을 뜻함	var n = parseInt("abc"); // 이때 parseInt()는 NaN을 리턴

# 문자열 상수

- 이중 인용 부호(“”)와 단일 인용 부호(‘’) 모두 사용
- 문자열 내에 문자열

`<p onmouseover = "document.body.style.color = 'brown' ">`

문자열 내 문자열

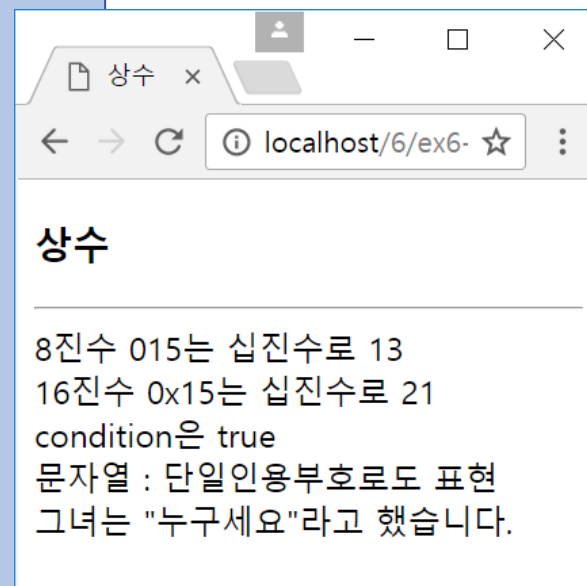


- “ 문자를 그대로 사용하고자 하는 경우 \”로 사용할 것

# 예제 6-7 상수

```
<!DOCTYPE html>
<html>
<head><title>상수</title></head>
<body>
<h3>상수</h3>
<hr>
<script>
  var oct = 015; // 015는 8진수. 10진수로 13
  var hex = 0x15; // 0x14는 16진수. 10진수로 21
  var condition = true; // True로 하면 안됨

  document.write("8진수 015는 십진수로 " + oct + "<br>");
  document.write("16진수 0x15는 십진수로 " + hex + "<br>");
  document.write("condition은 " + condition + "<br>");
  document.write('문자열 : 단일인용부호로도 표현' + "<br>");
  document.write("그녀는 \"누구세요\"라고 했습니다.");
</script>
</body>
</html>
```



# 자바스크립트의 식과 연산

- 자바스크립트의 연산과 연산자 종류

연산 종류	연산자	연산 종류	연산자
산술	+ - * / %	대입	= *= /= += -= &= ^=  = <<= >>= >>>=
증감	++ --	비교	> < >= <= == !=
비트	&   ^ ~	논리	&&    !
시프트	>> << >>>	조건	? :

- 산술 연산자

- 5 가지 : 더하기(+), 빼기(-), 곱하기(\*), 나누기(/), 나머지(%)

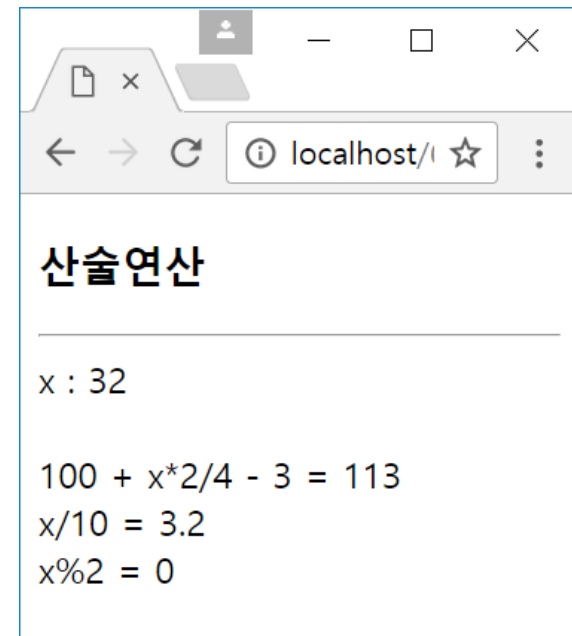
```
var x = 32;  
var total = 100 + x*2/4 - 3; // total은 113
```

```
var div = 32/10;           // div = 3.2
```

