기본 문법 (Syntax)

```
for (초기화식; 종결 제어식; 증감 제어식) {
  // 수행부분
}
```

for문의 조건 부분은 세 가지로 나누어 집니다:

1. 초기화식 (initialization)
2. 종결 제어식 (termination)
3. 증감 제어식 (increment)

for문을 이용해서 `1` 부터 `10` 까지 출력하려면 이렇게 하면 됩니다.

```
for (int i = 1; i <= 10; i++) {
    System.out.println(i)
}
```

```
1
2
3
4
5
6
7
8
9
10
```

1. 초기화식에서 `i` 라는 정수 변수를 선언해주고 `1` 을 지정해줍니다.
2. `i` 가 `10` 보다 작거나 같을 동안 반복합니다.
3. 수행 부분이 끝나면 `i` 를 `1` 늘려줍니다.

Q. for문은 언제 사용하나요?

for문은 while문과 달리 초기화식이 있고 for문 안에서만 쓸 수 있는 변수를 만들 수 있습니다. 이러한 장점 때문에 for문은,

1. 반복의 인덱스가 필요한 경우
2. 반복의 최대 횟수가 정해진 경우
3. 갯수가 정해진 데이터 셋(배열, 리스트 등)의 내용을 하나씩 봐야할 경우

에 주로 사용합니다. (3)번에 대해서는 배열 강의에서 좀 더 살펴보겠습니다.

물론 while문으로 작성된 내용을 for문으로, for문으로 작성된 내용을 while문으로 작성할 수 있지만 더 자주 쓰이는, 더 직관적이고 편한 경우들이 있겠죠?

예제

1 부터 100까지의 합을 구하시오.

```
int sum = 0;
for (int i = 1; i <= 100; i++) {
    sum += i;
}
System.out.println(sum);
```

반복문 밖에서 **`int`** 변수 **`sum`** 을 만들어 주고 **`0`** 으로 초기화 해줍니다.

그리고 반복문을 작성하는데, 내부에서만 사용할 변수 **`i`** 를 초기화식에 써주고 **`i`** 를 하나씩 늘려나가며 더하고 **`100`** 까지만 더하기로 합시다. 어렵지 않죠?

for문의 강력한 힘은 배열, 리스트등의 자료형과 함께있을 때 더 잘 드러납니다. 배열 강의에서 for문을 좀 더 다루니 집중해서 들으시길 바랍니다!

☑ 수업을 완료하셨으면 체크해주세요.

수강생 Q&A 보기 (/questions?assignment_id=267&sort_by=popular)

질문하기 (/questions/new?assignment_id=267&op1=%EA%B0%9D%EC%B2%B4+%EC%A7

이전 강의 for문 (/assignments/249)

다음 강의 구구단 (/assignments/274)