



2023 N.E.T Python

Today

1. Review

- String
- print formatting

2. Loop

3. List

4. Comprehension

Data type

```
>>> float(2)
2.0
>>> str(2)
'2'
```

```
>>> print("50")
50
# type : string
>>> print(50)
50
# type : integer
```

Print formatting

```
'''
```

1. %s : 문자열 ◀ 많이 사용
2. %d : 정수 ◀ 많이 사용
3. %c : 문자 1개
4. %f : 부동 소수
6. %o : 8진수
7. %x : 16진수
6. %% : '%' 자체를 출력

```
'''
```

```
점수 = 95  
이름 = "이포맷"
```

```
print("김파이의 점수는 %d 점 입니다." % 점수)  
print("%s의 점수는 70 점 입니다." % 이름)  
print("%s의 점수는 %d 점 입니다." % (이름, 점수))  
print("%s의 점수는 %d 점 입니다. %s" % (이름, 점수, "합격!"))
```

```
##### result #####
```

```
'''
```

```
김파이의 점수는 95 점 입니다.  
이포맷의 점수는 70 점 입니다.  
이포맷의 점수는 95 점 입니다.  
이포맷의 점수는 95 점 입니다. 합격!  
'''
```

Print formatting

...

1. %s : 문자열 ◀ 많이 사용
2. %d : 정수 ◀ 많이 사용
3. %c : 문자 1개
4. %f : 부동 소수
6. %o : 8진수
7. %x : 16진수
6. %% : '%' 자체를 출력

...

```
>>> print("%6d" % 1234)
```

```
1234 # 공백 출력
```

```
>>> print("%6d" % 1234567)
```

```
1234567 # 출력 X
```

```
>>> print("%06d" % 123)
```

```
000123
```

```
# 실수의 소수점 자리수 출력
```

```
# "%cn.nf"
```

```
>>> print("%05.2f" % 3.14)
```

```
03.14
```

```
# 소수점도 한 칸을 차지하는 것을 알 수 있다.
```

Print formatting

'''

1. %s : 문자열 ◀ 많이 사용
2. %d : 정수 ◀ 많이 사용
3. %c : 문자 1개
4. %f : 부동 소수
6. %o : 8진수
7. %x : 16진수
6. %% : '%' 자체를 출력

'''

```
print("%10s 왼쪽에 공백10개" % "안녕!")
print("%-10s 오른쪽에 공백10개" % "안녕!")
print("%.3f 소수점 포현" % 3.141592) #소수점 인식
print("%.3s 소수점 포현" % 3.141592) #문자열 인식
print("%10.3f 소수점 포현" % 3.141592)
```

result

'''

```
|      |      | 안녕! 왼쪽에 공백10개
안녕!      |      | 오른쪽에 공백10개
3.142 소수점 포현
3.1 소수점 포현
|      |      | 3.142 소수점 포현
```

'''

Print formatting

```
점수 = 95
이름 = "이포맷"
print("김파이의 점수는 {0} 점 입니다." .format(점수))
print("{0}의 점수는 70 점 입니다." .format("이포맷"))
print("{0}의 점수는 {1} 점 입니다." .format(이름, 점수))
print("{0}의 점수는 {1} 점 입니다.{2}" .format(이름, 점수, "합격!"))
```

```
##### result #####
```

```
...
김파이의 점수는 95 점 입니다.
이포맷의 점수는 70 점 입니다.
이포맷의 점수는 95 점 입니다.
이포맷의 점수는 95 점 입니다.합격!
...
```

