



## 06장 튜플과 딕셔너리

# 튜플 자료형 1/2



항목	설명	예시
튜플 기본 사용법	여러 값을 하나의 변수에 저장하는 불변(immutable) 자료형입니다.	<code>tuple()</code>
튜플 요소값 삭제 오류	튜플은 불변 자료형이므로 요소를 삭제하려 하면 오류가 발생합니다.	<pre>&gt;&gt;&gt; t1 = (1, 2, 'a', 'b') &gt;&gt;&gt; del t1[0]</pre>
튜플 요소값 변경 오류	튜플의 요소는 한 번 정해지면 변경할 수 없습니다.	<pre>&gt;&gt;&gt; t1 = (1, 2, 'a', 'b') &gt;&gt;&gt; t1[0] = 'c'</pre>
튜플 인덱싱	리스트와 동일하게 인덱스를 사용하여 요소에 접근할 수 있습니다.	<pre>&gt;&gt;&gt; t1 = (1, 2, 'a', 'b') &gt;&gt;&gt; t1[0] 1 &gt;&gt;&gt; t1[3] 'b'</pre>

# 튜플 자료형 2/2



항목	설명	예시
튜플 슬라이싱	리스트처럼 슬라이싱을 이용하여 일부 요소를 가져올 수 있습니다.	<pre>&gt;&gt;&gt; t1 = (1, 2, 'a', 'b') &gt;&gt;&gt; t1[1:] (2, 'a', 'b')</pre>
튜플 더하기 (+)	두 개의 튜플을 합쳐 새로운 튜플을 생성할 수 있습니다.	<pre>&gt;&gt;&gt; t2 = (3, 4) &gt;&gt;&gt; t1 + t2 (1, 2, 'a', 'b', 3, 4)</pre>
튜플 곱하기 (*)	튜플을 특정 횟수만큼 반복하여 새로운 튜플을 생성할 수 있습니다.	<pre>&gt;&gt;&gt; t2 * 3 (3, 4, 3, 4, 3, 4)</pre>

# 형성평가 1/4



## 문제 1

Python에서 튜플을 올바르게 선언한 것은?

1. {1, 2, 3}
2. [1, 2, 3]
3. (1, 2, 3)
4. <1, 2, 3>

## 문제 2

아래 코드 실행 시 결과는?

```
my_tuple = (10, 20, 30)
my_tuple[1] = 50
```

1. (10, 50, 30)
2. (10, 20, 30, 50)
3. 오류 발생
4. [10, 50, 30]

# 형성평가 2/4



## 문제 3

아래 코드 실행 결과는?

```
my_tuple = (5, 10, 15, 20)  
print(my_tuple[2])
```

1. 5
2. 10
3. 15
4. 20

## 문제 4

아래 코드 실행 결과는?

```
my_tuple = (1, 2, 3, 4, 5)  
print(my_tuple[1:4])
```

1. (1, 2, 3)
2. (2, 3, 4)
3. (3, 4, 5)
4. (2, 3, 4, 5)

# 형성평가 3/4



## 문제 5

튜플을 연결(더하기)하려면 어떤 연산자를 사용해야 하나요?

1. +
2. -
3. \*
4. /

## 문제 6

아래 코드 실행 결과는?

```
tuple1 = (1, 2, 3)
tuple2 = (4, 5)
print(tuple1 + tuple2)
```

1. (1, 2, 3, 4, 5)
2. (1, 2, 3)(4, 5)
3. [1, 2, 3, 4, 5]
4. 1, 2, 3, 4, 5

# 형성평가 4/4



## 문제 7

튜플을 여러 번 반복하여 새로운 튜플을 생성하려면 어떤 연산자를 사용해야 하나요?

1. +
2. -
3. \*
4. /

## 문제 8

아래 코드 실행 결과는?

```
tuple1 = (1, 2, 3)
print(tuple1 * 2)
```

1. (1, 2, 3, 1, 2, 3)
2. (1, 2, 3, 2, 3, 1)
3. (1, 2, 3, (1, 2, 3))
4. [1, 2, 3, 1, 2, 3]

