Python 기초

3주차

Python 리스트

리스트 (list)

기본 사용 법

```
변수명 = []
```

```
변수명 = [value1, value2, …]
```

Indexing (인덱싱)

리스트의 특정 값에 접근 하기 위한 방법

```
  0   1   2   3   4   5
[ 1 , 2 , 3 , 4 , 5 , 6 ]
```

Indexing (인덱싱)

사용 예제

```
lst = [1, 2, 3]


print(lst)

print(lst[0])

print(lst[1])
```

Indexing (인덱싱)

음수 값으로 접근

```
  -4    -3    -2    -1
[  1  ,  2  ,  3  ,  4  ]
```

Indexing (인덱싱)

사용 예제

```
lst = [1, 2, 3]


print(lst)

print(lst[-1])

print(lst[-3])
```

Slicing (슬라이싱)

리스트의 특정 값들의 범위에 접근하기 위한 방법

```
0   1   2   3   4   5   6
[ 1 , 2 , 3 , 4 , 5 , 6 ]
```

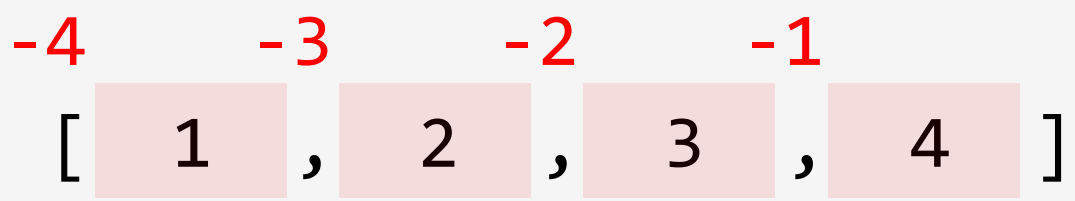
Slicing (슬라이싱)

사용 예제

```
lst = [1, 2, 3]


print(lst)

print(lst[0:2])

print(lst[1:3])
```

Slicing (슬라이싱)

음수 값으로 접근



Slicing (슬라이싱)

사용 예제

```
lst = [1, 2, 3]


print(lst)

print(lst[-3:-1])

print(lst[-2:])
```

리스트 - 반복문

for문을 활용한 리스트 접근

```
lst = [1, 2, 3]

for idx in range(3):

    print(lst[idx])
```

리스트 - 반복문

for문을 활용한 리스트 접근

```
lst = [1, 2, 3]

for idx in range(len(lst)):

    print(lst[idx])
```

리스트 - 반복문

for문을 활용한 리스트 접근

```
lst = [1, 2, 3]

for value in lst:

    print(value)
```

리스트 함수

함수	설명
append(value)	리스트 끝에 값을 추가 한다.
extend(iter)	리스트 끝에 list, tuple, dict의 값을 하나씩 추가 한다.
insert(idx, value)	특정 인덱스 위치에 값을 추가 한다.
pop([idx])	마지막 인덱스의 값을 반환 후 삭제 한다. 인덱스 번호를 지정 할 수도 있다.
remove(value)	특정 값에 해당하는 것을 찾아 삭제 한다.
clear()	모든 값을 삭제하여 빈 리스트만 남긴다.
count(value)	리스트에서 일치하는 값의 수를 반환 한다.
index(value)	리스트에서 일치하는 값의 인덱스 번호를 반환 한다.
reverse()	리스트의 모든 값을 뒤집어 나열 한다.
sort([reverse=False])	리스트의 값을 오름차순(False), 내림차순(True) 정렬 한다.