Print formatting

```
점수 = 95
이름 = "이포맷"

#{인덱스:전체자리수}
print("김파이의 점수는 {0:10} 점 입니다." .format(점수)) #치환하는 값에 10칸 자리수가 생성
print("김파이의 점수는 {0:5} 점 입니다." .format(점수)) #치환하는 값에 5칸 자리수가 생성

#{인덱스:정렬방향 전체자리수}
print("김파이의 점수는 {0:<5} 점 입니다." .format(점수)) #5칸 자리수 주고 왼쪽 정렬
print("김파이의 점수는 {0:>5} 점 입니다." .format(점수)) #5칸 자리수 주고 오른쪽 정렬

#{인덱스:공백에문자넣기 정렬방향 전체자리수}
print("김파이의 점수는 {0:*>5} 점 입니다." .format(점수)) #5칸 자리수 주고 공백엔 *표시 오른쪽 정렬

##### result #####
...
김파이의 점수는          95 점 입니다.
김파이의 점수는      95 점 입니다.

김파이의 점수는 95      점 입니다.
김파이의 점수는      95 점 입니다.

김파이의 점수는    ***95 점 입니다.
```


Print formatting

```
점수=70
print("{name}의 점수는 {score} 점 입니다." .format(name="철수",score=점수))

##### result #####

# 철수의 점수는 70 점 입니다.
```

F-string

```
# 문자열 맨 앞에 f를 붙이고, 출력할 변수, 값을 중괄호 안에 넣습니다.  
s = 'coffee'  
n = 5  
result1 = f'저는 {s}를 좋아합니다. 하루 {n}잔 마셔요.'  
print(result1)
```

F-string

```
# f-string 왼쪽 정렬
s1 = 'left'
result1 = f'|{s1:<10}|'
print(result1)

# f-string 가운데 정렬
s2 = 'mid'
result2 = f'|{s2:^10}|'
print(result2)

# f-string 오른쪽 정렬
s3 = 'right'
result3 = f'|{s3:>10}|'
print(result3)
```

```
|left      |
|   mid    |
|   right  |
```

String function

```
# len()
# 문자열의 길이 반환
# len(str)

str = "I'm king"
>>> print(len(str))
8
```

```
# count()
# 문자열에서 포함된 특정 문자열의 수를 반환
# str.count(idx)

str = "fuxking test"
scnt = str.count('t')
>>> print(scnt)
2
```

String function

```
# find()
# 문자열 중에서 검색 문자열의 위치 정보 반환
# 값이 없다면 -1 출력
# 반환 함수 x 출력 함수
# string.find(value,start,end)
# 첫 번째 위치를 반환한다.

str = 'headache'
>>> str.find(a)
4
```

```
# split()
# 문자열을 구분문자로 나누어 문자열 리스트로 변환
# string.split(str)
# 여기서는 str이 구분 문자열

ret = "aa bb cc".split(' ')
>>> print(ret)
['aa', 'bb', 'cc']
```

String function

```
# upper()
# 문자열을 대문자로 변환
# string.upper()
>>> print('abcd'.upper())
ABCD

# lower()
# 문자열을 대문자로 변환
# string.lower()
>>> print('ABCD'.lower())
abcd
```

```
# replace()
# 문자열에서 특정 부분 문자열을 다른 문자열로 변환
# string.replace('str1','str2')

str = 'aaaabaaabababababababababbababababba'.replace('b','a')
>>> print(str)
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

Loop

```
while <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    <수행할 문장3>  
    ...
```

```
>>> treeHit = 0  
>>> while treeHit < 10:  
...     treeHit = treeHit + 1  
...     print("나무를 %d번 찍었습니다." % treeHit)  
...     if treeHit == 10:  
...         print("나무 넘어갑니다.")  
...  
나무를 1번 찍었습니다.  
나무를 2번 찍었습니다.  
나무를 3번 찍었습니다.  
나무를 4번 찍었습니다.  
나무를 5번 찍었습니다.  
나무를 6번 찍었습니다.  
나무를 7번 찍었습니다.  
나무를 8번 찍었습니다.  
나무를 9번 찍었습니다.  
나무를 10번 찍었습니다.  
나무 넘어갑니다.
```

Loop

