

Python 문법

정규 수업 3차시



저번 시간에 배운 내용들

- 파이썬에 사용하는 자료형들(int, float, bool 등등)
 - 자료형 변환
 - 입력 방법
- 리스트 자료구조 기본 개념
 - if 및 for문 기본 개념

기본적인 자료형들

```
int_n = 123 # <class 'int'>
float_n = 1.0 # <class 'float'>
string = '나는야유채호' # <class 'str'>
boolean = True # <class 'bool'>
```

저번 시간 복습

```
string = '1' #문자열임  
string = int(string) #숫자로 형변환!  
print(string + 1) # 1+1 = 2므로 출력은 2
```

저번 시간 복습

```
a = input()  
print(a)
```

저번 시간 복습

```
if (조건):  
    print( '참' )  
else:  
    print( '거짓' )
```

저번 시간 복습



```
1 a = not True
```

```
2 b = True and False
```

```
3 c = False or True
```

저번 시간 복습

배열

리스트

저번 시간 복습

배열	리스트
고정적인 크기	유동적인 크기
1개의 자료형	다양한 자료형
연속적	비연속적

저번 시간 복습

```
li = []  
li = [1, 2, 3, 4, 5]  
li = ['hi', 'my', 'name', 'is', 'jh lee']  
li = [1, '2', 3.0]  
li.append("이거 추가 됨ㅋㅋ")  
li.pop()
```

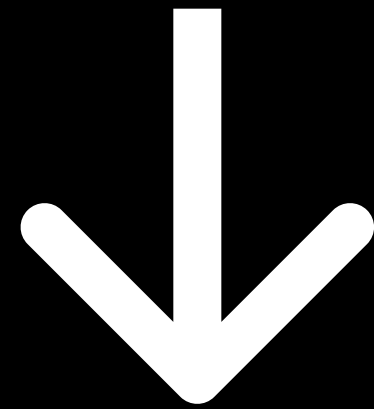
저번 시간 복습

1부터 9까지 출력

```
for i in range(1, 10, 1):  
    print(i)
```

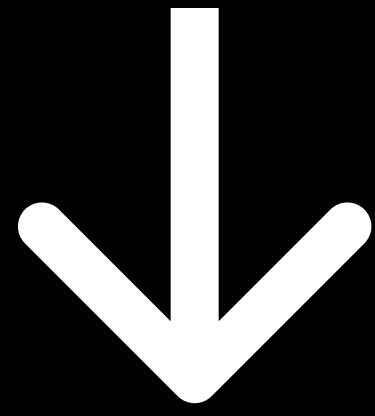
저번 시간 복습

```
range(1, 10, 1):
```



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

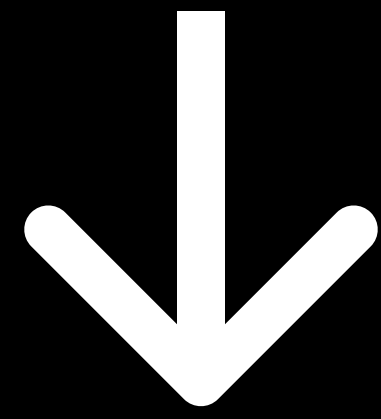
저번 시간 복습



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

$i = 1$

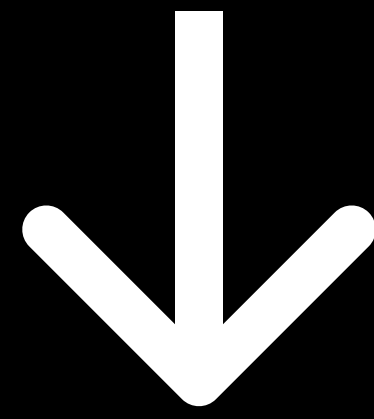
저번 시간 복습



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

$i = 2$

저번 시간 복습



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

$i = 3$

저번 시간 복습

$1 \leq i < 10$ 의 범위를
 $i++$ 하며 차례대로 순회

오늘 배울 내용

그래서 오늘 배울건..

