

[Python] 파이썬 기초 8 - 반복문(for , while)에 대한 정의와 기본적인 함수 사용법출처 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-10-23 오후 10:32

URL: <https://continuous-development.tistory.com/68?category=736681>

Python

[Python] 파이썬 기초 8 - 반복문(for , while)에 대한 정의와 기본적인 함수 사용법출처

2020. 8. 12. 22:35 수정 삭제 공개



파이썬 반복문

반복문에는 for문과 while문이 있다.

for 문과 while문 둘다 조건에 만족 할 때 까지 계속 반복을 한다.

for문 형태

1. for value in <collection> :

2. for idx in range(start, stop, step)

#range 를 이용한 for 문(인수 1개)

0~10 전 까지 for문을 돌린다.

```
for idx in range(10):  
    print("idx -", idx)
```

result

idx - 0

idx - 1

idx - 2

idx - 3

idx - 4

idx - 5

idx - 6

idx - 7

idx - 8

idx - 9

#range 를 이용한 for 문(인수 2개)

2부터 10 전 까지 for문을 돌린다.

```

for idx in range(2,10):
    print("idx -", idx)

## result
# idx - 2
# idx - 3
# idx - 4
# idx - 5
# idx - 6
# idx - 7
# idx - 8
# idx - 9

```

#range 를 이용한 for 문(인수 3개)

0 부터 11 전까지 for 문을 돌리는데 2의 간격으로 돌린다.

```

for idx in range(0, 11, 2):
    print("idx -", idx)

# !result
# idx - 0
# idx - 2
# idx - 4
# idx - 6
# idx - 8
# idx - 10

```

변수로 for 문 돌리는 방법 1 - 이때는 idx에 scores의 값이 들어간다.

```

scores = []

for idx in range(5):
    scores.append(int(input("성적을 입력하세요 : ")))
    print(scores)

# !result
# 성적을 입력하세요 : 100
# 성적을 입력하세요 : 99
# 성적을 입력하세요 : 97
# 성적을 입력하세요 : 98
# 성적을 입력하세요 : 100

for idx in scores:
    print(idx)

```

```
#result
# 100
# 100
# 100
# 100
# 100
```

변수로 for 문 돌리는 방법 2 - 이때는 scores의 길이 만큼 돌고 idx는 0부터 시작 해서 길이만큼 돈다.

```
for idx in range(len(scores)):
    print(scores[idx], "\t", end=" ")
    print()

# 입력받은 성적의 총합과 평균을 구하세요
sum_value1 = 0
sum_value2 = 0

# 첫번째 형식
for element in scores:
    sum_value1 += element # sum_value1 = sum_value1 + element

# 두번째 형식
for idx in range(len(scores)):
    sum_value2 += scores[idx] # sum_value2 = sum_value2 + scores[idx]

print(sum_value1)
print(sum_value1 / len(scores))

print(sum_value2)
print(sum_value2 / len(scores))

# for 문 없이 함수로 푸는 방법
print(sum(scores))
print(sum(scores)/len(scores))
```

리스트 자료형에 들어있는 개수를 세서 dict에 넣기

```
cnt_list = [1, 2, 3, 4, 5, 3, 1, 5, 4, 2, 4, 2, 1, 3, 4, 2, "", ""]

freq = {} # dict 선언

# freq[""] = 4 # 이렇게 하면 [] 안에 있는 값이 key 값이 되고 오른쪽에 있는게 value 값이 된다.
# 이 개념이 있어야 문제를 푼다.

print(freq)
```

```

for i in cnt_list:
    if i in freq: # 그래서 freq 안에 i 값이 존재하면 즉 key 값이 있다면
        freq[i] += 1 # freq[i] 에서 나온 값에 + 1을 더한다.
    else:
        freq[i] = 1 # 만약 freq 에 값이 없으면 새로 재정의 한다.
print(freq)

# {'': 2, 1: 3, 2: 4, 3: 3, 4: 4, 5: 2}

```

1~1000합

```

range_sum = 0

for i in range(1, 1001):
    range_sum += i
print(range_sum) # 500500

```

구구단의 3단만 출력한다.

```

dan = 3
for i in range(1, 10):
    print(dan, "*", i, "=", 3 * i)

```

홀수번째만 출력한다면?

```

dan = 3
for i in range(1, 10, 2):
    print(dan, "*", i, "=", 3 * i)

# 3 * 1 = 3
# 3 * 2 = 6
# 3 * 3 = 9
# 3 * 4 = 12
# 3 * 5 = 15

```

```

# 3 * 6 = 18
# 3 * 7 = 21
# 3 * 8 = 24
# 3 * 9 = 27

dan = 3
for i in range(1, 10):
    if idx % 2 == 1:
        print(dan, "*", i, "=", 3 * i)

# 3 * 1 = 3
# 3 * 3 = 9
# 3 * 5 = 15
# 3 * 7 = 21
# 3 * 9 = 27

