

데이터 처리를 위한 Python 프로그래밍 입문

6-2강. 반복문 : range

ERICA 2018-2

강의 내용

- ▶ range() 함수 활용한 for문
- ▶ 문자열 활용한 for문
- ▶ 중첩 for문

range 함수 활용 for문(1/8)

▶ range 함수 활용 반복문

- ▶ Range() 범위의 모든 값들 각각에 대해 반복을 수행 가능

표기	내 용
for variable in range() : True_statements	range 범위내의 variable 각각에 대해서 True_statements를 반복적으로 수행

▶ 예제코드

- ▶ 0부터 3까지 모든 값에 대해 range()함수 활용하여 for 문 활용 출력
- ▶ 반복문을 활용하여 “LOOP”라는 문구를 4번 출력

```
>>> for count in range(4):
      print(count)
```

```
0
1
2
3
>>>
```

```
>>> for num in range(4):
      print('LOOP')
```

```
LOOP
LOOP
LOOP
LOOP
~~~
```

range 함수 활용 for문(2/8)

▶ range 함수 활용 반복문 : 입력된 수 만큼 문장 반복

▶ 상황 : 'You did a good job!'이라는 문장을

사용자로부터 입력받은 반복 횟수만큼 연속적으로
결과 화면과 같이 출력하는 프로그램

▶ 결과 화면

```
Enter the number of times to repeat : 3
You did a goog job!
You did a goog job!
You did a goog job!
>>>
```

▶ 해결 코드

```
File Edit Format Run Options Window Help
repeat_num=int(input('Enter the number of times to repeat : '))

for i in range(repeat_num):
    print('You did a goog job!')
```

range 함수 활용 for문(3/8)

- ▶ range 함수 활용 반복문 : 팩토리얼
 - ▶ 상황 : 1부터 시작해 사용자가 입력한 정수까지
팩토리얼 값을 프로그램으로
결과 화면과 같이 출력하는 프로그램
 - ▶ 결과 화면

```
Enter the number : 4
4 ! = 24
>>>
```

range 함수 활용 for문(4/8)

▶ range 함수 활용 반복문 : 팩토리얼

▶ 해결 코드

```
File Edit Format Run Options Window Help
n=int(input('Enter the number : '))
fact = 1
for i in range(1, n+1):
    fact *= i
print(n, '! = ', fact)
```

range 함수 활용 for문(5/8)

- ▶ range 함수 활용 반복문 : 합 구하기
 - ▶ 상황 : 사용자가 입력한 시작 값, 끝 값까지
입력 받은 증가 값을 사용하는 프로그램으로
결과 화면과 같이 출력하는 프로그램
 - ▶ 결과 화면

```
시작값 입력 : 2  
끝값 입력 : 300  
증가값 입력 : 3  
2에서 300까지 3씩 증가함 값의 합 : 15050
```

range 함수 활용 for문(6/8)

▶ range 함수 활용 반복문 : 합 구하기

▶ 해결 코드

```
File Edit Format Run Options Window Help
num1 = int(input("시작값 입력: "))
num2 = int(input("끝값 입력: "))
num3 = int(input("증가값 입력: "))

total = 0

for i in range(num1, num2+1, num3):
    total = total + i

print("%d에서 %d까지 %d씩 증가한 값의 합: %d" %(num1, num2, num3, total))
```


range 함수 활용 for문(7/8)

- ▶ range 함수 활용 반복문 : 구구단 구하기
 - ▶ 상황 : 사용자가 입력한 값의 구구단 프로그램으로
결과 화면과 같이 출력하는 프로그램
 - ▶ 결과 화면

```

몇 단 ? 7
7 X 1 = 7
7 X 2 = 14
7 X 3 = 21
7 X 4 = 28
7 X 5 = 35
7 X 6 = 42
7 X 7 = 49
7 X 8 = 56
7 X 9 = 63

```

range 함수 활용 for문(8/8)

▶ range 함수 활용 반복문 : 구구단 구하기

▶ 해결 코드

```
File Edit Format Run Options Window Help
dan = int(input(" 몇 단 ? "))
for i in range(1, 10, 1) :
    print(" %d X %d = %2d" % (dan, i, dan*i))
```

문자열 활용 for문(1/4)

▶ 문자열 활용 반복문

- ▶ 문자열의 모든 문자(spelling)에 대해 특정한 코드를 반복 수행 가능

표기	내 용
for character in 'string': True_statements	문자열 범위내의 character 각각에 대해서 True_statements를 반복적으로 수행

- ▶ 예제코드 : 'string', '파이썬'이라는 문자열의 모든 스펠링을 for 문 활용 출력

```
>>> for spelling in 'string':
      print(spelling)
```

```
s
t
r
i
n
g
>>>
```

```
>>> for spelling in '파이썬':
      print(spelling)
```

```
파
이
썬
>>>
```

문자열 활용 for문(2/4)

▶ 문자열 활용 반복문

- ▶ 상황 : 'Python'이라는 문자열을 사용해 string내의 spell 각각을 연속적으로 결과 화면과 같이 출력하는 프로그램

▶ 결과 화면

