

제어 구문

if 문

❖ if 문 구조

```
if(조건식) {  
    조건식이 true인 경우에 실행하는 명령  
}
```

```
if(조건식) {  
    조건식이 true인 경우에 실행하는 명령  
} else {  
    조건식이 false인 경우에 실행하는 명령  
}
```

```
if(조건식1) {  
    조건식1이 true인 경우에 실행하는 명령  
} else if(조건식2) {  
    조건식2이 true인 경우에 실행하는 명령  
  
} ...  
} else {  
    모든 조건식이 false일 때 실행하는 명령  
}
```

if 문

❖ if문

```
var x = 15;
if (x >= 10) {
  console.log('변수 x는 10이상이다. ');
} else {
  console.log('변수x는 10미만이다. ');
}
```

```
var x = 30;
if (x >= 20) {
  console.log('변수x는 20이상이다. ');
} else if (x >= 10) {
  console.log('변수x는 10이상이다. ');
} else {
  console.log('변수x는 10미만이다. ');
}
```

if 문

❖ if문

```
var x = 30;
if (x >= 10) {
  if (x >= 20) {
    console.log('변수 x는 20이상이다.');
```

if문은 조건이 참일 때 실행되는 코드를 블록으로 묶어서 실행합니다. 위 예제에서는 변수 x가 10 이상인지 확인하고, 만약 20 이상이면 '변수 x는 20이상이다.'를 출력하고, 그렇지 않으면 '변수 x는 10이상 20미만이다'를 출력합니다.

```
  } else {
    console.log('변수 x는 10이상 20미만이다');
```

else문은 조건이 거짓일 때 실행되는 코드를 블록으로 묶어서 실행합니다. 위 예제에서는 변수 x가 10 미만이면 '변수 x는 10미만이다.'를 출력합니다.

```
  }
} else {
  console.log('변수 x는 10미만이다.');
```

else if문은 조건이 참일 때 실행되는 코드를 블록으로 묶어서 실행합니다. 위 예제에서는 변수 x가 10 이상 20 미만이면 '변수 x는 10이상 20미만이다'를 출력합니다.

```
}
```

switch

❖ switch

```
switch (식) {  
    case 값1:  
        '식 = 값1'인 경우에 실행되는 명령(군)  
    case 값2:  
        '식 = 값2'인 경우에 실행되는 명령(군)  
    ...  
    default:  
        식의 값이 모든 값에 조건상 일치하지 않을 경우에 실행되는 명령(군)  
}
```

switch

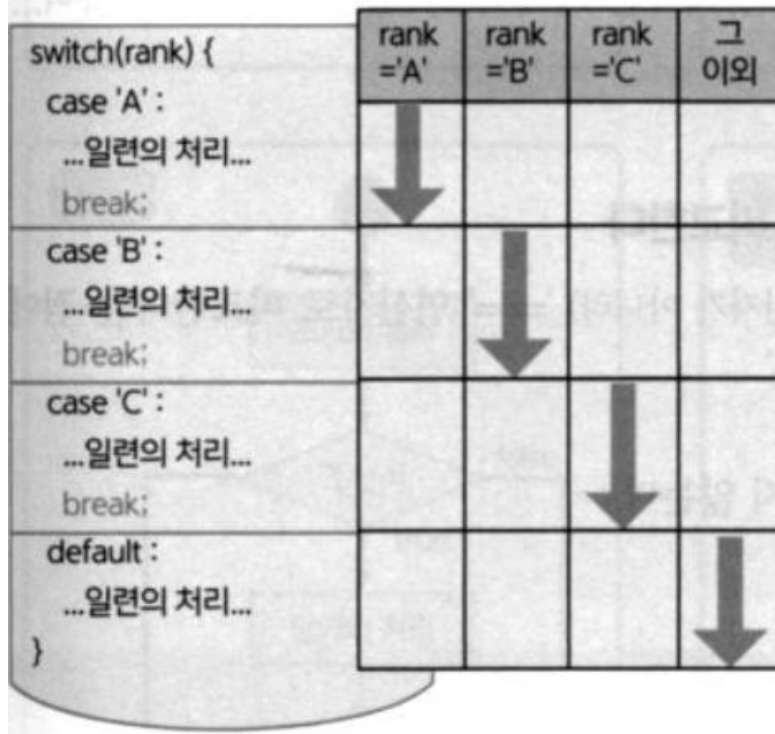
❖ switch

```
var rank = 'B';
switch(rank) {
  case 'A' :
    console.log('A랭크입니다. ');
    break;
  case 'B' :
    console.log('B랭크입니다. ');
    break;
  case 'C' :
    console.log('C랭크입니다. ');
    break;
  default :
    console.log('아무 랭크도 아닙니다. ');
    break;
}
```

