

UNIT 16

for 반복문으로 Hello,
world! 100번 출력하기

16 for반복문으로 Hello,world! 100번출력하기

» for 반복문으로 Hello, world! 100번 출력하기

- 'Hello, world!' 문자열을 100번 출력하려면 어떻게 해야 할까?
- 가장 간단한 방법은 print를 100번 사용해서 출력하는 것임

```
# print 100번 사용
print('Hello, world!')
print('Hello, world!')
print('Hello, world!')
print('Hello, world!')
# ... (생략)
print('Hello, world!')
print('Hello, world!')
print('Hello, world!')
print('Hello, world!')
```

16.1 for와 range 사용하기

» for와 range 사용하기

- 복사, 붙여 넣기로 `print('Hello, world!')`를 100번 붙여 넣으면 어렵지 않게 완성할 수 있음
- 1,000번 또는 10,000번을 출력한다면 어떻게 될까?
- 코드를 붙여 넣는데 시간이 너무 오래 걸리기도 하고, 프로그래밍 측면에서도 비효율적임
- 대부분의 프로그래밍 언어에서는 반복되는 작업을 간단하게 처리하기 위해 반복문이라는 기능을 제공해줌
- 반복문은 반복 횟수, 반복 및 정지 조건을 자유자재로 제어할 수 있음

16.1 for와 range 사용하기

» for와 range 사용하기

- 파이썬의 for 반복문은 다양한 사용 방법이 있지만, 먼저 range와 함께 사용하는 방법부터 알아보자
- for 반복문은 range에 반복할 횟수를 지정하고 앞에 in과 변수를 입력
- 끝에 :(콜론)을 붙인 뒤 다음 줄에 반복할 코드를 넣음

```
for 변수 in range(횟수):  
    반복할 코드
```

16.1 for와 range 사용하기

» for와 range 사용하기

- for 반복문으로 'Hello, world!'를 100번 출력해볼까?

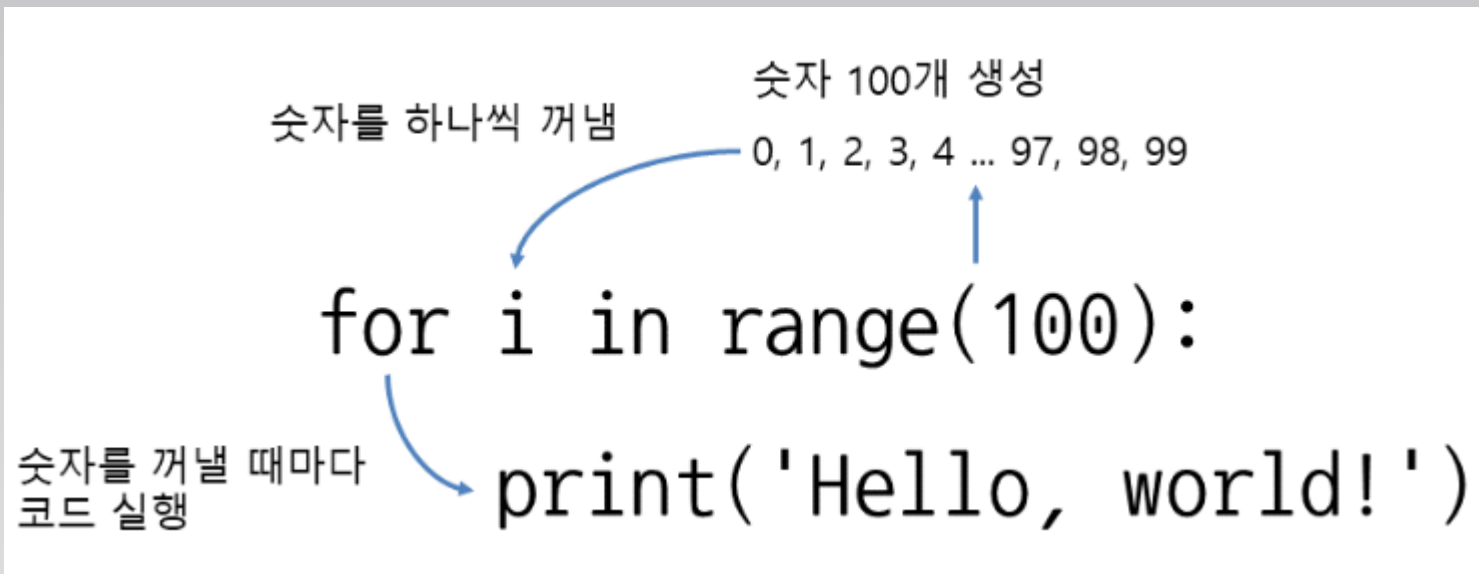
```
>>> for i in range(100):  
...     print('Hello, world!')  
...  
Hello, world!  
... (생략)  
Hello, world!  
Hello, world!  
Hello, world!
```