```
Current Letter : P  
Current Letter : y  
Current Letter : t  
Current Letter : h  
Current Letter : o  
Current Letter : n  
>>>
```

▶ 해결 코드

```
>>> for letter in 'Python':  
    print('Current Letter : ', letter)
```

문자열 활용 for문(3/4)

- ▶ 문자열 활용 반복문 : alphabet
 - ▶ 상황 : 1번째 알파벳부터 6번째 알파벳 순으로 구성된 문자열 'abc가나다'를 결과 화면과 같이 출력하는 프로그램
 - * alphabet이 아닌 가나다 포함 : 영문, 한글 혼용 가능 확인
 - ▶ 결과 화면

```
1 번째 알파벳 a
2 번째 알파벳 b
3 번째 알파벳 c
4 번째 알파벳 가
5 번째 알파벳 나
6 번째 알파벳 다
>>> |
```

문자열 활용 for문(4/4)

- ▶ 문자열 활용 반복문 : alphabet(계속)
 - ▶ 해결 코드

```
File Edit Format Run Options Window Help
order = 1
word = 'abc가나다'

for ABC in word:
    print(order, '번째 알파벳은', ABC)
    order +=1
```

중첩 for문(1/4)

▶ 중첩 for문의 개념

- ▶ 중첩 for문은 for문 내부에 또 다른 for문이 들어있는 형태
- ▶ 중첩 for문의 기본 코드

```
for i in range (0, 3, 1) :  
    for k in range(0, 2, 1) :  
        print("파이썬은 꿀잼입니다. ^^ (i값: %d, k값: %d)" % ( i, k ))
```

출력 결과

```
파이썬은 꿀잼입니다. ^^ (i값: 0, k값: 0)  
파이썬은 꿀잼입니다. ^^ (i값: 0, k값: 1)  
파이썬은 꿀잼입니다. ^^ (i값: 1, k값: 0)  
파이썬은 꿀잼입니다. ^^ (i값: 1, k값: 1)  
파이썬은 꿀잼입니다. ^^ (i값: 2, k값: 0)  
파이썬은 꿀잼입니다. ^^ (i값: 2, k값: 1)
```

중첩 for문(2/4)

▶ 중첩 for문 : 구구단 출력

▶ 상황 : 구구단 2단부터 9단까지 출력

▶ 결과 화면

```
2 X 1 = 2  
2 X 2 = 4  
2 X 3 = 6
```

...

```
9 X 7 = 63  
9 X 8 = 72  
9 X 9 = 81
```

▶ 해결 코드

```
File Edit Format Run Options Window Help  
for i in range(2, 10, 1):  
    for k in range(1, 10, 1):  
        print("%d * %d = %2d" % (i, k, i*k))  
    print("")
```


중첩 for문(3/4)

- ▶ 중첩 for문 : 구구단 옆으로 출력
 - ▶ 상황 : 구구단 2단부터 9단까지 옆으로 출력하기
 - ▶ 결과 화면

```
# 2단 # # 3단 # # 4단 # # 5단 # # 6단 # # 7단 # # 8단 # # 9단 #
2X 1= 2 3X 1= 3 4X 1= 4 5X 1= 5 6X 1= 6 7X 1= 7 8X 1= 8 9X 1= 9
2X 2= 4 3X 2= 6 4X 2= 8 5X 2=10 6X 2=12 7X 2=14 8X 2=16 9X 2=18
2X 3= 6 3X 3= 9 4X 3=12 5X 3=15 6X 3=18 7X 3=21 8X 3=24 9X 3=27
2X 4= 8 3X 4=12 4X 4=16 5X 4=20 6X 4=24 7X 4=28 8X 4=32 9X 4=36
2X 5=10 3X 5=15 4X 5=20 5X 5=25 6X 5=30 7X 5=35 8X 5=40 9X 5=45
2X 6=12 3X 6=18 4X 6=24 5X 6=30 6X 6=36 7X 6=42 8X 6=48 9X 6=54
2X 7=14 3X 7=21 4X 7=28 5X 7=35 6X 7=42 7X 7=49 8X 7=56 9X 7=63
2X 8=16 3X 8=24 4X 8=32 5X 8=40 6X 8=48 7X 8=56 8X 8=64 9X 8=72
2X 9=18 3X 9=27 4X 9=36 5X 9=45 6X 9=54 7X 9=63 8X 9=72 9X 9=81
```

중첩 for문(4/4)

▶ 중첩 for문 : 구구단 옆으로 출력(계속)

▶ 해결코드

```
File Edit Format Run Options Window Help
guguLine = ""
for i in range(2, 10):
    guguLine = guguLine + (" # %d단 #" % i)
print(guguLine)

for i in range(1, 10):
    guguLine = ""
    for k in range(2, 10):
        guguLine = guguLine + str("%2d*%2d=%2d" % (k, i, k*i))
    print(guguLine)
```

Thank you