# 예제 6-8 산술 연산

```
<!DOCTYPE html>
<html>
<head><title>산술연산</title></head>
<body>
<h3>산술연산</h3>
<hr>
<script>
  var x=32;
  var total = 100 + x*2/4 - 3; // total은 113
  var div = x / 10; // div는 3.2
  var mod = x % 2; // x를 2로 나눈 나머지, 0

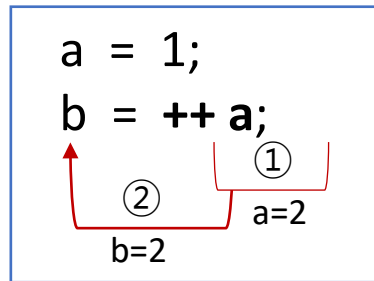
  document.write("x : " + x + "<br><br>");
  document.write("100 + x*2/4 - 3 = " + total + "<br>");
  document.write("x/10 = " + div + "<br>");
  document.write("x%2 = " + mod + "<br>");
</script>
</body>
</html>
```



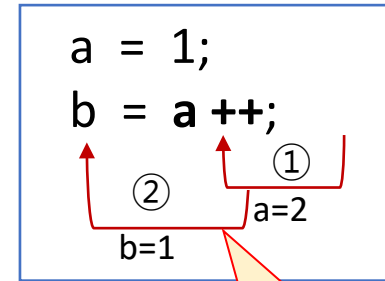
# 증감 연산자

- 증감 연산자 : ++, --

(a) 전위연산자



(b) 후위연산자



a++의 연산은 증가  
전의 값 1을  
반환하여 b의 값은  
1이 됨

연산자	내용	연산자	내용
a++	a를 1 증가하고 증가 전의 값 반환	++a	a를 1 증가하고 증가된 값 반환
a--	a를 1 감소하고 감소 전의 값 반환	--a	a를 1 감소하고 감소된 값 반환

# 대입 연산자

- 대입 연산 : 오른쪽 식의 결과를 왼쪽 변수에 대입

```
var a=1, b=3;  
a = b;    // a에 b의 값이 대입되어 a=3, b=3이 된다.  
a += b;   // a = a + b의 연산이 이루어져, a=6, b=3이 된다.
```

- 대입연산자 종류

연산자	내용	연산자	내용
a = b	b 값을 a에 대입	a &= b	a = a & b와 동일
a += b	a = a + b와 동일	a ^= b	a = a ^ b와 동일
a -= b	a = a - b와 동일	a  = b	a = a   b와 동일
a *= b	a = a * b와 동일	a <<= b	a = a << b와 동일
a /= b	a = a / b와 동일	a >>= b	a = a >> b와 동일
a %= b	a = a % b와 동일	a >>>= b	a = a >>> b와 동일

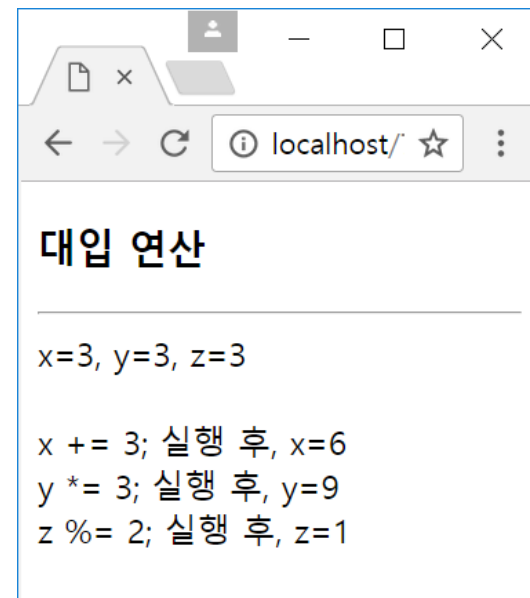


# 예제 6-9 대입 연산

```
<!DOCTYPE html>
<html>
<head><title>대입 연산</title></head>
<body>
<h3>대입 연산</h3>
<hr>
<script>
  var x=3, y=3, z=3;
  document.write("x=" + x + ", y=" + y);
  document.write(", z=" + z + "<br><br>");

  x += 3; // x=x+3 -> x=6
  y *= 3; // y=y*3 -> y=9
  z %= 2; // z=z%2 -> z=1

  document.write("x += 3; 실행 후, x=" + x + "<br>");
  document.write("y *= 3; 실행 후, y=" + y + "<br>");
  document.write("z %= 2; 실행 후, z=" + z);
</script>
</body>
</html>
```



