

K G 아 이 티 뱅 크

파이썬

P Y T H O N

반복문for

반복문for

〈파일이름 : 01. for.py〉

❖ for : 지정된 범위 안에서 반복을 수행하려고 할 때 사용한다

리스트, 튜플 범위 내에서 각 요소들을 차례대로 지정된 변수로 대입 후 종속문장을 수행
즉, 요소의 개수만큼 반복

- for의 기본 구조

```
for 변수 in 리스트(또는 튜플, 딕셔너리):  
    for종속문장1  
    for종속문장2  
    ...  
다음 문장
```

- for의 기본 사용

```
count = [0, 1, 2, 3]  
  
for num in count:  
    print("num = ", num)
```

반복문for

<파일이름 : 01. for.py>

- for예제1

```
count = ["one", "two", "three",  
"four", "five"]
```

```
print("숫자세기")
```

```
for i in count:
```

```
    print(i)
```

```
count.reverse()
```

```
print("\n거꾸로세기")
```

```
for i in count:
```

```
    print(i)
```

- for예제2

```
jumsu = (90, 50, 60, 80, 40)
```

```
number = 1
```

```
for i in jumsu:
```

```
    if i >= 60:
```

```
        print("{0}번째 학생 : 합격
```

```
!".format(number))
```

```
    else:
```

```
        print("{0}번째 학생 : 불합격"
```

```
".format(number))
```

```
        number += 1
```

반복문for

〈파일이름 : 02. for.py〉

- for예제4

```
a = [(1, 2), (3, 4), (5, 6)]
for (a, b) in a:
    print("{0} + {1} = {2}".format(a, b, a + b))

for a in a:
    print("{0}".format(a))
```

- for예제5

```
filmFestival = {
    '최우수 작품상' : '택시운전사',
    '감독상' : '아이 캔 스피크',
    '남우주연상' : '송강호',
    '여우주연상' : '나문희',
}

for prize in filmFestival.keys():
    print(prize)

for winner in filmFestival.values():
    print(winner)

for prize_winner in filmFestival.items():
    print(prize_winner)

for prize, winner in filmFestival.items():
    print(prize, ":", winner)
```

문제

<파일이름 : 03.문제.py>

❖ for문제

1. while 무한반복문으로 0을 입력할 때 까지 list변수에 값을 입력 받고
for문을 이용하여 그 값들을 출력하고 합계를 출력하시오

반복문for

〈파일이름 : 04. range.py〉

- ❖ range : 범위를 지정하는 명령어. 숫자리스트를 자동으로 만들어 주는 함수
반복문에서 유용하게 사용

- range 사용법

```
range(a) : 0 부터 a - 1 까지(a는 제외)  
range(a, b) : a부터 b - 1 까지(b는 제외)
```

- range 예

```
print(range(5))  
print(list(range(5)))  
print(tuple(range(5)))  
print(list(range(10)))  
print(list(range(1, 10, 2)))  
print(list(range(5, 12)))  
print(list(range('a', 'f')))
```

반복문for

<파일이름 : 04. range.py>

❖ for + range : range를 for문에서 사용할 경우 list로 자동형변환 된다

- for + range

```
for i in range(5):  
    print("i = ", i)  
print()  
for i in range(3, 10):  
    print("i = ", i)
```

- for + range 활용1

```
sum = int()  
for x in range(1,11):  
    sum += x  
print("1부터 10까지의 합계 :",sum)
```


반복문for

<파일이름 : 04. range.py>

- for + range 활용2

```
print("\n1 ~ 10까지 홀수와 홀수의 합계")
result = int()
for x in range(1,11):
    if x % 2 != 0:
        print(x, end=', ')
        result += x

print("\n\n1~10까지 홀수의 합계 :",result)
```

- for + range 활용3

```
result = int()
for x in range(1,101):
    if x % 3 ==0:
        print(x, end=', ')
        result += x
print("\n\n1 ~ 100까지의
3의 배수의 합을 출력 :",result)
```

반복문for

<파일이름 : 04. range.py>

- for + range 활용4

```
num = int(input("정수 입력 : "))  
print("1 ~ %d사이의 짝수는 : "%num)  
for i in range(1,num+1):  
    if i % 2 == 0:  
        print(i)
```

