



조건문과 반복문





- 자바에서 활용되는 조건문과 반복문에 대하여 기본 구조를 이해한다.
 - 조건문의 종류(**if, switch**)를 상황에 따라 효과적으로 사용할 수 있다.
 - 반복문의 활용되는 코드와 해당 내용의 **keyword**를 사용할 수 있다.
 - 반복문(**for, while, do~while**)의 종류에 따른 사용 방법을 안다.
 - 제어문(**break, continue**)가 활용되는 곳을 알고, 사용할 수 있다.
-



statement의 종류 :

- 자바 소스코드로 활용되는 것을 **statement(문)**이라고 하는데, 문의 종류는 다음 같이 분류할 수 있다.

문(statement)

실행문

순차문

제어문

분기문(조건)

반복문

비실행문

주석(//, /**/)

- 실행문

```
int num01 = 25;  
데이터type 변수명 = 할당데이터;
```



- 단일 if
 - if(비교연산자, 논리연산자 → true/false){
 - 해당 조건이 true일 때, 처리될 내용..
 - }
 - if else
 - if(조건 boolean){
 - true일때, 처리될 내용
 - }else{
 - false일 때, 처리될 내용
 - }
-



- 다중 if
 - if(조건){
 - if(조건){
 - }
 - }else if(조건){
 - }else{
 - }



분기문 예제 :

```
// 조건문에 대한 처리..
// 0, 1, 2
// 가위 바위 보 출력 되는 내용..
// (int) casting!!
// Math.random()*3 실수..
int gameIdx = (int)(Math.random()*3); // 범위가 0~2까지만..
// 해당 조건
if( gameIdx==0 ){
System.out.println("가위");
// 위의 조건을 제외하고 1 일때..
}else if(gameIdx==1){
System.out.println("바위");
}else{
System.out.println("보");
}
```



switch :

- switch(변수){
 - case 변수가 값1일때 : 처리할 내용 ; break;
 - case 변수가 값2일때 : 처리할 내용 ; break;
 - default : 위 case문 이외에 처리할 내용;
 - }
 - switch(변수)에 들어갈 데이터 type
 - 정수형 ex) int choiceBtn=5;
switch(choiceBtn){}
 - 문자형 ex) char grade='B'; switch(grade){}
 - 문자열형(자바7~) ex) String pos="부장";
switch(pos){}
-



switch문 예제 :

```
// switch( 변수 ) : 정수, char, 문자열
int score = (int)(Math.random()*101); // 0~100 랜덤수 발생..
char ptGrade; // 등급..
switch( score/10 ){ // 정수/정수 ==> 정수, 정수/실수 ==> 실수,
// 실수/정수 ==> 실수
case 10: // break가 없으면 다음 flow로 흘러간다.
    // ptGrade = 'A'가 처리 됨.
case 9: ptGrade = 'A'; break;
case 8: ptGrade = 'B'; break;
case 7: ptGrade = 'C'; break;
case 6: ptGrade = 'D'; break;
default : ptGrade = 'F';
}
System.out.println("점수는 "+ score + ", 학점등급은 "+ ptGrade);
```




switch문 예제 :

```
/* 문자열 처리 switch*/  
//String [] compos=new String[]{"", "", ""};  
String [] compos = {"사원", "대리", "과장", "차장", "부장"};  
// 배열의 경우만 처리 되게..  
int ranPosIdx = (int)(Math.random()*compos.length);  
String chPos = compos[ ranPosIdx ];  
System.out.println("선택된 직위:"+chPos);  
switch(chPos){  
    case "사원": System.out.println("200~300"); break;  
    case "대리": System.out.println("240~350"); break;  
    case "과장": System.out.println("300~650"); break;  
    case "차장": System.out.println("400~700"); break;  
    case "부장": System.out.println("450~"); break;  
}
```



for문(반복문):

- 반복처리에서 특정한 변수가 필요로 하고, 제어처리를 할 때 가장 많이 활용되는 문이 for문이다.
- for(초기값 ; 반복할 조건,한계 ; 증감식){
 - 반복할 구문 (선언한 변수-초기값 활용)
 - 초기값 : `int count = 1; int idx=0; cnt = 100;`
 - 반복할 조건 :
 - `count < 1000`(증가);
 - `cnt > 0`;(감소)
 - `idx < 배열.length`
 - 증감식
 - `count++`, `cnt--` (1씩)
 - `count+=2`, `cnt-=5` (단위로 처리)



for문(반복문)예제 :

```
// 15 ~ 50 출력..  
for(int cnt=15;cnt<=50;cnt++){  
    System.out.println("번호:"+cnt);  
}  
// 5개 단위로 tab 및 줄바꿈 처리..  
for(int cnt=15;cnt<=50;cnt++){  
    // print() 줄바꿈 없음.. \t : 탭, \n : 줄바꿈처리..  
    System.out.print(cnt);  
    if(cnt%5==0){  
        // println() : 줄바꿈  
        System.out.println();  
    }else{  
        System.out.print("\t");  
    }  
}
```



for문(반복문)(예제):

```
// for문을 통한 데이터 연산 처리..  
// 1~100까지 데이터를 합산하세요!!  
// 합산할 변수 선언  
int sum=0; // 누적처리할 변수+=추가할 데이터;  
for(int cnt=1;cnt<=100;cnt++){  
    System.out.print(cnt);  
    if(cnt!=100){  
        System.out.print(" + ");  
    }  
    sum+=cnt; // 데이터 누적처리..  
}  
System.out.println(" = "+ sum);  
}
```



for문2 (반복문):