# 비교 연산자

- 비교 연산 : 두 값 비교, true나 false의 결과를 내는 연산

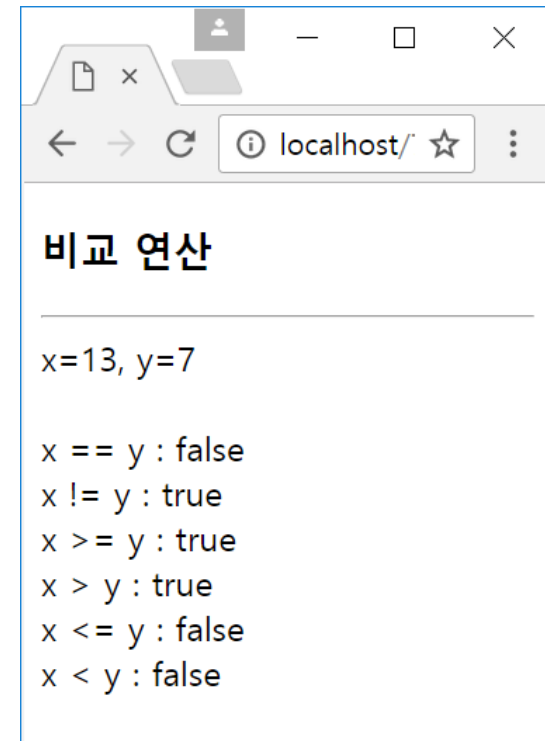
```
var age = 25;  
var result = (age > 20); // age가 20보다 크므로 result는 true
```

- 비교 연산자 종류

연산자	내용	연산자	내용
a < b	a가 b보다 작으면 true	a >= b	a가 b보다 크거나 같으면 true
a > b	a가 b보다 크면 true	a == b	a가 b와 같으면 true
a <= b	a가 b보다 작거나 같으면 true	a != b	a가 b와 같지 않으면 true

# 예제 6-10 비교 연산

```
<!DOCTYPE html>
<html>
<head><title>비교 연산</title></head>
<body>
<h3>비교 연산</h3>
<hr>
<script>
  var x=13, y=7;
  document.write("x=" + x + ", y=" + y + "<br><br>");
  document.write("x == y : " + (x == y) + "<br>");
  document.write("x != y : " + (x != y) + "<br>");
  document.write("x >= y : " + (x >= y) + "<br>");
  document.write("x > y : " + (x > y) + "<br>");
  document.write("x <= y : " + (x <= y) + "<br>");
  document.write("x < y : " + (x < y) + "<br>");
</script>
</body>
</html>
```



# 논리 연산자

- 논리 연산 : AND, OR, NOT

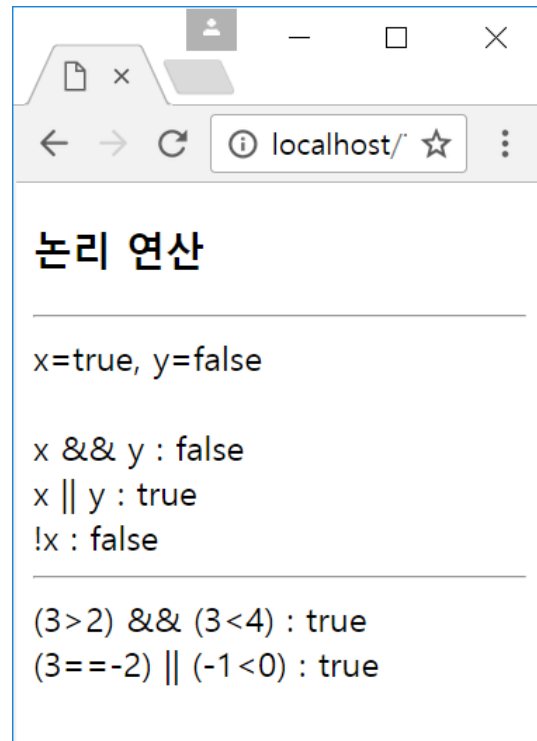
```
var score = 90;  
var age = 20;  
var res = ((score > 80) && (age < 25)); // res=true
```

