

# JS기본-2

☰ 태그	
📅 날짜	@2023년 5월 26일

비교 연산자, 조건문

비교 연산자

= 하나는 할당을 의미한다

== 같은가를 비교

!= 다른가를 비교 // true나 false로 반환

```
console.log(10>5); // true
console.log(10 == 5); // 동등 연산자 false
console.log( 10 != 5); // true
```

```
const a = 1;
const b = "1";
console.log(a == b);
// true 숫자형과 문자형을 비교 했는데 true가 나왔다.
// 이때 ===을 사용하면 타입까지 비교하게 된다.

const a = 1;
const b = "1";
console.log(a === b);
// false , === 를 일치 연산자라고 하는데, 가급적
// 동등연산자 보단 타입까지 비교해주는 일치 연산자를
// 사용하는게 좋다.
```

# 조건문

if문 ( 괄호안에 결과가 true일 경우 조건문 실행)  
boolean으로 판단한다.

if 문이 여러개라면 else를 활용  
else문일 경우 if문이 false 일 경우에 실행된다.

```
// if
const age = 30;

if (age > 19) {
  console.log('환영합니다. ');
}

if (age <= 19){
  console.log('안녕히 가세요. ')
}

console.log('-----');

// "환영합니다."
// "-----"

-----

// else
const age = 30;

if (age > 19) {
  console.log('환영합니다. ');
}else{
  console.log('안녕히 가세요. ')
}
console.log('-----');
// "환영합니다."
// "-----"

-----

// else
const age = 10;
if (age > 19) {
  console.log('환영합니다. ');
}else{
```

```

    console.log('안녕히 가세요. ')
  }
  console.log('-----');
  // "안녕히 가세요."
  // "-----"
  -----
  // else if
  // 추가 요구 :
  // 19살이면 수능 잘 치세요 라는 문구를 출력
  // age === 19
  const age = 19;

  if (age > 19) {
    console.log('환영합니다. ');
  } else if (age === 19) {
    console.log('수능 잘 치루세요');
  }
  else {
    console.log('안녕히 가세요. ')
  }
  console.log('-----');
  //"수능 잘 치루세요"
  //"-----"

  // age가 10일때
  // "안녕히 가세요."
  // "-----"

  // age 가 30일때
  //"환영합니다."
  // "-----"

```

## 논리연산자

|| (OR) : 여러개 중 하나라도 true면 true

즉, 모든 값이 false 일때만 false를 반환

a || b : a나 b중 true가 있으면 true

&& (AND) : 모든 값이 true면 true

즉, 하나라도 false라면 false를 반환

a && b : a와 b 둘 다 true면 true

! (NOT) : true 면 false, false면 true

!a : a가 false 이면 true

평가

OR는 첫번째 true를 발견하는 즉시 평가를 멈춘다.

ex) 스티브 잡스는 남자 이거나 OR, 한국인 이거나, 군인이거나....

남자 이거나 OR에서 true를 반환하기 때문에 평가를 멈춘다.

AND는 첫번째 false를 발견하는 즉시 평가를 멈춘다.

ex) 스티브 잡스는 남자이고 and, 한국인이며, 군인인 동시에...

에서 한국인이며, 가 false이기 때문에 즉시 평가를 멈춘다.

