

UNIT 17

while 반복문으로 Hello,
world! 100번 출력하기

17 while반복문으로Hello,world!100번출력하기

» while 반복문으로 Hello, world! 100번 출력하기

- while 반복문에 대해 알아보자
- while 반복문은 조건식으로만 동작하며 반복할 코드 안에 조건식에 영향을 주는 변화식이 들어감

```
i = 0                # 초기식
while i < 100:        # while 조건식
    print('Hello, world!') # 반복할 코드
    i += 1            # 변화식
```

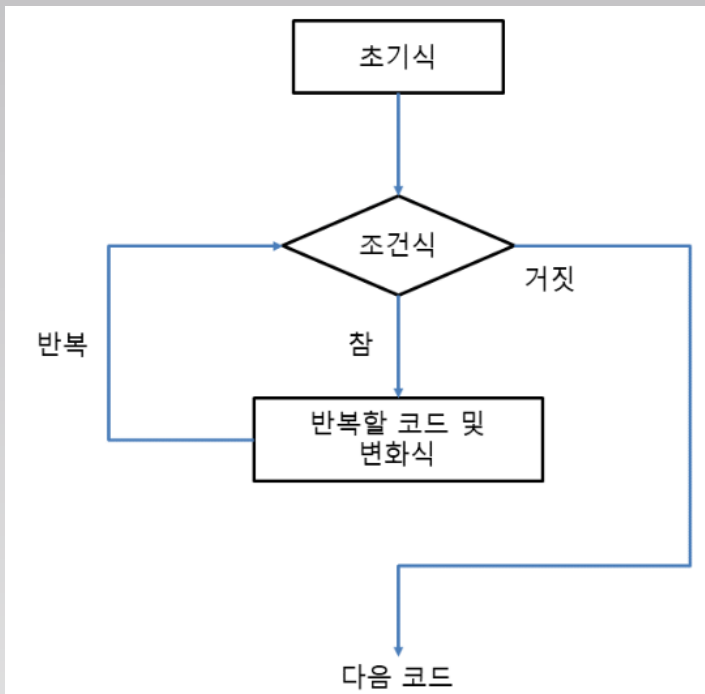
- while 반복문의 실행 과정임
- 초기식부터 시작하여 조건식을 판별함
- 조건식이 참(True)이면 반복할 코드와 변화식을 함께 수행함
- 조건식을 판별하여 참(True)이면 코드를 계속 반복하고, 거짓(False)이면 반복문을 끝낸 뒤 다음 코드를 실행함

17 while반복문으로Hello,world!100번출력하기

» while 반복문으로 Hello, world! 100번 출력하기

■ 조건식 → 반복할 코드 및 변화식 → 조건식으로 순환하는 부분이 루프(loop)임

▼ 그림 while 반복문의 동작 과정



17.1 while 반복문 사용하기

» while 반복문 사용하기

초기식

`while` 조건식:

반복할 코드

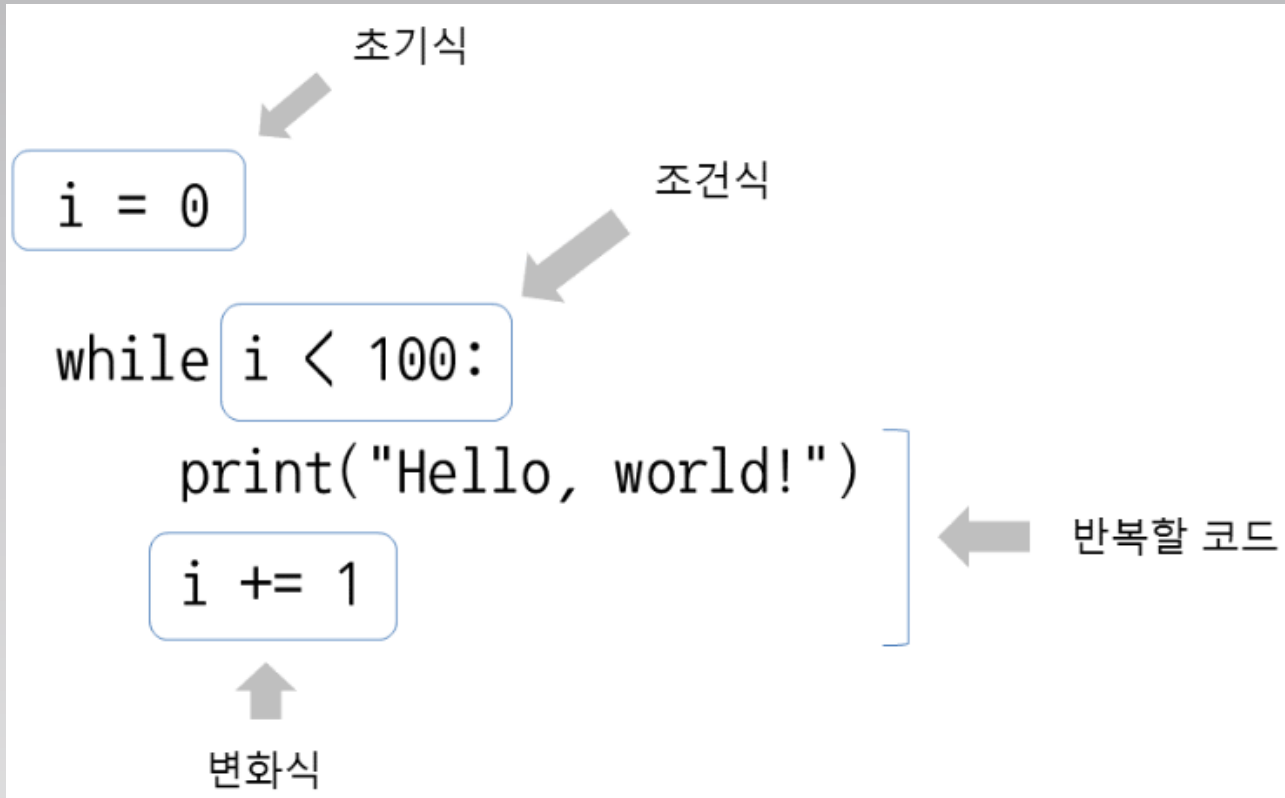
변화식

- while 다음 줄에 오는 코드는 반드시 들여쓰기를 해줌
- while 반복문으로 'Hello, world!'를 100번 출력해보자

```
>>> i = 0
>>> while i < 100:
...     print('Hello, world!')
...     i += 1
...
Hello, world!
... (생략)
Hello, world!
Hello, world!
Hello, world!
```

17.1 while 반복문 사용하기

▼ 그림 while 반복문의 기본 형태



17.1 while 반복문 사용하기

» 초깃값을 1부터 시작하기

- i에 0이 아닌 1을 할당하여 'Hello, world!'를 100번 출력해보자

```
>>> i = 1
>>> while i <= 100:
...     print('Hello, world!', i)
...     i += 1
...
Hello, world! 1
Hello, world! 2
Hello, world! 3
... (생략)
Hello, world! 99
Hello, world! 100
```

17.1 while 반복문 사용하기

» 초깃값을 감소시키기

- 반대로 초깃값을 크게 주고, 변수를 감소시키면서 반복할 수도 있음
- 100부터 1까지 100번 반복함

```
>>> i = 100
>>> while i > 0:
...     print('Hello, world!', i)
...     i -= 1
...
Hello, world! 100
Hello, world! 99
Hello, world! 98
... (생략)
Hello, world! 2
Hello, world! 1
```

17.1 while 반복문 사용하기

» 입력한 횟수대로 반복하기

- 입력한 횟수대로 반복을 해보자
- IDLE의 소스 코드 편집 창에 입력하자

while_input_increment.py

```
count = int(input('반복할 횟수를 입력하세요: '))

i = 0
while i < count:    # i가 count보다 작을 때 반복
    print('Hello, world!', i)
    i += 1
```

- 소스 코드를 실행하면 '반복할 횟수를 입력하세요: '가 출력됨
- 3을 입력하고 엔터 키를 누르자

실행 결과

```
반복할 횟수를 입력하세요: 3 (입력)
Hello, world! 0
Hello, world! 1
Hello, world! 2
```


17.1 while 반복문 사용하기

» 입력한 횟수대로 반복하기

while_input_decrement.py

```
count = int(input('반복할 횟수를 입력하세요: '))

while count > 0:    # count가 0보다 클 때 반복
    print('Hello, world!', count)
    count -= 1      # count를 1씩 감소시킴
```