```
>>> coffee = 10
>>> money = 300
>>> while money:
...     print("돈을 받았으니 커피를 줍니다.")
...     coffee = coffee - 1
...     print("남은 커피의 양은 %d개입니다." % coffee)
...     if coffee == 0:
...         print("커피가 다 떨어졌습니다. 판매를 중지합니다.")
...         break
... 
```


Loop

```
for 변수 in 리스트(또는 튜플, 문자열):  
    수행할 문장1  
    수행할 문장2  
    ...
```

```
>>> test_list = ['one', 'two', 'three']  
>>> for i in test_list:  
...     print(i)  
...  
one  
two  
three
```

Loop

```
marks = [90, 25, 67, 45, 80]

number = 0
for mark in marks:
    number = number + 1
    if mark >= 60:
        print("%d번 학생은 합격입니다." % number)
    else:
        print("%d번 학생은 불합격입니다." % number)
```

Loop

```
marks = [90, 25, 67, 45, 80]

number = 0
for mark in marks:
    number = number + 1
    if mark < 60:
        continue
    print("%d번 학생 축하합니다. 합격입니다. " % number)
```

Loop

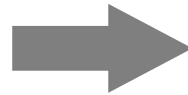
```
>>> a = range(10)
>>> a
range(0, 10)
```

```
>>> add = 0
>>> for i in range(1, 11):
...     add = add + i
...
>>> print(add)
55
```

List

```
list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
tinylist = [123, 'john']

print(list)
print(list[0] )
print(list[1:3])
print(list[2:])
print(tinylist * 2)
print(list + tinylist)
```



```
['abcd', 786, 2.23, 'john', 70.2]
abcd
[786, 2.23]
[2.23, 'john', 70.2]
[123, 'john', 123, 'john']
['abcd', 786, 2.23, 'john', 70.2, 123, 'john']
```

List

```
inputList = list(map(int,input.split()))
```

List

```
>>> a = [1, 2, 3]
>>> a[2] = 4
>>> a
[1, 2, 4]
```

```
>>> a = [1, 2, 3]
>>> del a[1]
>>> a
[1, 3]
```

List

```
>>> a = [1, 2, 3]
>>> a.append(4)
>>> a
[1, 2, 3, 4]

>>> a.append([5,6])
>>> a
[1, 2, 3, 4, [5, 6]]
```

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


Comprehension

1. Using loop

```
[i for i in range(10)]  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
  
[i*3 for i in range(10)]  
[0, 3, 6, 9, 12, 15, 18, 21, 24, 27]
```

Comprehension

2. Using Conditional

```
[i for i in range(10)]  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
  
[i for i in range(10) if i%2 == 0]  
[0, 2, 4, 6, 8]
```

Comprehension

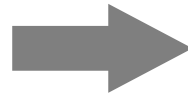
2. Using Conditional

```
[i for i in range(10)]  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
  
['even' if i%2 == 0 else 'odd' i for i in range(10)]  
['even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd']
```

Tuple

```
tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )
tinytuple = (123, 'john')

print(tuple)
print(tuple[0])
print(tuple[1:3])
print(tuple[2:])
print(tinytuple * 2)
print(tuple + tinytuple)
```

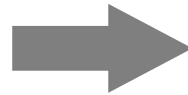


```
('abcd', 786, 2.23, 'john', 70.2)
abcd
(786, 2.23)
(2.23, 'john', 70.2)
(123, 'john', 123, 'john')
('abcd', 786, 2.23, 'john', 70.2, 123, 'john')
```

Dictionary

```
list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
tinylist = [123, 'john']

print(list)
print(list[0] )
print(list[1:3])
print(list[2:])
print(tinylist * 2)
print(list + tinylist)
```



```
['abcd', 786, 2.23, 'john', 70.2]
abcd
[786, 2.23]
[2.23, 'john', 70.2]
[123, 'john', 123, 'john']
['abcd', 786, 2.23, 'john', 70.2, 123, 'john']
```