

Control Statement

제어문

Control Statement

제어문 : 조건식을 사용해 코드의 흐름을 제어하는 문법

조건문 : if
 if – else
 if – else if – else
 switch

반복문 : for
 while
 do - while

Control Statement - 조건문

if문 : 조건식의 결과가 참이면 주어진 문장을 실행

```
if(조건식){  
    //조건식이 참일 때 실행 될 문장  
}
```

if - else문

```
if(조건식){  
    //조건식이 참일 때 실행 될 문장  
}else{  
    //조건식이 거짓일 때 실행 될 문장  
}
```

Control Statement - 조건문

다중 조건문

```
if(조건식1){  
    //조건식1이 참일 때 실행 될 문장  
}else if(조건식2){    //조건식1이 거짓일 경우 조건식2 검증  
    //조건식2가 참일 때 실행 될 문장  
}else if(조건식3){    //조건식2가 거짓일 경우 조건식3 검증  
    //조건식3이 참일 때 실행 될 문장  
}else{  
    //조건문의 모든 조건식이 거짓일 때 실행 될 문장  
}
```

중첩 조건문 : 모든 조건문의 형태로 중첩 가능, 중첩 개수에 제한 없음

```
if(조건식1){  
    //조건식1이 참일 때 실행 될 문장  
}else if(조건식2){    //조건식1이 거짓일 경우 조건식2 검증  
    //조건식2가 참일 때 실행 될 문장  
    if(조건식3){  
        //조건식2가 참이면서 조건식 3이 참일 때 실행 될 문장  
    }  
}else{  
    //조건문의 모든 조건식이 거짓일 때 실행 될 문장  
}
```

Control Statement - 조건문

switch문 : 조건 값에 따라 주어진 문장을 실행

```
switch(조건 값){ //정수형, 문자, 문자열, Enum만 가능
    case 값1 : //조건 값이 값1이면 실행
        break;    //switch case문 탈출
    case 값2 : //조건 값이 값2이면 실행
        break;    //switch case문 탈출
    case 값3 : //조건 값이 값3이면 실행
        break;    //switch case문 탈출
    default :    //case에 일치하는 조건 값이 없으면 실행
}
}
```

* Fall-Through문 : 조건 값에 해당하는 case절에 break가 없으면 해당 case절 뿐 아니라 다음 break를 만나기 전까 case절과 default절을 실행한다.

의도적으로 break를 생략해 모든 case절과 default절을 실행시키는 것을 Fall-Through문이라 한다.

코드를 읽는 중 혼란스러울 여지가 있으므로 사용하지 않는 것을 권장.

Control Statement - 반복문

while문 : 조건식의 결과가 참이면 주어진 문장을 반복 실행, 결과가 거짓이면 반복 종료

```
while(조건식){ //조건식을 검증

    //조건식이 참일 때 실행 될 문장
    //문장의 실행이 끝나면 조건식으로 이동
}
```

5번 반복하는 while문 예시

```
int i = 0;        //반복 횟수를 기억할 int형 변수 선언
while(i < 5){      //반복 횟수가 5회 미만인지 확인

    /*
     * 수행할 문장
     * ...
     */

    i++;          //반복 횟수를 1회 증가
}
```

Control Statement - 반복문

do-while문 : 조건식의 결과가 참이면 주어진 문장을 반복 실행, 결과가 거짓이면 반복 종료
while문과 달리 먼저 문장을 실행하고 조건식을 판단.

```
do{  
    //첫 실행 때는 조건식 검증 없이 문장을 실행  
    //2회차 부터는 조건식이 참일 때만 문장을 실행  
  
} while(조건식); //문장의 실행이 끝나면 조건식을 검증
```

5번 반복하는 do-while문 예시

```
int i = 0;        //반복 횟수를 기억할 int형 변수 선언  
do{  
  
    /*  
    * 수행할 문장  
    * ...  
    */  
  
    i++;          //반복 횟수를 1회 증가  
}while(i < 5);    //반복 횟수가 5회 미만인지 확인
```

Control Statement - 반복문

for문 : n회 반복하는 while문을 보다 편리하게 사용하기 위해 제공되는 반복문

```
for( 초기식; 조건식; 증감식){  
    //조건식이 참일 때 실행될 문장  
}
```

5번 반복하는 for문 예시

```
for (int i = 0; i < 5; i++ ) {  
    //수행할 문장  
}
```

초기식 int i = 0 : 반복 횟수를 기억할 int형 변수 선언

조건식 i < 5 : 반복 횟수가 5회 미만인지 확인

증감식 i++ : 반복 횟수를 1회 증가

Control Statement - 반복문

while, do-while과 for문의 차이

1. 지정된 n회 반복을 수행해야 하는 경우 : for문
2. 무한반복문이나, 특정 조건이 충족 될 때 까지 반복하는 경우 : while, do-while

ex) 사용자가 "exit" 명령어를 입력할 때 까지 반복
다른 서버로 부터 모든 리소스를 받아올 때 까지 반복

Control Statement – break, continue

break : 조건문이나 반복문을 종료

continue: 반복문 블록 안의 문장 수행을 건너뛰

while문 : 블록안의 문장 수행을 건너뛰고 조건식을 검증

for문 : 블록안의 문장 수행을 건너뛰고 증감식을 실행