```

```

my_list = [(1, 2), (3, 4), (5, 6)]
my_sum = 0
for (key, value) in my_list:
    my_sum += key+value
print("key{} , value : {}".format(key, value))

# key1 , value : 2
# key3 , value : 4
# key5 , value : 6

print(my_sum) # 21

```

#list comprehension

for문과 if 문을 한 문장으로 만들 수 있다.

```

temp_list = [1, 2, 3, 4, 5, 6, 7, 8, 9]

# list 컴프리헨션 - list에서 연산이 가능하다.
# 가운데 for 문 넣고 이 결과값이 왼쪽에 오른쪽은 if 문이 나온다.

score01 = [idx * 3 for idx in temp_list if idx % 2 == 0]
print(score01)

# [6, 12, 18, 24]

```

올림픽은 4년 마다 개최된다.

2000 ~ 2050년까지 중 올림픽이 개최되는 년도를 출력

```
# 1. 일반 for 형식
for idx in range(2000, 2051):
    if idx % 4 == 0:
        print(idx)

# 2000 ... 2048

# 2. range에 간격 두기
for idx in range(2000, 2051, 4):
    print(idx)

# 2000 ... 2048

# 3. list comprehension
print([idx for idx in range(2000, 2051, 4)]) # [2000, 2004, 2008, 2012, 2016, 2020, 2024, 2028, 2032, 2036, 2040, 2044, 2048]
```

아래 리스트에서 20보다 작은 3의 배수를 출력하라

```
list_value1 = [12, 21, 12, 14, 30, 38, 34, 18]

# 1. 일반 for 형식
for i in list_value1:
    if i < 20:
        if i % 3 == 0:
            print(i)

# 12
# 12
# 18

for i in list_value1:
    if i < 20 and i % 3 == 0:
        print(i)

# 12
```

```
# 12
# 18

# 2. list comprehension
print([i for i in list_value1 if i < 20 if i % 3 == 0]) # [12, 12, 18]
```

아래 리스트에서 세글자 이상의 문자를 출력하라

```
list_value = ['I', 'am', 'study', 'python', 'language', '!']

for value in list_value:
    if len(value) >= 3:
        print(value)

# study
# python
# language
```

아래 리스트에서 대문자만 화면에 출력하라

```
# isupper() , islower()
list_value2 = ['I', 'A', 'study', 'PYTHON', 'JSLIM', 'for', 'if']

# 1. 일반 for 형식
for i in list_value2:
    if i.isupper(): # 애는 true false로 반환한다.
        print(i)

# I
# A
# PYTHON
# JSLIM

# 2. list comprehension 형식
print([i for i in list_value2 if i.isupper()])

# ['I', 'A', 'PYTHON', 'JSLIM']
```


아래 리스트에 파일 이름이 저장되어 있다.

확장자를 제거하고 파일 이름만 출력하라

```
list_value3 = ['greeting.py', 'ex01.py', 'introm.hwp', 'bigdata.doc']

# 1. 일반 for 형식
for i in list_value3:
    print(i.split(".")[0])

# greeting
# ex01
# introm
# bigdata

# 2. list comprehension 형식
print([i.split(".")[0] for i in list_value3]) # ['greeting', 'ex01', 'introm', 'bigdata']
```

아래리스트에서 첫글자를 대문자로 출력하라

```
list_value5 = ['dog', 'cat', 'pig', 'parrot', 'horse']

for value in list_value5:
    print(value.capitalize())

# Dog
# Cat
# Pig
# Parrot
# Horse
```

문자열을 range로 for 문이 가능하다.

```
word = "Handsome"
for idx in word:
    print('word - ', idx, end=',') # word - H,word - a,word - n,word - d,word - s,word - o,word -
```

```
my_dict = {"name": "jslim",
"age": 48,
"city": "seoul"}

print()
cnt = 1
print(my_dict.pop())
len(my_dict.keys())
for key in my_dict.keys():
    if key == "city":
        print(key, end="")
    else:
        print(key, end=", ")

cnt = 1
for key in my_dict.keys():
    if cnt == len(my_dict.keys()):
        print(key, end="")
    else:
        print(key, end=", ")
    cnt += 1

# cityage,name,city,age,name
```

upper()

전체 문자를 대문자로 출력하라

```
fruit_name = 'FineApple'

print(fruit_name.upper())
for name in fruit_name:
    if name.isupper():
        print(name, end="")
    else:
        print(name.upper(), end="")
```

```
# result #  
# FINEAPPLE
```

break- 해당 반복을 종료한다. / **continue** - 밑 부분을 실행하지 않고 다음 반복으로 넘어간다. ,

#for else 문 - for문이 다 돌면 else를 실행한다.

```
# break, continue  
numbers =[14, 3, 6, 17, 34, 25]  
  
# for else - for문을 다돌았을때 else를 실행한다.  
for num in numbers:  
    if num == 34:  
        continue  
    print(num)  
    break  
else:  
    print("Not Found") # Not Found
```