반복문for

<파일이름 : 04. range.py>

❖ range에 증감옵션을 포함하여 변수가 1씩 증가하지 않고 다양하게 증감하게 할 수 있다

- 사용 방법

```
for 변수 in range(시작, 끝, 증감):  
    for의 종속문장1  
    for의 종속문장2  
    ...  
다음문장
```

- 사용 예

```
for x in range(0, 10, 2):  
    print(x)  
  
for x in range(10, 0, -1):  
    print(x)
```

문제

<파일이름 : 05.문제.py>, <파일이름 : 06.문제.py>

❖ for+range문제1

정수를 입력 받아 입력한 수부터 0까지 카운트다운을 출력하시오
([::-1] : 역순으로 출력기능 사용)

출력>

정수 입력 : 4

4 3 2 1

❖ for+range문제2

양의 정수를 4개 입력 받고, 짝수, 홀수를 구분하여 출력하시오

양수 입력 : -1
잘못된 입력입니다.
양수 입력 : 0
0는 0입니다.
양수 입력 : 2
2는 짝수입니다.
양수 입력 : 3
3는 홀수입니다.

문제

<파일이름 : 07.문제.py>

❖ for+range문제3

첫 날에 1원을 예금하고, 다음날에는 전날의 두 배를 예금하는
방식으로 한달(30일)동안 저축한 금액은?

한달 동안 저축한 금액 : 1073741823

문제

〈파일이름 : 08.문제.py〉, 〈파일이름 : 09.문제.py〉

❖ for+range문제4

정수를 입력 받아 소수인지 판별하시오
(소수 : 1과 자기자신으로만 나누어지는 수)

❖ for+range문제5

정수 두 개를 입력 받아 두 수의 공약수를
모두 출력하시오
최대공약수도 구해주세요

이중 for

〈파일이름 : 10. 이중for.py〉

❖ 이중 for문 : 두 개의 for문을 이용하여 복잡한 반복을 제어할 수 있다

- 이중 for문 예제

```
for i in range(3):  
    print("====={0}학년=====".format(i + 1))  
    for j in range(12):  
        print("----{0}학년--{1}반----".format(i + 1, j + 1))  
print("\n")
```

이중 for

〈파일이름 : 10. 이중for.py〉

- 이중 while로 구구단 만들기

```
a, b = 2, 1
while a < 10:
    while b < 10:
        print ("%d * %d = %d"%(a, b, a * b))
        b += 1
    a += 1
    b = 1
```

- 이중 for로 구구단 만들기

```
for a in range(2,10):
    for b in range(1,10):
        print ("%d * %d = %d"%(a, b, a * b))
```


다중루프

<파일이름 : 12.문제.py>

출력

```
*  
**  
***  
****  
*****
```

num

5

i	j
0	0
1	0 1
2	0 1 2
3	0 1 2 3
4	0 1 2 3 4

i는 별의 행을 결정. j는 한 행에 별을 몇 개 출력할지 결정합니다.

다중루프

<파일이름 : 13.문제.py>

출력

```
  *
 **
***
****
*****
```

num

5

i	j
0	0
1	0 1
2	0 1 2
3	0 1 2 3
4	0 1 2 3 4

j의 앞에 공백을 출력 후 *을 출력
공백과 *의 반복횟수가 num과 같습니다.

2차원 리스트

❖ 1차원 리스트는 선, 2차원 리스트는 면의 개념

2차원 리스트는 가로 x 세로의 표 형태이며 행(column)과 열(row)는 0부터 시작한다.

	열 0	열 1	열 2	열 3
행 0				
행 0				
행 0				

2차원 리스트

〈파일이름 : 16. 2차원리스트.py〉

❖ 2차원 리스트 생성 방법

리스트변수명 = [[요소, 요소], [요소, 요소], [요소, 요소]]

- 2차원 리스트 생성

```
a = [[10, 20], [30, 40], [50, 60]]  
print(a)  
  
a = [[10,20],  
      [30,40],  
      [50,60]]  
print(a)
```

	열 0	열 1
행 0	10	20
행 1	30	40
행 2	50	60

2차원 리스트

<파일이름 : 16. 2차원리스트.py>

❖ 2차원 리스트 접근 방법

2차원 리스트의 요소에 접근하거나 새로운 값을 할당할 때는 리스트 뒤에 []를 두 번 사용
첫 번째 []는 세로(row) 인덱스, 두 번째 []는 가로(column) 인덱스를 지정한다