- 논리 연산 종류

연산자	별칭	내용
a && b	논리 AND 연산	a, b 모두 true일 때 true 리턴
a    b	논리 OR 연산	a, b 중 하나라도 true이면 true 리턴
!a	논리 NOT 연산	a가 true이면 false 값을, false이면 true 값 리턴

# 예제 6-11 논리 연산

```
<!DOCTYPE html>
<html>
<head><title>논리 연산</title></head>
<body>
<h3>논리 연산</h3>
<hr>
<script>
  var x=true, y=false;
  document.write("x=" + x + ", y=" + y + "<br><br>");
  document.write("x && y : " + (x&&y) + "<br>");
  document.write("x || y : " + (x||y) + "<br>");
  document.write("!x : " + (!x) + "<br>");
  document.write("<hr>");
  document.write("(3>2) && (3<4) : " + ((3>2)&&(3<4)) + "<br>");
  document.write("(3==2) || (-1<0) : " + ((3==2)||(-1<0)));
</script>
</body>
</html>
```



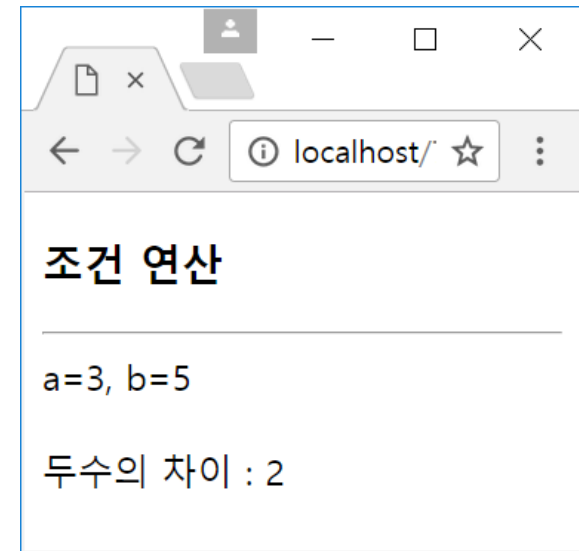
# 조건 연산자

- 조건 연산
  - `condition ? expT : expF`
    - `condition`이 `true`이면 전체 결과는 `expT`의 계산 값
    - `false`이면 `expF`의 계산 값

```
var x=5, y=3;  
var big = (x>y) ? x : y; // (x>y)가 true이므로 x 값 5가 big에 대입된다.
```

# 예제 6-12 조건 연산

```
<!DOCTYPE html>
<html>
<head><title>조건 연산</title></head>
<body>
<h3>조건 연산</h3>
<hr>
<script>
  var a=3, b=5;
  document.write("a=" + a + ", b=" + b + "<br><br>");
  document.write("두수의 차이 : " + ((a>b)?(a-b):(b-a)));
</script>
</body>
</html>
```



# 문자열 연산자

- 문자열 연결

- +, +=

```
"abc" + "de"    // "abcde"
"abc" + 23      // "abc23"
23 + "abc"      // "23abc"
23 + "35"       // "2335"
23 + 35         // 58, 정수 더하기
```

- 순서에 유의

```
23 + 35 + "abc";    // 23 + 35 -> 58로 먼저 계산, 58 + "abc" -> "58abc"
"abc" + 23 + 35;    // "abc" + 23 -> "abc23"로 먼저 계산, "abc23" + 35 -> "abc2335"
```