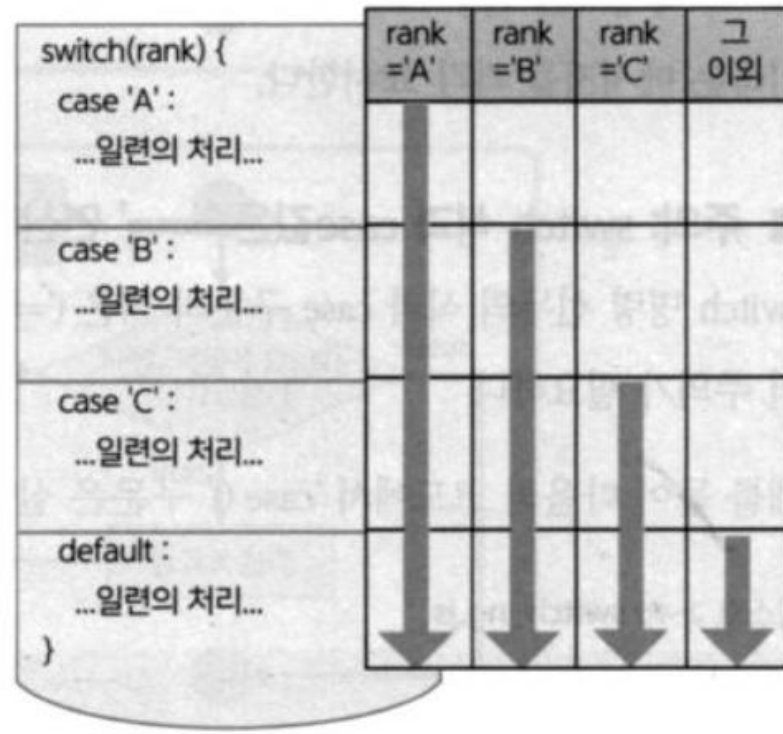
switch

❖ switch

break가 있는 경우



break가 없는 경우



switch

❖ switch

```
var rank = 'B';
switch(rank) {
  case 'A' :
  case 'B' :
  case 'C' :
    console.log('합격 ! ');
    break;
  case 'D' :
    console.log('불합격...');
    break;
}
```


while/do..while

❖ while

```
while(조건식) {  
    조건식이 true일 때 실행되는 명령(군)  
}
```

```
var x = 8;  
while(x < 10) {  
    console.log('x의 값은 ' + x);  
    x++;  
}
```

while/do..while

❖ do ... while

```
do {  
    조건식이 true일 때 실행되는 명령(군)  
} while ( 조건식 );
```

```
var x = 8;  
do {  
    console.log('x의 값은 ' + x);  
    x++;  
} while(x < 10);
```