리스트변수명[세로인덱스][가로인덱스] = 값

	열 0	열 1	열 2	열 3
행 0	[0][0]	[0][1]	[0][2]	[0][3]
행 1	[1][0]	[1][1]	[1][2]	[1][3]
행 2	[2][0]	[2][1]	[2][2]	[2][3]

2차원 리스트

〈파일이름 : 16. 2차원리스트.py〉

- 2차원 리스트 접근 방법

```
a = [[10, 20], [30, 40], [50, 60]]  
print(a[0][0])
```

```
a[0][0] = 1000  
a[1][1] = 2000  
a[2][1] = 3000  
print(a)
```

	열 0	열 1
행 0	1000	20
행 1	30	2000
행 2	50	3000

2차원 리스트

〈파일이름 : 16. 2차원리스트.py〉

- 2차원 튜플 접근 방법

```
a = [(10, 20), (30, 40), (50, 60)] #리스트 안에 튜플을 넣음
b = ([10, 20], [30, 40], [50, 60]) #튜플 안에 리스트를 넣음
c = ((10, 20), (30, 40), (50, 60)) #튜플 안에 튜플을 넣은 2차원 튜플
a[0][0] = 1000
b[0][1] = 2000
c[0][2] = 3000
print(a)
print(b)
print(c)
```

2차원 리스트

<파일이름 : 16. 2차원리스트.py>

❖ 반복문을 이용하면 list에 요소의 입력, 출력을 간편하게 할 수 있다

- 2차원 리스트 반복문

```
a = [[10, 20], [30, 40], [50, 60]]  
for x, y in a:  
    print(x, y)
```



2차원 리스트

〈파일이름 : 16. 2차원리스트.py〉

- 2중 for문을 이용한 2차원 배열

```
a = [[10,20],  
      [30,40],  
      [50,60]]  
  
for i in range(len(a)):  
    for j in range(len(a[i])):  
        print(a[i][j],end = ' ')  
    print()
```

- 2중 while문을 이용한 2차원 배열

```
a = [[10,20],  
      [30,40],  
      [50,60]]  
  
i = 0;  
while i < len(a):  
    j = 0  
    while j < len(a[i]):  
        print(a[i][j], end = ' ')  
        j += 1  
    print()  
    i += 1
```

2차원 리스트

〈파일이름 : 16. 2차원리스트.py〉

- 2중 for문을 이용한 2차원 리스트 초기화

```
a = []  
  
for i in range(3):  
    line = []  
    for j in range(2):  
        line.append(0)  
    a.append(line)  
print(a)  
a[0][0] = 10  
a[0][1] = 20  
print(a)
```

2차원 리스트

〈파일이름 : 16. 2차원리스트.py〉

❖ 표현식을 이용하여 코드를 더 간결하게 작성 가능

- 리스트 표현식을 이용한 2차원 리스트

[표현식 for 항목 in 반복가능객체 if 조건]

```
a = [[0 for j in range(2)] for i in range(3)]  
print(a)
```

```
a = [[0]*2 for i in range(3)]  
print(a)
```

문제

<파일이름 : 11.문제.py>

❖ 이중 리스트 for문 문제

3층짜리 빌라가 있다. 한 층에 4개 집이 있고,
각 층, 각 집에 전단지를 붙여야 한다.
1층부터 3층까지 차례로 전단지를 붙이시오
(아래의 리스트를 사용하여 작성하시오)

101호	부착완료
102호	부착완료
103호	부착완료
104호	부착완료
1층에	모두 부착완료
201호	부착완료
202호	부착완료
203호	부착완료
204호	부착완료
2층에	모두 부착완료
301호	부착완료
302호	부착완료
303호	부착완료
304호	부착완료
3층에	모두 부착완료

```
apart = [[101, 102, 103, 104], [201, 202, 203, 204], [301, 302, 303, 304]]
```

문제

<파일이름 : 17.문제.py>

❖ 이중 리스트 문제

8x8 2차원 리스트를 생성하여
1부터 64까지 차례대로 채워주소