

"I can code too!" 라는 문자열을 10번 출력하려면 `print("I can code too!")` 라는 코드를 10번 반복해서 쓰면 됩니다.

```
print("I can code too!")
print("I can code too!")
print("I can code too!")
print("I can code too!")
print("I can code too!")
print("I can code too!")
print("I can code too!")
print("I can code too!")
print("I can code too!")
print("I can code too!")
```

```
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
```

하지만 똑같은 코드를 반복하는 것은 프로그래머로서 용납할 수 없는 일입니다. 이럴때에는 **while** 반복문을 떠올려야 합니다.

while 문의 문법 구조는 다음과 같습니다:

1. **while** 조건부분:
2. 실행할 문장 1
3. 실행할 문장 2
4. 실행할 문장 3

while 반복문은 조건부분(header)과 수행부분(body)으로 나뉩니다. 위의 문법 구조에서 (1) 번 줄이 조건부분에, (2) ~ (4) 번 줄이 수행부분에 해당됩니다.

파이썬이 **while** 반복문의 조건부분에 도착하면 다음과 같은 일들이 발생합니다:

1. 파이썬은 조건부분의 결과값(불린값)이 **True** 인지 **False** 인지 체크합니다.
2. 조건부분이 **True** 면, 파이썬은 반복문의 수행부분을 실행합니다. 그리고 수행부분이 끝나면 다시 조건부분으로 돌아갑니다.
3. 조건부분이 **False** 면, 파이썬은 **while** 반복문의 수행부분을 건너 뛰어 그 이후의 코드를 실행합

1111

`while` 문의 문법 구조를 참고하여, "**I can code too!**" 를 10번 출력하는 프로그램을 작성해보도록 합시다.

```
1. i = 1
2. while i <= 10:
3.     print("I can code too!")
4.     i = i + 1
```

```
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
I can code too!
```

1. (1) 번 줄에서 변수 `i` 에 1의 값이 지정됩니다.
2. (2) 번 줄에서 `i <= 10` 는 `True` 이기 때문에 파이썬은 `while` 문의 수행부분으로 들어갑니다.
3. (3) 번 줄에서 콘솔에 "**I can code too!**" 라는 문자열이 출력됩니다.
4. (4) 번 줄에서 변수 `i` 의 값에 1이 추가됩니다. 지정 연산자 기억하시죠?
5. 조건부분이 `False` 가 될 때까지, 즉 `i` 가 10보다 큰 값을 갖게 될 때까지, (2) ~ (4) 번의 과정이 반복됩니다.
6. 조건부분이 `False` 가 되면, 파이썬은 반복문을 빠져나오고 프로그램은 끝납니다.

위 예시는 간단하기 때문에, 누군가는 "이 단순한 예시에 왜 복잡하게 반복문까지 쓸까?"하는 생각을 할 수도 있습니다. 하지만 "**I can code too!**" 를 100번 출력하는 프로그램을 만들려고 한다면 어떻게 해야 할까요? `print` 문만 사용하면 100개의 `print` 문을 직접 써야하지만, 반복문을 사용하면 위의 코드에서 숫자 10을 100으로 바꾸어주기만 하면 됩니다.

이제 반복문을 쓰는 이유를 아시겠나요? 반복문은 귀찮은 작업을 컴퓨터에게 떠넘기기 위한 용도입니다. 컴퓨터는 반복적인 일을 아무런 불평없이 하는 최강의 도구 아닙니까?

좋은 소프트웨어 엔지니어가 되려면 게을러져야 합니다. 귀찮은 일을 싫어해야 합니다. 역설적이죠? 100줄의 코드를 직접 쓰기보다, 컴퓨터에게 맡기기를 원해야 합니다. 매일 2시간씩을 들여 이름, 나이, 전화번호, 주소 등의 정보를 직접 분류하기보다, 총 10시간을 들여 코딩을 한 후 컴퓨터가 매일 알아서 해주기를 원해야 합니다.

컴퓨터 과학에서 다루는 여러 도구(tool)들을 배워서, 게으른 소프트웨어 엔지니어가 되세요. 어떻게 하면 똑똑한 프로그램을 작성할 수 있을지, 효율적(efficient)인 알고리즘을 작성할 수 있을지 끊임없이 고민해보세요.

유상무 잘 생겼다

호미빙의 CEO인 개그맨 유상무씨가 생일 이벤트를 열었었습니다. 네이버에 '유상무 잘생겼다'를 검색해서 검색한 순위의 끝 두자리가 35이면 아메리카노 무료 기프티콘을 주는 이벤트였죠. 그 당시 하루 동

안의 종 검색수가 무려 8천만번이었다고 합니다. 어마어마하죠? 유상무씨는 그로 인해 음료값 8억원과 메시지 발송비용 8천만원이 들었다고 합니다.

방송에서는 "파산할 지경이었다", "돈이 많이 들었다"고 이야기하기 하지만, 그만큼 호미빙의 홍보 효과도 어마어마했겠죠? 어찌되었든 아주 재미있는 이벤트였습니다.

이 이벤트에 당첨되기 위해 사람들은 계속해서 '유상무 잘생겼다'를 검색했어요. 확률이 **1/100** 이니까요. 그렇지만 소프트웨어 엔지니어들은 **1/100**의 확률을 **1**로 만들 수 있는 능력자입니다. 당첨될 때까지 컴퓨터가 스스로 검색하도록 프로그램을 만들 수 있기 때문이죠. 실제로 개발자 이두희씨는 페이스북에 '자동으로 검색해서 번호를 뽑아내는 걸 봤다. 지금부터 돌리면 아메리카노 2015년 내내 공짜로 마실 수 있을 것 같다.'고 글을 작성했었습니다. 어떻게 이런 코드를 작성한 것일까요?

```
while True:
```

```
    네이버에 "유상무 잘생겼다"를 검색한다
```

위는 파이썬으로 그 코드를 아주 단순화하여 나타낸 것입니다. 만약 이벤트에 딱 세번만 당첨되고 싶다면 어떻게 할까요?

```
coffee_count = 0
```

```
while coffee_count는 3보다 작다:
```


```
    네이버에 "유상무 잘생겼다"를 검색한다
```


```
    당첨되면 coffee_count의 값을 1 늘린다
```

위의 코드에는 당첨되면 이라는 조건이 있죠? 이 코드를 보다 자세하게 작성하기 위해서는 **if** 문이 필요합니다. 다음 강의에서 이에 대해 자세히 배워보도록 하겠습니다.



수업을 완료하셨으면 체크해주세요.

 수강생 Q&A 보기

 [/questions?](/questions?assignment_id=61&sort_by=popular)
질문하기

[assignment_id=61&sort_by=popular\)](/questions/new?assignment_id=61&sort_by=popular)
[/questions/new?](/questions/new?assignment_id=61&sort_by=popular)

[assignment_id=61&op1=%ED%94%84%EB%A1%9C%EA%B7%B](/questions/new?assignment_id=61&op1=%ED%94%84%EB%A1%9C%EA%B7%B)

< 이전 강의
유상무 잘생겼다 [\(/assignments/60\)](/assignments/60)

다음 강의 > [\(/assignments/62\)](/assignments/62)
while문 연습 1