

<p>다중 선택이 가능한 switch ~ case 조건문</p> <pre> 형식) switch(정수식) { case 값2 : 처리할 문장 2; break; ...중략 default: 처리할 문장; } </pre> <p>정수식과 case문의 값과 일치하면 해당 처리할 문장이 실행된다. 그리고 문장 실행 후 break문을 만나면 switch~case문을 종료 한다.</p> <p>그리고 default문은 해당 조건이 없을때 실행된다.</p> <p>참고) jdk1.7 이후부터는 switch 문의 조건식에도 문자열이 올수 있다. case문의 값도 문자열이 올 수 있다.</p>
--

<p>for 반복문</p> <pre> 형식) for(초기치;조건식;증감식){ 조건식이 참일 동안만 반복; } </pre> <p>가) 처음엔 초기값을 실행한다. 초기값이란 변수에 최초의 값을 저장하는 것을 말한다.</p> <p>나) 조건식을 평가하여 참일 동안 문장을 반복 실행한다.</p> <p>다) 증감식을 실행한다. 다시 조건식을 평가하여 참이면 문장을 반복하고, 거짓이면 for반복문을 종료 한다.</p>
<p>향상된 for문(확장 for문)</p> <p>가) 향상된 for문은 J D K 1.5부터 배열과 컬렉션에 저장된 요소에 접근할 때 기존 for문보다 편리한 방법으로 처리할 수 있도록 새롭게 추가되었다.</p> <p>나) 확장 for문 형식</p> <pre> for(타입 변수명:배열 또는 컬렉션){ //배열 또는 컬렉션에 저장된 값이 매 반복마다 하나씩 순서대로 읽혀져 //변수에 저장된다. } </pre> <p>다) 확장 for문은 일반적인 for문과 달리 배열과 컬렉션에 저장된 요소들을 읽어오는 용도로만 사용할 수 있다.</p>

while 반복문
형식) <pre> while(조건식){ 조건식이 참일 동안만 반복; 증감식; } </pre>

무한 루프란?
가. 무한 루프란 조건식이 무조건 참이어서 영원히 반복하는 반복문을 뜻한다.

do~while 반복문
형식) <pre> do{ 조건식이 참일 동안만 반복; 증감식; }while(조건식); </pre> 주의 사항) 나중에 조건식을 검사하기 때문에 조건식이 거짓이라도 무조건 한번은 반복한다는 단점이 있다. 실제 자바 프로젝트에서는 do while반복문은 잘 사용 되지 않는다.

break, continue문
가. break문: 반복문 내에서 break문을 만나면 반복문을 중단한다. break 자신과 가장 가까운 반복문을 벗어난다. 나. continue문: 반복문안에서 이 문을 만나면 아래문장을 실행하지 않고, 다음 반복을 위해서 반복문 처음으로 돌아가서 그 다음 반복을 계속한다. 즉,continue문과 반복문 블록의 끝 ‘}’ 사이의 문장들을 건너뛰고 반복을 이어가는 것이다.