구구단 5단 까지 출력

```
for i in range(2, 10):  
    if i == 6:  
        break  
    for j in range(1, 10):  
        print("{} * {} = {}".format(i, j, i*j))  
    print()
```

5단만 출력

```
for i in range(2, 10):  
    if i != 5:  
        continue  
    for j in range(1, 10):  
        print("{} * {} = {}".format(i, j, i * j))  
    print()
```

5단만 빼고 출력

```
for i in range(2, 10):
    if i == 5:
        continue
    for j in range(1, 10):
        print("{} * {} = {}".format(i, j, i * j))
    print()
```

예제

```
string = \
"""
저는 파이썬 강의중인 임섭순입니다.
주소는 광주광역시 입니다.
나이는 숫자에 불과합니다.
"""

sentences = []
words = []

for s in string.split("\n"):
    if s == "":
        continue
    sentences.append(s)
    for b in s.split(" "):
        words.append(b)
    print(sentences, len(sentences))
    print(words, len(words))

# ['저는 파이썬 강의중인 임섭순입니다.', '주소는 광주광역시 입니다.', '나이는 숫자에 불과합니다.']
# ['저는', '파이썬', '강의중인', '임섭순입니다.', '주소는', '광주광역시', '입니다.', '나이는', '숫자에', '']
```

[실습] 분류 정확도

```
answer = [1, 0, 2, 1, 0]
predict = [1, 0, 2, 0, 0]

count = 0
predict_count = 0
for i in answer:
    if predict[count] == i:
        predict_count += 1
```

```

count += 1

print(predict_count)
print("{}퍼센트 확률로 맞췄습니다.".format(predict_count/len(predict) * 100))

acc = 0
for i in range(len(answer)):
    fit = answer[i] == predict[i]
    acc += int(fit) * (100 / len(answer)) # 여기서 true false가 나온걸 int 형으로 바꿔서 값을 넣어준다.
print("분류 정확도", acc)

```

enumerate - 반복문 사용시 인덱스 번호와 컬렉션 요소를 확인 할 수 있다. / 몇번째 실행인지 알 수 있다.

```

# 반복문 사용 시 몇번째 반복문인지 확인
# 인덱스 번호와 컬렉션 요소를 확인 할 수 있다.

# list = {} - > set 또는 dict
list_value6 = ['SQL', 'R', 'TEXT-MINING', 'DATA-MINING', 'PYTHON']
for idx, value in enumerate(list_value6):
    print("{} 번째 인덱스 | 값 {}".format(idx, value))

word_vec = ["corona", "corona", "corona", "corona", "big", "big", "big", "big"]
print(word_vec.count("big"))

# 0 번째 인덱스 | 값 SQL
# 1 번째 인덱스 | 값 R
# 2 번째 인덱스 | 값 TEXT-MINING
# 3 번째 인덱스 | 값 DATA-MINING
# 4 번째 인덱스 | 값 PYTHON

```

#while - 해당 조건을 만족할때 까지 반복 한다.

```

# while <expression> : 조건문
#     statement
#     증감식 or break 문

n = 5
while n > 0: # 이 부분이 조건절로 만족할때 까지 반복한다.
    print(n)
    n -= 1 # 그래서 이렇게 증감식으로 조건을 만족하게 한다.

a = ['foo', 'bar', 'baz']

```

```
while a :  
    print(a.pop())
```

while 이용하여 1~100까지의 합, 5의 배수의 합, 3의 배수이면서 5의 배수가 아닌 합

```
sum100 = 0  
value7 = 0  
value8 = 0  
count = 1  
while count < 101:  
    sum100 += count  
    if count % 5 == 0:  
        value7 += count  
    if count % 5 != 0 and count % 3 == 0:  
        value8 += count  
    count += 1  
print(sum100)  
print(value7)  
print(value8)  
print("1")
```

while ~ else 는 while이 조건에 만족했을때 else 로직을 따르게 된다.

```
cnt = 10  
while cnt > 0:  
    if cnt == 11:  
        break  
    cnt -= 1  
    print(cnt)  
else:  
    print('while ~ else')
```

'Python' 카테고리의 다른 글

[Python] 파이썬 기초 10 - 클래스에 대한 정의와 사용법

[Python] 파이썬 기초 9 - 패키지와 모듈에 대한 정의와 다양한 함수 형태□

[Python] 파이썬 기초 8 - 반복문(for , while)에 대한 정의와 기본적인 함수 사...

[Python] 파이썬 기초 7 - 조건문(IF, elif ,else)에 대한 정의와 기본적인 함수 사...

[Python] 파이썬 기초 6 - Date type에 대한 정의와 기본적인 함수 사용법□

[Python] 파이썬 기초 5 - 집합(set)에 대한 정의와 기본적인 함수 사용법□

python for문

python while문

파이썬 for 문

파이썬 while문



나무늘보스

혼자 끄적끄적하는 블로그 입니다.