리스트 – append()

append 예제

```
lst = [1, 2, 3]

lst.append('a')

lst.append([4, 'b'])
```

리스트 – extend()

extend 예제

```
lst = [1, 2, 3]

lst.extend(['a', 'b', 'c'])
```

리스트 – insert()

insert 예제

```
lst = [1, 2, 3]

lst.insert(1, 'b')
```

리스트 – pop()

pop 예제

```
lst = [1, 2, 3]

lst.pop()

lst.pop(0)
```

리스트 – remove()

remove 예제

```
lst = [1, 2, 3]

lst.remove(2)
```

리스트 – clear()

clear 예제

```
lst = [1, 2, 3]

lst.clear()
```

리스트 – count()

count 예제

```
lst = [1, 2, 3, 1]

lst.count(1)
```

리스트 – index()

index 예제

```
lst = [1, 2, 3, 1]

lst.index(2)
```

리스트 – reverse()

reverse 예제

```
lst = [1, 3, 2]

lst.reverse()
```

리스트 – sort()

sort 예제

```
lst = [1, 3, 2]

lst.sort()

lst.sort(reverse=True)
```

문제

앞에서 학습한 내용을 바탕으로 다음 문제를 풀어보세요.

1. numbers = [10, 20, 30, 40, 50, 60, 70]
   위 리스트의 모든 값을 더한 결과를 출력 하시오.

2. 1 ~ 45 까지 임의의 값을 중복 없이 6개 생성하여 출력하는 코드를 작성 하시오.

3. lst_sec = [['홍길동', '남', 36], ['김수양', '여', 32], ['박담소', '남', 28]]
   위의 2차 리스트 자료를 다음과 같은 형식으로 출력 하시오.

   ```
   이름 : 홍길동
   성별 : 남
   나이 : 36
   ```

4. 구구단을 출력하는 코드를 작성하되, 2차 리스트에 결과 값을 저장하고 출력 할 수 있도록 하시오.

Python 튜플

튜플 (tuple)

기본 사용 법

```
변수명 = (value1, value2, …)
```

Indexing (인덱싱)

튜플의 특정 값에 접근 하기 위한 방법

```
  0   1   2   3   4   5
( 1 , 2 , 3 , 4 , 5 , 6 )
```

```
  -4    -3    -2    -1
(  1  ,  2  ,  3  ,  4  )
```

Slicing (슬라이싱)

튜플의 특정 값들의 범위에 접근하기 위한 방법

```
0   1   2   3   4   5   7
( 1 , 2 , 3 , 4 , 5 , 6 )
```

```
-4    -3    -2    -1
 ( 1 , 2 , 3 , 4 )
```

Packing / Unpacking

튜플 생성 및 해제

```
변수명 = value1, value2, …
```

```
변수명1, 변수명2 = (1, 2)
```

튜플 함수

함수	설명
count(value)	튜플에서 일치하는 값의 수를 반환 한다.
index(value)	튜플에서 일치하는 값의 인덱스 번호를 반환 한다.

튜플 – count()

count 예제

```
tup = (1, 2, 3, 1, 2)

tup.count(2)
```

튜플 – index()

index 예제

```
tup = (1, 2, 3, 1, 2)

tup.index(2)
```

문제

앞에서 학습한 내용을 바탕으로 다음 문제를 풀어보세요.

1. numbers = (10, 20, 30, 40, 50, 60, 70)
   위 튜플 자료에서 30과 40을 따로 num1, num2 변수에
   할당 하시오.

2. menu = (('칼국수', 6000), ('비빔밥', 5500), ('돼지국밥',
   7000))
   위 자료의 값을 다음의 양식 처럼 출력 하시오.
   칼국수 - 6,000원
   비빔밥 - 5,500원
   돼지국밥 – 7,000원

Python 사전

사전 (dict)

기본 사용 법

```
변수명 = {}
```

```
변수명 = {key1:value1, key2:value2, …}
```

사전 (dict)

키를 가지고 값에 접근

```
dic[key] = value
```

```
print(dic[key])
```

사전 함수

함수	설명
update(dict)	사전형 자료에 값을 추가 한다.
fromkeys(iter, value)	리스트, 튜플에 존재하는 값을 키로 사전형 자료를 생성하여 반환한다.
get(key[, value])	사전형의 키를 통해 값을 반환 한다.
keys()	사전형의 모든 키를 반환 한다.
values()	사전형의 모든 값을 반환 한다.
items()	사전형의 모든 키-값의 쌍을 튜플로 반환 한다.
pop(key)	사전형의 키를 통해 값을 반환 후 삭제 한다.
popitem()	사전형의 키-값의 쌍을 튜플로 반환 후 삭제 한다.
clear()	사전형의 모든 키-값을 삭제하여 빈 사전형 자료만 남긴다.

