UNIT 10

리스트와 튜플 사용하기

10 리스트와 튜플 사용하기

리스트와 튜플 사용하기

- 지금까지 변수에는 값을 한 개씩만 저장함

```
a = 10
b = 20
```

- 값을 30개 저장하려면 어떻게 해야 할까? 다음과 같이 변수 30개에 값 30개를 저장하면 됨

```
a1 = 10
a2 = 20
# ... (생략)
a29 = 60
a30 = 40
```

- 변수 30개를 일일이 타이핑하기는 쉽지 않으니 이때는 리스트를 사용하면 편리함
- 리스트는 말 그대로 목록이라는 뜻이며 값을 일렬로 늘어놓은 형태임
  (보통 리스트의 값은 코드로 생성하는 경우가 많아서 타이핑할 일이 거의 없음)
- 리스트를 만드는 방법과 기본 사용 방법을 알아보자

10.1 리스트 만들기

리스트 만들기

- 변수에 값을 저장할 때 [ ](대괄호)로 묶어주면 리스트가 되며 각 값은 ,(콤마)로 구분해줌

  - 리스트 = [값, 값, 값]

- 그럼 숫자가 5개 들어있는 리스트를 만들어보자

```
>>> a = [38, 21, 53, 62, 19]
>>> a
[38, 21, 53, 62, 19]
```

- a = [38, 21, 53, 62, 19]와 같이 변수에 [ ]로 값을 저장하여 리스트를 만들어보자
- 리스트에 저장된 각 값 요소(element)라고 부름

10.1 리스트 만들기

리스트에 여러 가지 자료형 저장하기

- 리스트는 문자열, 정수, 실수, 불 등 모든 자료형을 저장할 수 있으며 자료형을 섞어서 저장해도 됨

```
>>> person = ['james', 17, 175.3, True]
>>> person
['james', 17, 175.3, True]
```

- 리스트에 여러 가지 자료형을 사용하면 관련 정보를 하나로 묶기 좋음

10.1 리스트 만들기

빈 리스트 만들기

- 빈 리스트를 만들 때는 [ ]만 지정하거나 list를 사용하면 됨

```
• 리스트 = []
• 리스트 = list()
```

```
>>> a = []
>>> a
[]
>>> b = list()
>>> b
[]
```

- 빈 리스트는 쓸모가 없을 것 같지만, 보통 빈 리스트를 만들어 놓은 뒤에 새 값을 추가하는 방식으로 사용함

10.1 리스트 만들기

range를 사용하여 리스트 만들기

- range를 사용하여 리스트를 만들어보자
- range는 연속된 숫자를 생성하는데 range에 10을 지정하면 0부터 9까지 숫자를 생성함
- 지정한 횟수 숫자는 생성되는 숫자에 포함되지 않음

• range(횟수)

```
>>> range(10)
range(0, 10)
```

- range(0, 10)이라고 나와서 10까지 생성될 것 같지만 10은 포함되지 않음
- list에 range(10)을 넣어보면 0부터 9까지 들어있는 리스트가 생성됨

• 리스트 = list(range(횟수))

```
>>> a = list(range(10))
>>> a
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

10.1 리스트 만들기

range를 사용하여 리스트 만들기

- range는 시작하는 숫자와 끝나는 숫자를 지정할 수도 있음
- 이때도 끝나는 숫자는 생성되는 숫자에 포함되지 않음
- list에 range(5, 12)를 넣으면 5부터 11까지 들어있는 리스트가 생성됨

• 리스트 = list(range(시작, 끝))

```
>>> b = list(range(5, 12))
>>> b
[5, 6, 7, 8, 9, 10, 11]
```

- 증가폭을 사용하는 방법
- range에 증가폭을 지정하면 해당 값만큼 증가하면서 숫자를 생성

• 리스트 = list(range(시작, 끝, 증가폭))

```
>>> c = list(range(-4, 10, 2))
>>> c
[-4, -2, 0, 2, 4, 6, 8]
```

10.1 리스트 만들기

range를 사용하여 리스트 만들기

- range(-4, 10, 2)는 -4부터 8까지 2씩 증가함
- 여기서 끝나는 값은 10이므로 10까지 증가하지 않고 8까지 생성
- 증가폭을 음수로 지정하면 해당 값만큼 숫자가 감소함

```
>>> d = list(range(10, 0, -1))
>>> d
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

- range(10, 0, -1)은 10부터 1씩 감소하며 0은 포함되지 않으므로 1까지 생성됨

10.2 튜플 사용하기

튜플 사용하기

- 파이썬에서는 튜플이라는 자료형도 제공함
- 튜플은 리스트처럼 요소를 일렬로 저장하지만, 안에 저장된 요소를 변경, 추가, 삭제를 할 수 없음
- 간단하게 읽기 전용 리스트라고 할 수 있음
- 변수에 값을 저장할 때 ( )(괄호)로 묶어주면 튜플이 되며 각 값은 ,(콤마)로 구분해줌
- 묶지 않고 값만 콤마로 구분해도 튜플이 됨
  - 튜플 = (값, 값, 값)
  - 튜플 = 값, 값, 값
- 숫자가 5개 들어있는 튜플을 만들어보자

```
>>> a = (38, 21, 53, 62, 19)
>>> a
(38, 21, 53, 62, 19)
```