16.1 for와 range 사용하기

» for와 range 사용하기

- 파이썬의 for 반복문은 range에서 in으로 숫자를 하나하나 꺼내서 반복하는 방식임
- for는 숫자를 꺼낼 때마다 코드를 실행함

▼ 그림 for와 range

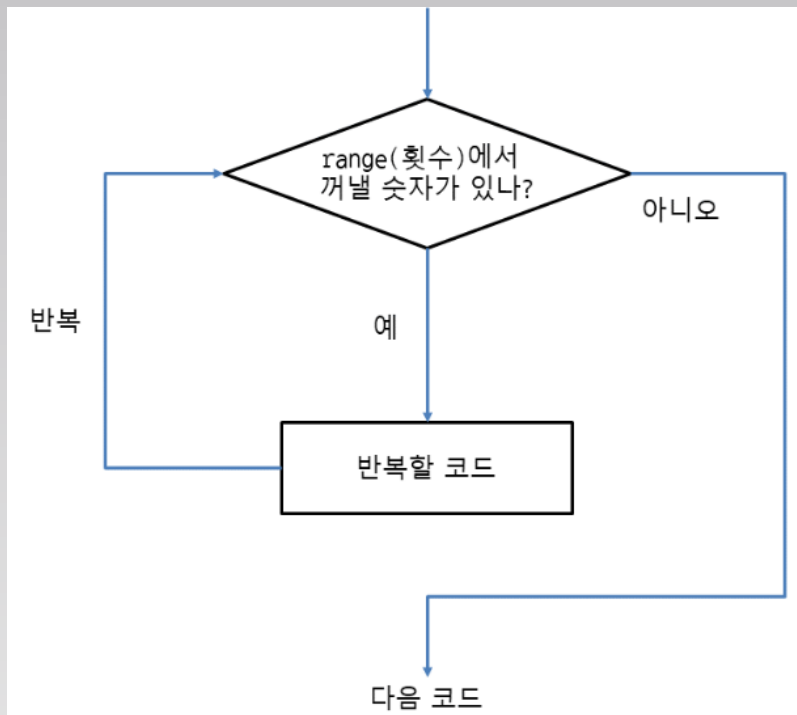


16.1 for와 range 사용하기

» for와 range 사용하기

- for 반복문의 동작 과정 그림으로 표현하면 다음과 같음
- for 변수 in range(횟수) → 반복할 코드로 순환하는 것을 루프(loop)라고 부름

▼ 그림 for와 range의 동작 과정



16.1 for와 range 사용하기

» 반복문에서 변수의 변화 알아보기

- range에서 꺼낸 숫자를 눈으로 확인해보자

```
>>> for i in range(100):  
...     print('Hello, world!', i)  
...  
Hello, world! 0  
Hello, world! 1  
Hello, world! 2  
... (생략)  
Hello, world! 98  
Hello, world! 99
```

- range에서 꺼낸 숫자는 변수 i에 저장되며 반복할 코드에서 사용할 수 있음

16.2 for와 range 응용하기

» for와 range 응용하기

- range의 다양한 기능을 활용하여 for 반복문을 사용해보자

» 시작하는 숫자와 끝나는 숫자 지정하기

- range에 횟수만 지정하면 숫자가 0부터 시작하지만, 다음과 같이 시작하는 숫자와 끝나는 숫자를 지정해서 반복할 수도 있음

• for 변수 in range(시작, 끝):

```
>>> for i in range(5, 12):    # 5부터 11까지 반복
...     print('Hello, world!', i)
...
Hello, world! 5
Hello, world! 6
Hello, world! 7
Hello, world! 8
Hello, world! 9
Hello, world! 10
Hello, world! 11
```