# 평가

전체 군인의 60%

운전면허가 있고 시력이 좋은 여군

전체 군인의 80%

전체 군인의 7%

→ 여군인데 시력이 좋고 운전면허가 있는 사람 의 순서로 코딩을 하는 것이 좋다.  
첫번째 평가에서 93%를 걸러낼 수 있으니 성능 최적화에 도움이 된다.

```
// OR
// 이름이 TOM 이거나, 성인이면 통과

const name = "mike";
const age = 30;

if (name === 'tom' || age > 19) {
  console.log('통과');
}

// 통과

-----
// and
// 이름이 mike 이고, 성인이면 통과

const name = "mike";
const age = 30;

if (name === 'mike' && age > 19) {
  console.log('통과');
}else {
  console.log('돌아가');
}

// age 가 10이면 돌아가
```

```

-----
// not
// 나이를 입력받아 성인이 아니면 돌아가라고 출력

const age = prompt('나이가?');
const isAge = age > 19;

if(!isAge){
  console.log('돌아가!');
}

-----

// 우선순위
// 남자이고, 이름이 mike 이거나 성인이면 통과

const gender = 'F';
const name = 'Jane';
const isAdult = true;

if ( gender === 'M' && name === 'mike' || isAdult){
  console.log('통과')
}else {
  console.log('돌아가.')
}

// &&가 || 보다 우선순위가 높다.
// 그래서 m 이고 mike에서 false를 반환하고 마지막 isAdult에서 true를 반환하게 되어 true를 반환하게 된다.

// 위 코드는 아래 코드와 같다.
if ( (gender === 'M' && name === 'mike') || isAdult){
  console.log('통과')
}else {
  console.log('돌아가.')
}

의도한 바로 만드려면 아래처럼 만들면 된다.
if ( gender === 'M' && (name === 'mike' || isAdult)){
  console.log('통과')
}else {
  console.log('돌아가.')
}

// 돌아가

```

# 반복문 loop

: 동일한 작업을 여러번 반복

for

```
for ( 초기값; 조건(false가 되면 멈춤); 코드 실행 후 작업){
  //반복할 코드
}

// ; 으로 구분
// 초기값을 지정하고 조건이 true이면 코드를 실행하고
// 후 작업

// 1 부터 10까지 로그

for (let i = 0; i < 10; i++){
  console.log(i+1);
}
//1
//2
//3
//...
//10
```

while

```
let i = 0;

while(i < 10){
  //코드
}

// i는 계속 0이기 때문에 무한 반복된다.
// 코드 부분에 i를 증가시키는 구문을 추가한다.

// while

let i = 0;
```

```

while (i < 10) {
  console.log(i);
  i++;
};
// 0
// 1
// ...
// 8
// 9

```

## do while

```

let i = 0;
do {
  // 코드
  i ++;
}
while (i < 10)

// i 가 10보다 작으면 do내부의 코드를 실행한다.
// 코드를 실행하고 조건을 체크하기 때문에 무조건 한번은
// 실행된다

```

## break, continue

break : 멈추고 빠져나옴

continue : 멈추고 다음 반복으로 진행

```

// break

while (true) {
  let answer = confirm('계속할까요?');
  if(!answer){
    break;
  }
}

// 확인을 누르면 계속 반복해서 실행되고

```



```
// 취소를 누르면 break를 실행하게 되어 빠져나온다.
```

```
// continue
// 짝수만

for (let i = 0; i < 10; i++){
  if(i%2){
    continue;
  }
  console.log(i);
}

// 0
// 2
// 4
// 6
// 8
```

## switch

모든 스위치 문은 if else문으로 작성할 수 있다.

하지만 케이스가 다양할 경우 보다 간결하게 작성할 수 있다.

```
switch(평가){
  case A:
    // A일때 코드
  case B:
    // B일때 코드
    ...
}
```

위 코드를 if else로 변환 시

```
if ( 평가 == A ){
  // A일때 코드
} else if (평가 == B){
```

```

    // B일때 코드
}

-----

// 사과 100원
// 바나나 200원
// 키위 300원
// 멜론 500원
// 수박 500원
// 사고싶은 과일 물어보고 가격 알려주기

// const want = prompt('뭐살래?');
const A = '사과';
const B = '바나나';
const C = '키위';
const D = '멜론';
const E = '수박';

switch(want) {
  case A:
    console.log('100원');
    break;
  case B:
    console.log('200원');
    break;
  case C:
    console.log('300원');
    break;
  case D:
    console.log('500원');
    break;
  case E:
    console.log('500원');
    break;
  default :
    console.log ( '그런 과일은 없다.' );
}

// break를 넣지않으면 해당하는 case이후의 값이 모두 출력되어버린다.
// 해당하는 case가 없으면 default가 실행
// D와 E가 같은 실행값이므로

  case D:
  case E:
    console.log('500원');
    break;

로 변경가능

```