**Web개발\_Day05\_노트정리**

**제어문, Control Statement**

ㆍ반복문의 종류 : for(), while(), do~while() 등이 있다.

ㆍ사용자가 처리하기 부담스러운 반복작업을 처리하기에 용이하다.

* for(), while() : 조건에 따라 한번도 실행되지 않을 수도 있다.
* Do{}while(); : 조건에 상관없이 무조건 최소한 한번 실행되는 것을 보장한다.

ㆍ각 loop의 사용차이 :

* for() 는 반복 “횟수” 기반으로 사용한다.
* While() 은 반복 “조건” 기반으로 사용하며, 횟수가 불분명할 때 사용하기 용이하다.
* 두 loop는 서로 유사하여 서로 변환하여 구현이 가능하다.

**반복문3, do~while()**

ㆍwhile의 변형, 기본적인 구조는 같다.

ㆍwhile()은 단 한번도 수행되지 않을 수 있으나, do~while()은 최소한 한번은 반드시 실행 될 것을 보장한다.

ㆍdo{

…조건이 true일 때 실행할 구문

} while ( 조건식 ) ;

**Break 커맨드**

ㆍ라인을 순차적으로 실행도중, break를 만나면 해당 반복문을 즉시 종료한다.

ㆍloop를 도는 중 if (~이러한 조건을 만난다면) break; 의 형식으로 많이 사용된다.

* 해당 조건 만족시 break = 반복문을 즉시 종료한다.

**Continue 커맨드**

ㆍbreak 와 비슷하게 라인을 순차적으로 실행도중, continue를 만나면 해당 구문의 아래 라인들을 무시하고 넘어간다

* 반복문에서 사용될 경우, continue를 만나는 순간, 아래의 라인들을 건너뛰고 다음 반복으로 넘어간다.

**이중 반복문의 break**

ㆍ이중 반복문에서 내부 반복문에 break; 가 들어간다면, 내부의 반복문만 종료되고 외부의 반복문은 계속 실행된다.

ㆍ내부의 break를 만날 때 외부의 반복문을 종료시키려면?

* 반복문에 이름을 붙이면 반복을 건너뛸 수 있다.
* 이름지정!
* 루프명 : for () {

For () {

If () break 루프명;

}

}

* 외부 반복문에 “루프명”을 지정해주고, “break 루프명”을 실행함으로써, 외부의 반복문도 종료한다.

**반복문 내부 변수의 특징**

ㆍ자신보다 하위 블록으로는 침투 할 수 있다.

* 이중 for()문을 사용한다는 가정 하에, 외부 for()문에서만 정의한 변수는 내부 for()문에서 사용할 수 있다.

ㆍ자신이 선언된 블록 바깥으로는 빠져나갈 수 없다.

* 이중 for()문을 사용한다는 가정 하에, 내부 for()문에서만 정의한 변수는 외부 for()문 에서 사용할 수 없다.

ㆍ블록안에서 선언된 변수는 밖에서 존재하는 동일한 이름의 변수와는 이름만 동일할 뿐, 다른 값으로 인식된다.