16.2 for와 range 응용하기

» 증가폭 사용하기

- range는 증가폭을 지정해서 해당 값만큼 숫자를 증가시킬 수 있음
- 0부터 9까지의 숫자 중에서 짝수만 출력해보겠음

• for 변수 in range(시작, 끝, 증가폭):

```
>>> for i in range(0, 10, 2):    # 0부터 8까지 2씩 증가
...     print('Hello, world!', i)
...
Hello, world! 0
Hello, world! 2
Hello, world! 4
Hello, world! 6
Hello, world! 8
```

16.2 for와 range 응용하기

» 숫자를 감소시키기

- for와 range는 숫자가 증가하면서 반복함
- 그럼 숫자를 감소시킬 수는 없을까?

```
>>> for i in range(10, 0):    # range(10, 0)은 동작하지 않음
...     print('Hello, world!', i)
... 
```

- 실행을 해보면 아무것도 출력되지 않음
- range는 숫자가 증가하는 기본 값이 양수 1이기 때문임

16.2 for와 range 응용하기

» 숫자를 감소시키기

- '10.1 리스트 만들기'에서 range에 증가폭을 음수로 지정하면 숫자가 감소한다고 함
- 증가폭을 음수로 지정해서 반복해보자

```
>>> for i in range(10, 0, -1):    # 10에서 1까지 1씩 감소
...     print('Hello, world!', i)
...
Hello, world! 10
Hello, world! 9
Hello, world! 8
... (생략)
Hello, world! 2
Hello, world! 1
```

16.2 for와 range 응용하기

» 숫자를 감소시키기

- 증가폭을 음수로 지정하는 방법 말고도 reversed를 사용하면 숫자의 순서를 반대로 뒤집을 수 있음

- for 변수 in reversed(range(횟수))
- for 변수 in reversed(range(시작, 끝))
- for 변수 in reversed(range(시작, 끝, 증가폭))

```
>>> for i in reversed(range(10)):    # range에 reversed를 사용하여 숫자의 순서를 반대로 뒤집음
...     print('Hello, world!', i)    # 9부터 0까지 10번 반복
...
Hello, world! 9
Hello, world! 8
Hello, world! 7
... (생략)
Hello, world! 1
Hello, world! 0
```

16.2 for와 range 응용하기

» 입력한 횟수대로 반복하기

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

for_range_input.py

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

for i in range(count):
    print('Hello, world!', i)
```

- 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
```

16.3 시퀀스 객체로 반복하기

» 시퀀스 객체로 반복하기

- for에 range를 사용하면서 눈치챘겠지만, for에 range 대신 시퀀스 객체를 넣어도 됨
- for는 리스트, 튜플, 문자열 등 시퀀스 객체로 반복할 수 있음
- for에 range 대신 리스트를 넣으면 리스트의 요소를 꺼내면서 반복함

```
>>> a = [10, 20, 30, 40, 50]
>>> for i in a:
...     print(i)
...
10
20
30
40
50
```

16.3 시퀀스 객체로 반복하기

» 시퀀스 객체로 반복하기

- 튜플도 마찬가지로 튜플의 요소를 꺼내면서 반복함

```
>>> fruits = ('apple', 'orange', 'grape')
>>> for fruit in fruits:
...     print(fruit)
...
apple
orange
grape
```

- 참고로 여기서는 for 반복문의 변수를 i 대신 fruit로 사용함
- for에서 변수 i는 다른 이름으로 만들어도 상관없음
- 문자열도 시퀀스 객체임
- for에 문자열을 지정하면 문자를 하나씩 꺼내면서 반복함

```
>>> for letter in 'Python':
...     print(letter, end=' ')
...
P y t h o n
```


16.3 시퀀스 객체로 반복하기

» 시퀀스 객체로 반복하기

- 문자열 'Python'을 뒤집어서 문자를 출력할 수는 없을까?
- 이때는 앞에서 배운 `reversed`를 활용하면 됨

• `reversed(시퀀스객체)`

```
>>> for letter in reversed('Python'):
...     print(letter, end=' ')
...
n o h t y P
```

- 문자열 'Python'에서 문자 n부터 P까지 출력됨
- `reversed`는 시퀀스 객체를 넣으면 시퀀스 객체를 뒤집어 줌(원본 객체 자체는 바뀌지 않으며 뒤집어서 꺼내줌)
- for 반복문은 반복 개수가 정해져 있을 때 주로 사용함
- for 반복문은 range 이외에도 시퀀스 객체를 사용할 수 있다는 점이 중요함