- 배열에 효과적으로 활용되는 for문 2번째 형식
 - 일반 primitive data배열, 객체 배열
- String
 - `names={"홍길동","김길동","신길동"};`
 - `names[0]` : "홍길동"이라는 데이터가 들어가 있는
단위 변수나 객체 : `String name= names[0];`
- `for(단위 객체 선언 : 배열객체){`
 - 배열의 단위객체를 활용해서 처리 가능
 - `for(String name : names){`
 - `System.out.println("이름:"+name);`
 - `}`



for문(반복문)(예제) :

```
String[] foods={"사과","짜장면","된장찌개"};
// 배열명[index번호] idx=0;부터시작
// 배열명.length : 배열의 크기 = index마지막번호 +1
// number(1~ ) = index +1
for(int idx=0;idx<foods.length;idx++){
    System.out.println((idx+1)+" "+ foods[idx]);
}
// for(단위변수/객체 : 배열객체 ){
//     해당 배열객체가 있을 때까지 단위 변수나 객체 할당
for(String food:foods){
    System.out.println("배열 for2 데이터:"+food);
}
```



for2 확인예제 :

- 물품 list 배열을 만들고,
- 가격 list 배열을 만들어서,
- 각각 **for 2**번째 형식을 이용하여 출력
- 마지막 **for 1**번째 형식을 이용해서 통합 출력

| 라면 |
|-----|
| 파 |
| 소고기 |
| 과자 |

| 5000 |
|-------|
| 2000 |
| 22000 |
| 4000 |

| 라면 | 5000 |
|-----|-------|
| 파 | 2000 |
| 소고기 | 22000 |
| 과자 | 4000 |



for2 확인예제 정답 :

```
String []products={"라면","파","소고기","과자"};
int []prices={5000,2000,22000,4000};
System.out.println("### 물건명###");
for(String product:products){
    System.out.println(product);
}
System.out.println("### 가격###");
for(int price:prices){
    System.out.println(price);
}
System.out.println("### 물건명/ 가격###");
for(int idx=0;idx<products.length;idx++){
    System.out.println(products[idx]+"\\t"+prices[idx]);
}
```




이중 for문 :

- 2중 배열이나, 특정 데이터의 2중적인 처리를 위해서 필요한 것을 이중 for문을 활용한다..

for 조건식(외부)

외부변수 outIdx = 0;

for 조건식(내부)

내부변수 : 외부변수에 영향을 미침



이중 **for**문 예제 :

```
/*구구단을 통한 2중 for문 처리
* */
// 외부는 단수에 관련된 내용 처리..
for(int grade=2;grade<=9;grade++){
// 내부는 cnt 1~9 연산 처리
System.out.println("###"+grade+"단 ###");
for(int cnt=1;cnt<=9;cnt++){
System.out.println(grade + " * " + cnt + " = " +
(grade*cnt));
}
}
```



2중 for문 확인예제 :

- 구구단을 2중 for문을 이용해서 가로로 처리..

| 2단 | 3단 | 4단 | 5단 | 6단 | 7단 | 8단 | 9단 |
|------------------------------|--------------------|--------------------|---------------------|---------------------|---------------------|---------------------|------------------------------------|
| $2*1=2$ $2*2=4$. . | $3*1=3$ $3*2=6$ | $4*1=4$ $4*2=8$ | $5*1=5$ $5*2=10$ | $6*1=6$ $6*2=12$ | $7*1=7$ $7*2=14$ | $8*1=8$ $8*2=16$ | $9*1=9$ $9*2=18$. . . |



확인예제 풀이 :

```
// 외부는 cnt 1~9 연산 처리
for(int cnt=0;cnt<=9;cnt++){
    // 내부는 단수에 관련된 내용 처리..
    for(int grade=2;grade<=9;grade++){
        // 초기 @@ 단 title 처리
        if(cnt==0){
            System.out.print(" "+grade+"단\t\t");
        }else{
            System.out.print(grade + " * " + cnt + " = " +
                (grade*cnt)+"\t");
        }
    }
    // 줄바꿈 처리..
    System.out.println();
}
```



while, do while

- 무한 반복 처리되는 demon을 활용할 때, 주로 사용하는 반복문 이다.
- 형식
 - while(조건식-boolean){
 - 반복처리할 내용!!
 - }
 - do{
 - 반복할 내용이지만, 초기에 한번은 반복조건이 맞지 않더라도 수행될 내용.
 - }while(조건식 - boolean);