- 리스트 인덱싱, 슬라이싱
 - 튜플, 딕셔너리
 - if,for문의 활용
 - 함수 기본 개념

이번에는 리스트 요소에 이렇게 접근했습니다



```
1 li = [1,2,3,4,5,6]  
2 print(li[3])#li의 3번 요소 : 4
```

이번에는 2차원 리스트 요소에 접근해봅시다



```
1 li = [[1,2,3],[4,5,6]]  
2 print(li[1][2])#출력 값은?
```

리스트 인덱싱

```
li[1][2]
```

||

li의 1번 요소의 2번 요소

리스트 인덱싱

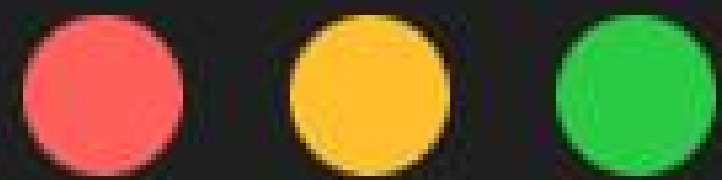
이러한 접근은
리스트 안에 리스트가
있을 때만 가능

리스트 내의 1에 접근하려면?



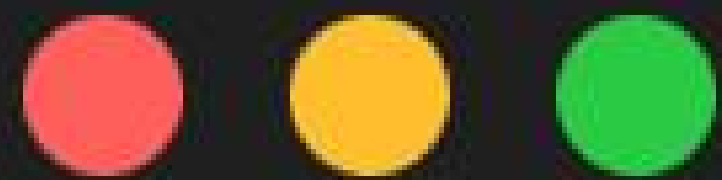
```
1 li = [[1,3],2,5],[6,7],9]
```

이런식으로 접근할 수 있다!



```
1 li = [[1,3],2,5],[6,7],9]
2 print(li[0][0][0])
```

여기서 9에 접근하려면 이런 방법도 있지만..



```
1 li = [[1,3],2,5],[6,7],9]
2 print(li[2])
```


-1를 사용해서 이렇게도 표현이 가능하다!



```
1 li = [[1,3],2,5],[6,7],9]
2 print(li[-1])
```

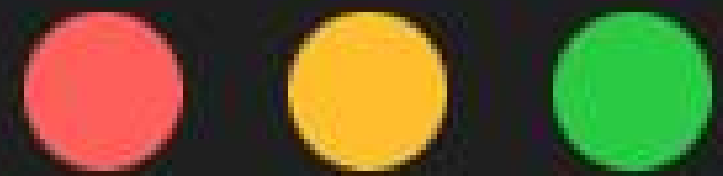
리스트 슬라이싱

리스트를 자르고 싶을 때에는 어떻게 할까!?!

이 친구를 [2,3,4]로 만들고 싶으면 어떻게 할까?