# 딕셔너리 자료형 1/2



항목	설명	예시
딕셔너리 기본 사용법	{Key: Value} 형식으로 데이터를 저장하는 자료형	<pre>dict1 = {'name': 'Alice', 'age': 25}</pre>
딕셔너리 쌍 추가하기	새로운 Key-Value 쌍을 추가할 수 있음	<pre>&gt;&gt;&gt; a = {'1': 'a'} &gt;&gt;&gt; a[2] = 'b' &gt;&gt;&gt; a {'1': 'a', 2: 'b'}</pre>
딕셔너리 요소 삭제하기	del 키워드를 사용하여 특정 Key-Value 삭제 가능	<pre>&gt;&gt;&gt; del a[1] &gt;&gt;&gt; a {2: 'b'}</pre>
딕셔너리에서 Key 사용해 Value 얻기	Key를 이용하여 해당 Value를 가져올 수 있음	<pre>&gt;&gt;&gt; grade = {'pey': 10, 'julliet': 99} &gt;&gt;&gt; grade['pey'] 10 &gt;&gt;&gt; grade['julliet'] 99</pre>
딕셔너리 만들 때 주의할 사항	Key가 중복되었을때 1개를 제외한 나머지는 무시	<pre>&gt;&gt;&gt; a = {'1': 'a', 1: 'b'} &gt;&gt;&gt; a {1: 'b'}</pre>
Key 리스트 만들기 (keys())	모든 Key 값을 리스트 형태로 가져올 수 있음	<pre>&gt;&gt;&gt; a = {'name': 'pey', 'phone': '01199933 23', 'birth': '1118'} &gt;&gt;&gt; a.keys() dict_keys(['name', 'phone', 'birth'])</pre>



# 딕셔너리 자료형 2/2



항목	설명	예시
Value 리스트 만들기 (values())	모든 Value 값을 리스트 형태로 가져올 수 있음	<pre>&gt;&gt;&gt; a.values() dict_values(['pey', '0119993323', '1118'])</pre>
Key, Value 쌍 얻기 (items())	모든 Key-Value 쌍을 튜플 형태로 가져올 수 있음	<pre>&gt;&gt;&gt; a = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'} &gt;&gt;&gt; a.items() dict_items([('name', 'pey'), ('phone', '0119993323'), ('birth', '1118')])</pre>
Key: Value 쌍 모두 지우기 (clear())	딕셔너리의 모든 요소를 삭제	<pre>&gt;&gt;&gt; a.clear() &gt;&gt;&gt; a {} &gt;&gt;&gt; a = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'} &gt;&gt;&gt; a.get('name') 'pey'</pre>
Key로 Value 얻기 (get())	Key를 이용해 Value를 가져오며, Key가 없을 경우 None 반환	<pre>&gt;&gt;&gt; a = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'} &gt;&gt;&gt; 'name' in a True &gt;&gt;&gt; 'email' in a False</pre>
해당 Key가 딕셔너리에 있는지 조사 (in)	특정 Key가 딕셔너리에 존재하는지 확인	

# 형성평가 1/4



## 문제 1

다음 중 딕셔너리를 만드는 올바른 방법은?

1. `dict1 = {1: 'apple', 'name': 'Alice'}`
2. `dict1 = {[1, 2]: 'value'}`
3. `dict1 = (1: 'apple', 2: 'banana')`
4. `dict1 = [1: 'apple', 2: 'banana']`

## 문제 2

딕셔너리에 새로운 Key-Value 쌍을 추가하는 올바른 방법은?

1. `dict1.add('city', 'Seoul')`
2. `dict1['city'] = 'Seoul'`
3. `dict1.insert('city', 'Seoul')`
4. `dict1.append('city', 'Seoul')`

# 형성평가 2/4



## 문제 3

`del dict1['age']` 코드의 역할은?

1. `dict1` 전체를 삭제한다.
2. `dict1`의 모든 값을 `None`으로 만든다.
3. `dict1`에서 'age' 키와 해당 값을 삭제한다.
4. 새로운 키 'age'를 추가한다.

## 문제 4

`dict1.get('name', 'Not found')`의 의미는?

1. 'name' 키가 없으면 'Not found'를 반환한다.
2. 'name' 키가 있으면 삭제한다.
3. 'name' 키의 값을 변경한다.
4. 'name' 키를 무조건 'Not found'로 바꾼다.

# 형성평가 3/4



## 문제 5

딕셔너리에서 모든 Key를 리스트 형태로 가져오는 방법은?

1. `dict1.keys()`
2. `dict1.get_keys()`
3. `dict1.all_keys()`
4. `dict1.find_keys()`

## 문제 6

딕셔너리의 모든 Key-Value 쌍을 리스트 형태로 가져오려면?

1. `dict1.items()`
2. `dict1.values()`
3. `dict1.list()`
4. `dict1.get_all()`

# 형성평가 4/4



## 문제 7

`dict1.clear()`를 실행하면 어떻게 되나요?

1. `dict1`이 초기화되어 모든 요소가 삭제된다.
2. `dict1`이 삭제되고 사용 불가능해진다.
3. `dict1`의 Key는 남고 Value만 삭제된다.
4. `dict1`의 값을 `None`으로 변경한다.

## 문제 8

특정 Key가 딕셔너리에 존재하는지 확인하는 방법은?

1. `'name' exists in dict1`
2. `'name' in dict1`
3. `dict1.contains('name')`
4. `dict1.has_key('name')`