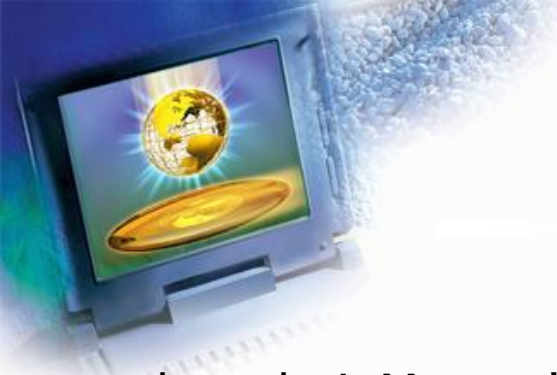
while문 예제.:

```
// System.in : java에서 지원하는 표준 입력 Stream
// 입력관련된 내용은 반드시 예외 처리를 해 주어야 한다.
// throws IOException
// 0 : 48~..., A : 65 ~~, a:97, <- 37, 38, 39, 40
int keyCode;
while(true){
    keyCode=System.in.read();
    System.out.println(keyCode);
    if(keyCode==81){
        System.out.println("중단!!");
        // break : 반복문에서 프로세스를 중단처리..
        break;
    }
}
```



while문 예제.:

```
char inputChar;  
// 무한으로 도는 내용에서 특정 문자를 입력했을 때,  
// 종료 처리..  
while(true){  
    inputChar=(char)System.in.read();  
    System.out.println("입력된 문자:"+inputChar);  
    if(inputChar=='Q'){  
        System.out.println("종단합니다!!");  
        break;  
    }  
}
```



do while :

```
char choieMenu='Q';
String choMenu="";
System.out.println("메뉴를 선택하세요!(번호입력)");
System.out.println("1) 짜장면");
System.out.println("2) 짬뽕");
System.out.println("3) 탕수육");
System.out.println("종료하실려면 Q를 입력!!");
do{

    choieMenu=(char)System.in.read();
    if(choieMenu=='1') choMenu+="짜장면 선택!!\n";
    if(choieMenu=='2') choMenu+="짬뽕 선택!!\n";
    if(choieMenu=='3') choMenu+="탕수육 선택!!\n";

}while(choieMenu!='Q');
System.out.println("주문된 내용");
System.out.println(choMenu);
System.out.println("주문 완료!!!");
```



break 와 continue

■ break

- loop문, switch문에서 해당 block(`{ }`)을 중단처리해서 빠져나오겠끔 처리하는 것을 말한다.
- 더 이상 해당 block에서 프로세스가 처리되지 않음..
- while(){
 - if(조건)
 - break;

■ continue

- loop문에서 조건에 의한 continue가 나타날 때, 해당 프로세스가 다음 loop단위 넘어 가는 것을 말한다.
- ex) `for(int cnt=1; cnt<=10 ;cnt++){`
 - `if(cnt == 5) continue;`
 - `System.out.println("넘버:"+cnt);`
 - 넘버 5 인경우만 출력되지 않고, 다른 내용은 출력 처리.



break와 continue 예제 :

```
System.out.println("## 사과먹는 곰돌이###");
for(int cnt=1;cnt<=10;cnt++){
    System.out.println("곰돌이가 사과를 "+cnt+"개 먹었다.");
    if(cnt==5){
        System.out.println("5개 먹는 순간!! 이제 그만!!");
        break;
    }
}
System.out.println("break 예제 완료!!!");
// continue 해당 블록({})에 특정 프로세스만 pass 처리..
System.out.println("### 떡먹는 병만이 ###");
for(int cnt=1; cnt<=10;cnt++){
    if(cnt==6){
        System.out.println("6번째 떡은 신지에게 빼앗김!!!");
        continue;
    }
    System.out.println("병만이 "+ cnt+"번째 떡을 먹다!!!");
}
System.out.println("continue와 병만이 예제 완료!!");
```



확인 및 정리 :

1. 조건문과 반복문의 종류를 괄호 속에 넣어 보세요
 - 조건문 : (), ()
 - 반복문 : (), (), ()
2. for문을 이용하여 10~150까지 정수 중에 5의 배수의 총합을 구하는 코드를 작성하세요.
3. while문, Math.random()을 이용하여, 주사위 숫자 두개가 나오게 하고, 주사위 합이 5가 아니면 계속, 합이 5이면 중단 처리되게 하세요.
 - 출력형식 @@@, @@@ 합 @@@
4. 이중 for문을 이용하여 방정식 $4x + 5y = 60$ 인 모든 해를 구하세요 x, y는 10이하 정수
 - 출력형식 x: @@, y : @@



감사합니다 !