사전 (dict) – update()

update 예제

```
dic = {'a':1, 'b':2, 'c':3}

dic.update({'d':4})
```

사전 (dict) – fromkeys()

fromkeys 예제

```
k = ['a', 'b', 'c', 'd']

dic1 = {}.fromkeys(k)

dic2 = {}.fromkeys(k, 1)
```

사전 (dict) – get()

get 예제

```
dic = {'a':1, 'b':2, 'c':3}

dic.get('b')

dic.get('d', 'Not exist key')
```

사전 (dict) – keys()

keys 예제

```
dic = {'a':1, 'b':2, 'c':3}

for key in dic.keys():

    print(key)
```

사전 (dict) – values()

values 예제

```
dic = {'a':1, 'b':2, 'c':3}

for value in dic.values():

    print(value)
```

사전 (dict) – items()

items 예제

```
dic = {'a':1, 'b':2, 'c':3}

for key, value in dic.items():

    print(key, value)
```

사전 (dict) – pop()

pop 예제

```
dic = {'a':1, 'b':2, 'c':3}

dic.pop('b')
```

사전 (dict) – popitem()

popitem 예제

```
dic = {'a':1, 'b':2, 'c':3}

dic.popitem()
```

사전 (dict) – clear()

clear 예제

```
dic = {'a':1, 'b':2, 'c':3}

dic.clear()
```

문제

앞에서 학습한 내용을 바탕으로 다음 문제를 풀어보세요.

1. 이름, 전화번호, 이메일 주소를 키로 사용하는 사전 자료를 생성 하시오.

2. 리스트형 변수에 1번 문제와 같은 사전 자료가 여러 개 생성될 수 있도록 하시오.

3. 2번에서 입력한 자료가 출력 될 수 있도록 하시오.

Python 문자열

string indexing

```
st = 'string indexing'

print(st[0])

print(st[7])
```

string slicing

```
st = 'string slicing'

print(st[:6])

print(st[7:])
```

string 함수

함수	설명
find(str)	문자열에서 특정 문자열을 찾아 해당 문자의 Index 값을 반환 한다.
count(str)	문자열에서 특정 문자열을 찾아 해당 문자열의 수를 반환 한다.
lower()	문자열에서 영문자를 소문자로 변경하여 반환 한다.
upper()	문자열에서 영문자를 대문자로 변경하여 반환 한다.
strip()	문자열의 앞뒤 공백을 제거한다.
lstrip()	문자열의 왼쪽 공백을 제거한다.
rstrip()	문자열의 오른쪽 공백을 제거한다.
replace(old, new)	문자열의 특정 문자열을 변경한다.
split(str)	문자열의 특정 문자열을 기준으로 분리 한다.

string – find()

find 예제

```
st = 'Python String'

print(st.find('String'))
```

string – count()

count 예제

```
st = 'Python String'

print(st.count('t'))
```

string – lower()

lower 예제

```
st = 'Python String'

print(st.lower())
```

string – upper()

upper 예제

```
st = 'Python String'

print(st.upper())
```

string – strip()

strip 예제

```
st = '  Python String  '

print(st.strip())
```

string – lstrip()

lstrip 예제

```
st = '   Python String   '

print(st.lstrip())
```

string – rstrip()

rstrip 예제

```
st = '  Python String  '

print(st.rstrip())
```

string – replace()

replace 예제

```
st = 'Python String'

print(st.replace('String', '문자열'))
```

string – split()

split 예제

```
st = 'Python String'

print(st.split(' '))
```

문제

앞에서 학습한 내용을 바탕으로 다음 문제를 풀어보세요.

1. input() 함수로 이름과 나이 값을 입력 받을 때 한 번에 입력 받아 처리하고 출력 하는 코드를 작성 하시오.

2. input() 함수로 입력 받은 더하기, 빼기, 곱하기, 나누기의 간단한 수식을 처리 할 수 있도록 코드를 작성 하시오.