

# for

2019년 5월 29일 수요일    오후 4:50

## ❖ for 반복문 사용법

**for 변수 in 컨테이너 :**

**실행할 명령1**

**실행할 명령2**

    .  
    .  
    .

-> 컨테이너에 있는 값을 차례대로 꺼내어 변수에 넣고,  
    아래있는 명령을 실행한다.  
    컨테이너에서 더 이상 꺼낼 값이 없을때 까지 반복.

```
for animal in animals:  
    print(animal)
```

```
강아지  
땅다람쥐  
코알라  
토끼  
하이에나  
토끼
```

-> 컨테이너의 자리에는 문자열이 들어갈 수도 있다.

## ○ range( )

for문에서 사용할 수 있는 내장함수

연속된 숫자의 범위를 지정

```
for n in range(0, 3):  
    print(n)
```

```
0  
1  
2
```

-> list의 [0, 1, 2]와 같음

    앞의 숫자를 생략하면 0부터, 시작 숫자를 지정할 수도 있음

    두번째 숫자는 포함하지 않음

## ○ 중첩 반복문

```
#중첩 for문, 구구단  
for j in range(2, 10):  
    for i in range(1, 10):  
        print('{}x{}={}'.format(j, i, j*i))
```

```
2x1=2
2x2=4
2x3=6
2x4=8
2x5=10
2x6=12
2x7=14
2x8=16
2x9=18
3x1=3
3x2=6
3x3=9
3x4=12
3x5=15
3x6=18
```

...9단까지

- Comprehension

리스트를 만들 때 for문 문법을 사용해서 만드는 것

```
# comprehension
numbers = [1,2,3,4,5,6,7,8,9,10]
odd_numbers = []

for number in numbers:
    if number % 2 == 1:
        odd_numbers.append(number)

print(odd_numbers)

comp = [number for number in numbers if number % 2 == 1]
print(comp)
```

```
[1, 3, 5, 7, 9]
[1, 3, 5, 7, 9]
```

-> 원래는 for문 안에 넣어서 처리해야하는 일을  
배열을 만들 때 그 안에서 처리할 수 있음