10.2 튜플 사용하기

튜플 사용하기

- a = (38, 21, 53, 62, 19)와 같이 값을 괄호로 묶은 뒤 변수에 저장하여 튜플을 만들었음
- a = 38, 21, 53, 62, 19와 같이 괄호를 사용하지 않아도 튜플을 만들 수 있음

```
>>> a = 38, 21, 53, 62, 19
>>> a
(38, 21, 53, 62, 19)
```

- 튜플도 리스트처럼 여러 자료형을 섞어서 저장해도 됨
- 저장된 요소를 변경, 추가, 삭제할 수도 없는 튜플을 만든 이유는 파이썬 프로그래밍에서 튜플을 사용하는 쪽이 더 유리한 경우도 있기 때문임
- 튜플은 요소가 절대 변경되지 않고 유지되어야 할 때 사용함
- 튜플을 만든 상태에서 요소를 변경하게 되면 에러가 발생하게 됨
- 요소를 실수로 변경하는 상황을 방지할 수 있음
- 요소를 자주 변경해야 할 때는 리스트를 사용함
- 실무에서는 요소를 변경하는 경우가 많기 때문에 튜플보다 리스트를 더 자주 사용하는 편임

10.2 튜플 사용하기

요소가 한 개 들어있는 튜플 만들기

- 요소가 한 개 들어있는 튜플은 값 한 개를 괄호로 묶으면 튜플이 아니라 그냥 값이 됨

```
>>> (38)
38
```

- 요소가 한 개인 튜플을 만들 때는 ( )(괄호) 안에 값 한 개를 넣고 ,(콤마)를 붙임
- 괄호로 묶지 않고 값 한 개에 ,를 붙여도 됨

```
• 튜플 = (값, )
• 튜플 = 값,
```

```
>>> (38, )
(38,)
>>> 38,
(38,)
```

- 튜플은 요소를 변경, 추가, 삭제할 수도 없는데 값 한 개짜리 튜플은 함수(클래스)를 사용하다 보면 값이 아닌 튜플을 넣어야 할 경우가 생김
- 이때 값은 한 개지만 튜플을 넣어야 할 때 (값,)과 같은 형식을 사용해야 함
- 실무에서는 가끔 이 문법을 사용하게 되는데, 그냥 튜플 형태를 유지하기 위한 문법이라고 생각하면 됨

10.2 튜플 사용하기

range를 사용하여 튜플 만들기

- range를 사용하여 튜플을 만드는 방법
- tuple 안에 range를 넣으면 튜플이 생성됨

• 튜플 = tuple(range(횟수))

```
>>> a = tuple(range(10))
>>> a
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
```

- 다음은 5부터 11까지 들어있는 튜플을 만듬

• 튜플 = tuple(range(시작, 끝))

```
>>> b = tuple(range(5, 12))
>>> b
(5, 6, 7, 8, 9, 10, 11)
```

10.2 튜플 사용하기

range를 사용하여 튜플 만들기

- range에 증가폭을 지정하는 방법도 가능함

```
• 튜플 = tuple(range(시작, 끝, 증가폭))
```

```
>>> c = tuple(range(-4, 10, 2))
>>> c
(-4, -2, 0, 2, 4, 6, 8)
```

- range(-4, 10, 2)는 -4부터 2씩 증가하며 10은 포함되지 않으므로 8까지 들어있는 튜플을 만듬

10.2 튜플 사용하기

튜플을 리스트로 만들고 리스트를 튜플로 만들기

- 튜플과 리스트는 요소를 변경, 추가, 삭제할 수 있는지 없는지만 다를 뿐 기능과 형태는 같음
- 튜플을 리스트로 만들거나 리스트를 튜플로 만들 수도 있음
- tuple 안에 리스트를 넣으면 새 튜플이 생김

```
>>> a = [1, 2, 3]
>>> tuple(a)
(1, 2, 3)
```

- 반대로 list 안에 튜플을 넣으면 새 리스트가 생성됨

```
>>> b = (4, 5, 6)
>>> list(b)
[4, 5, 6]
```

- 리스트를 생성할 때는 [ ](대괄호)를 사용하고, 튜플을 생성할 때는 ( )(괄호)를 사용한다는 점이 중요함
- 튜플은 안에 저장된 요소를 변경, 추가, 삭제할 수 없으므로 요소가 그대로 유지되어야 할 때 사용한다는 점도 기억해두자