

chapter 13. 문자열과 정규 표현식

1) 문자열 다루기 - 인덱스 조회

① 변수 **a**에 "문자열을 다루 봅시다."를 저장하고 인덱스를 사용해 두 번째 글자를 출력하시오.

In []:

```
1 a = "문자열을 다루 봅시다."  
2 print(a[1])
```

자

② 변수 **a**에 "문자열을 다루 봅시다."를 저장하고 인덱스를 사용해 세 번째 글자부터 여섯 번째 글자까지 출력하시오.

In []:

```
1 a = "문자열을 다루 봅시다."  
2 print(a[2:7])
```

열을 다루

③ 변수 **a**에 "문자열을 다루 봅시다."를 저장하고 인덱스를 사용해 뒤에서 두 번째 글자를 출력하시오.

In []:

```
1 a = "문자열을 다루 봅시다."  
2 print(a[-2])
```

다

④ 변수 **a**에 "문자열을 다루 봅시다."를 저장하고 인덱스를 사용해 "뤼 뵓"을 출력해 봅시다.

- 인덱스를 앞에서 부터, 뒤에서부터(-) 둘다 시도해보세요.

In [1]:

```
1 a = "문자열을 다루 봅시다."  
2 print(a[6:9])  
3 print(a[-6:-3])
```

뤼 뵓

뤼 뵓

⑤ [심화] 변수 **c**에 "계속해서 연습해 보겠습니다."를 저장하고 인덱스를 사용해 홀수 번째 글자들만 출력하시오.

- 리스트의 값들을 하나씩 불러오기 위해 for loop를 사용해 봅시다.
- 힌트: 문자열의 길이만큼 for loop를 돌려야 합니다. len(문자열) 사용

In []:

```
1 c = "계속해서 연습해 보겠습니다."  
2  
3 for i in range(len(c)) :  
4     if (i + 1) % 2 == 1 :  
5         print(c[i])
```

계
해

습

겠
니

.

2) 문자열 다루기 - in, not in

① 변수 a에 "문자열을 다뤄 봅시다."를 저장하고 문자열에 '문'이라는 문자가 포함되어 있는지 확인하시오.

- 힌트: 문자 in 문자열

In []:

```
1 a = "문자열을 다뤄 봅시다."  
2 print('문' in a)
```

True

② 변수 a에 "문자열을 다뤄 봅시다."를 저장하고 문자열에 '만'이라는 문자가 포함되어 있지 않은지 확인하시오.

- 힌트: 문자 not in 문자열

In []:

```
1 a = '문자열을 다뤄 봅시다.'  
2 print('만' not in a)
```

True

③ 변수 b에 "파이썬 배울수록 재미있죠?"를 저장하고 문자열에 '배'라는 문자가 포함되어 있는지 확인하시오.

In []:

```
1 b = "파이썬 배울수록 재미있죠?"  
2 print('배' in b)
```

True

④ 변수 b에 "파이썬 배울수록 재미있죠?"를 저장하고 문자열에 '가'이라는 문자가 포함되어 있지 않은지 확인하시오.

In []:

```
1 b = "파이썬 배울수록 재미있죠?"
2 print('가' not in b)
```

True

3) 문자열 다루기 - 문자열 안에 문자열 넣기

① 변수 a에 {}를 넣은 문자열 "이름 : {}, 나이 {}"를 저장하고 .format()으로 여러분의 이름과 나이를 넣어 출력하시오.

In []:

```
1 a = "이름 : {}, 나이 {}"
2
3 print(a.format('홍길동', 48))
```

이름 : 홍길동, 나이 48

② 바깥에서 지정한 변수 값을 문자열 안에 넣어서 출력해 봅시다.

- 변수 name에는 이름, age에는 나이 값을 저장하고
- 변수 a에는 f"문자열{변수}..." 형태를 사용해 위와 동일한 결과가 나오도록 출력하시오.

In []:

```
1 name = '홍길동'
2 age = 48
3 a = f"이름 : {name}, 나이 {age} "
4 print(a)
```

이름 : 홍길동, 나이 48

③ 딕셔너리로 부터 키와 값을 출력해 봅시다.

- 딕셔너리 : '아메리카노' : 3500, '카페라떼':3800, '콜드브루':4000, '모카프라프치노':4500
- for loop를 이용하여 딕셔너리의 모든 키와 값에 대해 아래 문장이 출력되도록 합시다.

```
"아메리카노는 3500원 입니다."
"카페라떼는 3800원 입니다."
"콜드브루는 4000원 입니다."
"모카프라프치노는 4500원 입니다."
```

- 힌트 : "{ }", format을 이용하세요

In [3]:

```

1 a = "{}는 {}원입니다."
2 d1 = {'아메리카노' : 3500, '카페라떼' : 3800, '콜드브루' : 4000, '모카프라프치노' : 4500 }
3
4 for k, v in d1.items() :
5     print(a.format(k, v))

```

아메리카노는 3500원입니다.

카페라떼는 3800원입니다.

콜드브루는 4000원입니다.

모카프라프치노는 4500원입니다.

4) 문자열 다루기 - 여러 줄 문자열

① 아래와 같은 결과가 나오도록 변수 a에 문자열을 저장하여 출력하시오.

이름 : 홍길동

나이 : 48

- 힌트: "" ""를 사용해 봅시다.