for

❖ for

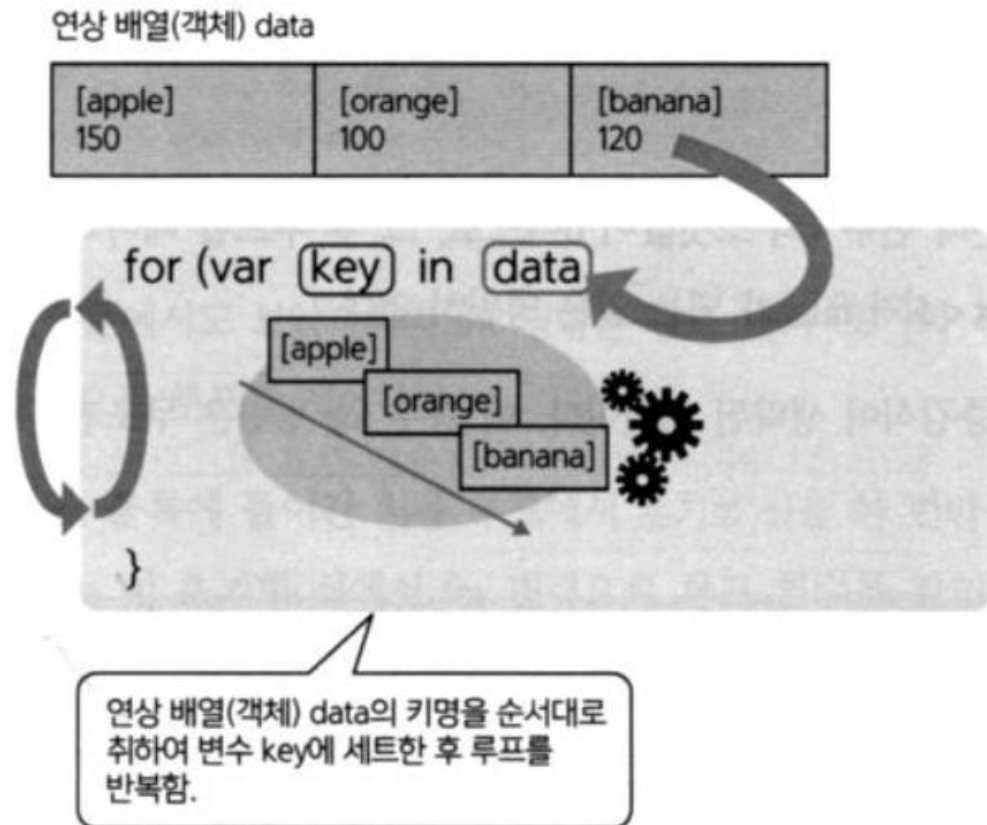
```
for (초기화 식; 루프 계속 조건식; 증감식) {  
    루프 내에서 실행하는 명령(군)  
}
```

```
for (var x = 8; x < 10; x++) {  
    console.log('x의 값은 ' + x);  
}
```

❖ for .. in

- 키를 가지고 순회
 - 배열 : 인덱스
 - 객체 : 프로퍼티명

```
for (가변수 in 연상 배열) {  
    루프 내에서 실행하는 명령(군)  
}
```



for

❖ for .. in

```
var data = {  
  apple:150,  
  orange:100,  
  banana: 120  
};  
  
for (var key in data) {  
  console.log(key + '=' + data[key]);  
}
```

```
var data = [ 'apple', 'orange', 'banana' ];  
  
for (var key in data) {  
  console.log(data[key]);  
}
```

for

❖ for ... of

- 배열/객체의 값으로 순회

```
for (가변수 of 열거 가능한 객체) {  
    루프 내에서 실행하는 명령(군)  
}
```

```
var data = [ 'apple', 'orange', 'banana' ];
```

```
for (var value of data) {  
    console.log(value);  
}
```

for

❖ break/continue

```
var result = 0;

for (var i = 1; i <= 100; i++) {
    result += i;
    if (result > 1000) { break; }
}

console.log('합계값이 1000을 넘은 것은 ' + i);
```

```
var result = 0;
for (var i = 1; i < 100; i++) {
    if (i % 2 === 0) { continue; }
    result += i;
}

console.log('합계 : ' + result);
```

for

❖ break/continue

kuku :

```
for (var i = 1; i < 10; i++) {  
  for (var j = 1; j < 10; j++) {  
    var k = i * j  
    if (k > 30) { break kuku; }  
    document.write(k + '&nbsp;');  
  }  
  document.write('<br />');  
}
```


예외 처리

❖ 예외 처리 구문

```
try {  
    예외가 발생할지 모를 명령(군)  
} catch (예외정보를 취할 변수) {  
    예외가 발생했을 시의 명령(군)  
} finally {  
    예외의 유무에 관계없이 최종적으로 실행되는 명령(군)  
}
```

예외 처리

❖ try ... catch ... finally

```
var i = 1;
try{
  i = i * j;
} catch(e) {
  console.log(e.message);
} finally {
  console.log('처리가 완료되었다.');
```

예외 처리

❖ throw

`throw new Error(에러 메시지)`

```
var x = 1;
var y = 0;
try{
  if (y === 0) { throw new Error('0으로 나누려고 하였다.'); }
  var z = x / y;
} catch(e) {
  console.log(e.message);
}
```