- 문자열 비교

- 비교 연산자(!=, ==, >, <, <=, >=)는 문자열 비교에 사용
- 사전 순으로 비교 결과 리턴

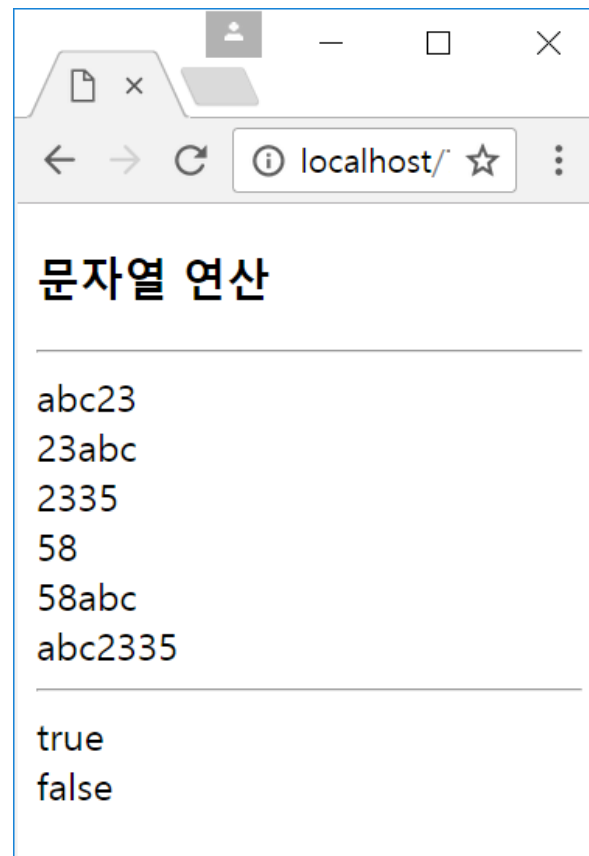
```
var name = "kitae";
var res = (name == "kitae"); // 비교 결과 true, res = true
var res = (name > "park"); // name이 "park"보다 사전순으로 앞에 나오므로 res = false
```



# 예제 6-14 문자열 연산

```
<!DOCTYPE html>
<html>
<head><title>문자열 연산</title></head>
<body>
<h3>문자열 연산</h3>
<hr>
<script>
  document.write("abc" + 23 + "<br>");
  document.write(23 + "abc" + "<br>");
  document.write(23 + "35" + "<br>");
  document.write(23 + 35 + "<br>");
  document.write(23 + 35 + "abc" + "<br>");
  document.write("abc" + 23 + 35 + "<br><hr>");

  var name = "kitae";
  document.write(name == "kitae");
  document.write("<br>");
  document.write(name > "park");
</script>
</body>
</html>
```



# if, if-else

- if, if-else 문

```
if(조건식) {  
    ... 실행문 ... // 조건식이 참인 경우  
}
```

```
if(a > b) {  
    document.write("a가 크다");  
}
```

```
if(조건식) {  
    ... 실행문1 ... // 조건식이 참인 경우  
}  
else {  
    ... 실행문2 ... // 조건식이 거짓인 경우  
}
```

```
if(a > b) {  
    document.write("a가 크다");  
}  
else {  
    document.write("a가 크지 않다");  
}
```

```
if(조건식1) {  
    실행문1 // 조건식1이 참인 경우  
}  
else if(조건식2) {  
    실행문2 // 조건식2가 참인 경우  
}  
.....  
else {  
    실행문n; // 앞의 모든 조건이 거짓인 경우  
}
```

```
if(a > b) {  
    document.write("a가 크다");  
}  
else if(a < b) {  
    document.write("b가 크다");  
}  
else  
    document.write("a와 b는 같다");
```

# 예제 6-15 if-else 사용

```
<!DOCTYPE html>
<html>
<head><title>if-else</title></head>
<body>
<h3>if-else를 이용한 학점 매기기</h3>
<hr>
<script>
  var grade;
  var score = prompt("황기태 님 점수를 입력하세요", 100);
  score = parseInt(score); // 문자열을 숫자로 바꿈
  if(score >= 90) // score가 90 이상
    grade = "A";
  else if(score >= 80) // 80 이상 90 미만
    grade = "B";
  else if(score >= 70) // 70 이상 80 미만
    grade = "C";
  else if(score >= 60) // 60 이상 70 미만
    grade = "D";
  else // 60 미만
    grade = "F";
  document.write(score + "는 " + grade + "입니다.<br>");
</script>
</body>
</html>
```

localhost 내용:

황기태 님 점수를 입력하세요

확인

취소

if-else를 이용한 학점 매기기

95는 A입니다.

# Q&A