실행 결과

```
반복할 횟수를 입력하세요: 3 (입력)
Hello, world! 3
Hello, world! 2
Hello, world! 1
```

17.2 반복 횟수가 정해지지 않은 경우

» 반복 횟수가 정해지지 않은 경우

- 난수를 생성해서 숫자에 따라 반복을 끝내 보자
- 난수(random number)란 특정 주기로 반복되지 않으며 규칙 없이 무작위로 나열되는 숫자를 뜻함
- 현실에서 쉽게 접할 수 있는 난수가 바로 주사위를 굴려서 나온 숫자임

▼ 그림 주사위



17.2 반복 횟수가 정해지지 않은 경우

» 반복 횟수가 정해지지 않은 경우

- 파이썬에서 난수를 생성하려면 random 모듈이 필요함
- import 키워드를 사용하여 가져올 수 있음

• import 모듈

```
import random    # random 모듈을 가져옴
```

- random.random()으로 random 모듈의 random 함수를 호출해보자

```
>>> random.random()
0.002383731799935007
>>> random.random()
0.3297914484498006
>>> random.random()
0.6923390064955324
```

- random.random()을 실행할 때마다 계속 다른 실수가 출력되는것은 바로 이 숫자가 바로 난수임

17.2 반복 횟수가 정해지지 않은 경우

» 반복 횟수가 정해지지 않은 경우

- randint 함수는 난수를 생성할 범위를 지정하며, 범위에 지정한 숫자도 난수에 포함됨

- `random.randint(a, b)`

- randint 함수로 주사위를 만들어보자
- 정육면체 주사위는 1부터 6까지 숫자는 `random.randint(1, 6)`처럼 1과 6을 넣으면 1과 6 사이의 난수가 생성됨

```
>>> random.randint(1, 6)
4
>>> random.randint(1, 6)
1
>>> random.randint(1, 6)
5
```

17.2 반복 횟수가 정해지지 않은 경우

» 반복 횟수가 정해지지 않은 경우

- 1과 6 사이의 난수를 생성한 뒤 3이 나오면 반복을 끝냄

while_random.py

```
import random    # random 모듈을 가져옴

i = 0
while i != 3:    # 3이 아닐 때 계속 반복
    i = random.randint(1, 6)    # randint를 사용하여 1과 6 사이의 난수를 생성
    print(i)
```

실행 결과

```
5
1
4
1
1
3
```

17.3 while 반복문으로 무한 루프 만들기

» while 반복문으로 무한 루프 만들기

- while 반복문으로 무한 루프를 만들어보자

while_infinite_loop.py

```
while True:    # while에 True를 지정하면 무한 루프
    print('Hello, world!')
```

실행 결과

```
... (생략)
Hello, world!
Hello, world!
Hello, world!
Hello, world!
... (계속 반복)
```

17.3 while 반복문으로 무한 루프 만들기

» while 반복문으로 무한 루프 만들기

- while에 조건식 대신 True를 지정하면 무한히 반복하는 무한 루프가 만들어짐
- 조건식이 항상 참(True)이므로 변화식도 필요 없음
- 스크립트 파일을 실행한 상태로 두면 'Hello, world!'는 끝나지 않고 계속 출력됨
- IDLE이나 콘솔(터미널, 명령 프롬프트)에서 Ctrl+C를 입력하여 무한 루프를 끝냄
- while에 True 대신 True로 취급하는 값을 사용해도 무한 루프로 동작

```
while 1:    # 0이 아닌 숫자는 True로 취급하여 무한 루프로 동작
    print('Hello, world!')
```

```
while 'Hello':    # 내용이 있는 문자열은 True로 취급하여 무한 루프로 동작
    print('Hello, world!')
```

- while 반복문은 조건식이 참(True)일 때 반복하고 거짓(False)일 때 반복을 끝냄
- while 반복문은 반복 횟수가 정해져 있지 않을 때 자주 사용하므로 이 부분을 꼭 기억해두자
- for 반복문은 반복 횟수가 정해져 있을 때 자주 사용함
- for와 while의 차이점을 정확히 익혀 두고 적절한 곳에 사용하는 것이 좋음