In []:

```

1 a = '''이름 : 홍길동
2   나이 : 48'''
3 print(a)

```

이름 : 홍길동

나이 : 48

② 아래와 같은 결과가 나오도록 변수 b에 문자열을 저장하여 출력하시오.

[전화번호부]

ㄱㄱㄱ : 010-1234-5678

ㄴㄴㄴ : 010-2346-1578

ㄷㄷㄷ : 010-3458-6712

In []:

```

1 b = '''[전화번호부]
2   ㄱㄱㄱ : 010-1234-5678
3   ㄴㄴㄴ : 010-2346-1578
4   ㄷㄷㄷ : 010-3458-6712
5   '''
6
7 print(b)

```

[전화번호부]

ㄱㄱㄱ : 010-1234-5678

ㄴㄴㄴ : 010-2346-1578

ㄷㄷㄷ : 010-3458-6712

5) 문자열 다루기 - 문자열을 다루는 메서드들

① 변수 a에 "I love you"를 저장하고 모든 문자들을 대문자로 변환하여 출력하시오.

- 힌트: 문자열.upper()

In []:

```

1 a = "I have a dream."
2 print(a.upper())

```

I LOVE YOU

② 변수 b에 "Life is a journey"를 저장하고 모든 문자들을 대문자로 변환하여 출력하시오.

In []:

```

1 b = "Life is a journey"
2 print(b.upper())

```

LIFE IS A JOURNEY

③ 변수 a에 "I Love You"를 저장하고 모든 문자들을 소문자로 변환하여 출력하시오.

- 힌트: 문자열.lower()

In []:

```

1 a = "I Love You"
2 print(a.lower())

```

i love you

④ 변수 b에 "LIFE is a JOURNEY"를 저장하고 모든 문자들을 소문자로 변환하여 출력하시오.

In []:

```
1 b = "LIFE is a JOURNEY"
2 print(b.lower())
```

life is a journey

⑤ 변수 a에 "i love you"를 저장하고 첫 번째 문자만 대문자로 변환하여 출력하시오.

- 힌트: 문자열.capitalize()

In []:

```
1 a = "i love you"
2 print(a.capitalize())
```

I love you

⑥ 리스트 a = ['사람', '코끼리', '낙타']일 때, 각각의 값 사이에 ', '이 들어가도록 a를 하나의 문자열로 결합하시오.

- 힌트: 값 사이 들어갈 문자.join(리스트)

In [4]:

```
1 a = ['사람', '코끼리', '낙타']
2 ', '.join(a)
```

Out[4]:

'사람, 코끼리, 낙타'

⑦ 리스트 b = ['this', 'is', 'me']일 때, 각각의 값 사이에 ' '(한칸 공백) 이 들어가도록 b를 하나의 값으로 결합하시오.

In []:

```
1 b = ['this', 'is', 'me']
2 ' '.join(b)
```

Out[32]:

'this is me'

⑧ 변수 a에 "나는 파이썬을 배우고 있다."를 저장하고 공백을 기준으로 분할하여 리스트 형태로 출력하시오.

- 힌트: 문자열.split(분할의 기준이 될 값)

In []:

```
1 a = "나는 파이썬을 배우고 있다."
2 print(a.split())
```

['나는', '파이썬을', '배우고', '있다.']

- ⑨ 변수 `b`에 '사람, 코끼리, 낙타'를 저장하고 ','를 기준으로 분할하여 리스트 형태로 출력하시오.

In []:

```
1 b = '사람, 코끼리, 낙타'
2 print(b.split(','))
```

['사람', ' 코끼리', ' 낙타']

- ⑩ 변수 `c`에 '010-1234-5678'을 저장하고 '-'를 기준으로 분할하여 리스트 형태로 출력하시오.

In []:

```
1 c = '010-1234-5678'
2 print(c.split('-'))
```

['010', '1234', '5678']

- ⑪ 변수 `a`에 ' 홍길동 '을 저장하고 왼쪽, 오른쪽, 양쪽 공백을 제거하여 각각 출력하시오.

- 힌트: 문자열.lstrip(), 문자열.rstrip(), 문자열.strip()

In []:

```
1 a = ' 홍길동 '
2 print(a.lstrip())
3 print(a.rstrip())
4 print(a.strip())
```

홍길동

홍길동

홍길동

6)정규표현식

- ① [심화]전화번호가 규칙에 맞는지 검사하는 코드를 작성해 봅시다.

- import re
- 아래 코드는 오전 예제 내용입니다. 수정을 합시다.

```
MobileNumberChk = re.compile(r'WdWdWd-WdWdWdWd-WdWdWdWd')
```

- 번호 규칙을 아래와 같이 수정합시다.
 - 010으로 시작
 - 4번째, 8번째 숫자는 1~9 중 하나 : [1-9]

In [6]:

```
1 import re
2
3 # 정규표현식으로 핸드폰 번호 검사기 만들기
4 MobileNumberChk = re.compile(r'010-[1-9]WdWdWd-[1-9]WdWdWd')
```

In [7]:

```
1 # 제대로 들어간 값
2 mo = MobileNumberChk.search('010-1234-5678')
3
4 if mo is not None :
5     print('전화번호가 정상입니다.')
6 else :
7     print('전화번호가 비정상입니다.')
```

전화번호가 정상입니다.

In []:

1