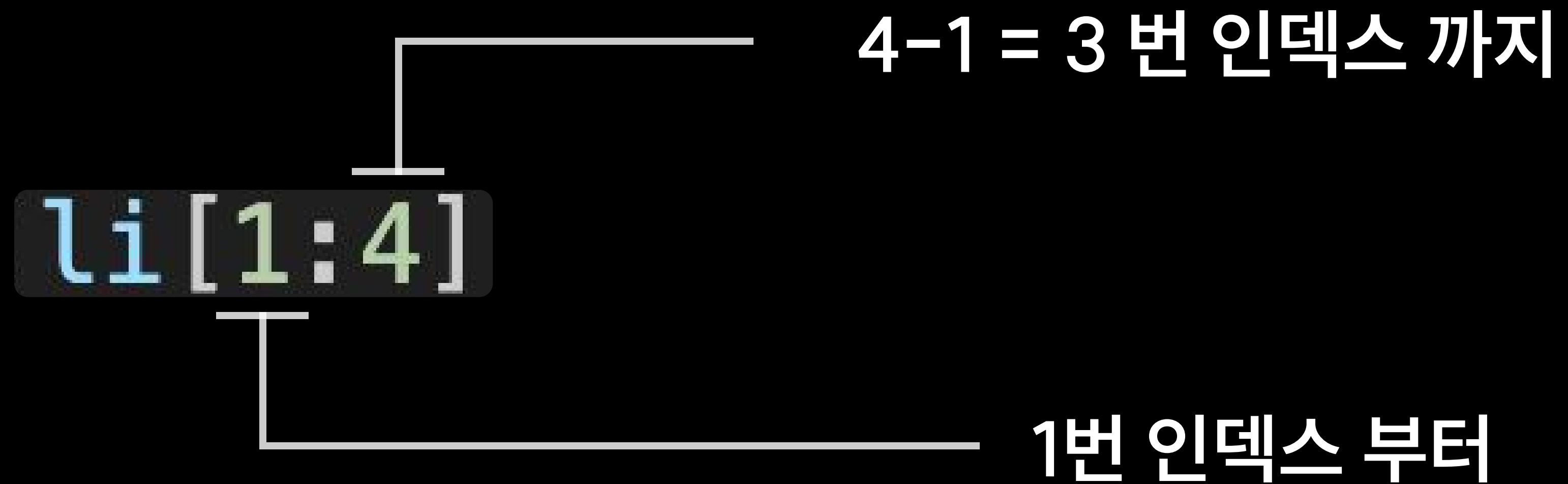
```
1 li = [1,2,3,4,5]
```

이런식으로 표현할 수 있다!



```
1 li = [1,2,3,4,5]
2 print(li[1:4])
```

리스트 슬라이싱



“li 리스트를 1~3 까지만 표현해라”

리스트 슬라이싱



```
1 li = [1,2,3,4,5]
2 print(li[1:4] == [2,3,4])
3 #출력은 True
```

“li 리스트를 1~3 까지만 표현해라”

리스트 슬라이싱



```
1 li = [1,2,3,4,5]
2 print(li[1:4] == [2,3,4])#True
3 print(li[:] == [1,2,3,4,5])#True
4 print(li[:3] == [1,2,3])#True
5 print(li[2:] == [3,4,5])#True
```

이런식으로도 표현할 수 있습니다.

튜플

리스트는 리스트인데 값이 안바뀌는 리스트가 있다?!

“튜플”



```
1 tp = (1,2,3,4,5)
```

요소의 추가, 변경이 불가함



```
1 tp = (1,2,3,4,5)
```

```
2 tp[0] = 3 #에러
```

dictionary는 무슨 뜻인가요?

영어사전

다른 어학정보 2 ▾

dictionary

미국식 [ˈdɪkʃənəri] 영국식 [ˈdɪkʃənri]

명사

1 (일반적인) 사전

a Spanish-English **dictionary**
스페인어-영어 사전

2 (특정 분야의) 사전

a **dictionary** of mathematics
수학 (용어) 사전

3 (컴퓨터의) 사전[딕셔너리]

[영어사전 결과 더보기](#)

“사전”

딕셔너리

{ Key : Value }



```
1 _dict = {'name' : 'chaeho'}
```

Key와 Value 한 쌍으로 이루어져 있다.

딕셔너리



```
1 _dict = {  
2     'name' : 'chaeho',  
3     'age' : 17,  
4     'hobby' : ['sleep', 'work', 'para']  
5 }
```

딕셔너리



1 #이런식으로 요소 추가나 변경이 가능하다.

2 `_dict = {'school' : 'sunrin'}`

3

4 #이런식으로 key를 사용해 value에 접근할 수 있다.

5 `_dict['name']`

딕셔너리, 실습 해봅시다!

**선린인터넷고등학교의 정보를 가진 딕셔너리 변수를
자유롭게 만드시오!**

if문의 활용

dksl!!! 아니!! 선배님!!

파이썬에는 삼항연산자 같은 거 없나요?!?!

if문의 활용

비슷한게 있습니다!!

if문의 활용



1 (참일 때) *if* (조건문) *else* (거짓일 때)

if문의 활용



```
1 print('짝') if n%2 == 0 else print('홀')
```

if문의 활용

리스트 안에 특정 원소가 있는지 검사하고 싶어요!

if문의 활용

네, 알겠습니다.

if문의 활용



```
1 if (특정원소) in (리스트):  
2     print( '있어요' )  
3 else:  
4     print( '없어요' )
```

if문의 활용



```
1 li = [1,2,3]
2 if 1 in li:#리스트에 1이 있는가?
3     print('굿')#출력
4 else:
5     print('힝')
```

1 ~ n의 원소를 가지는 리스트를 만들고 싶다..
어떻게 해야할까?



```
1 li = [1,2,3,4.....,n]
```


이렇게 표현할 수 있지만.. 너무 길다!!



```
1 li = []  
2 for i in range(1, n+1):  
3     li.append(i)
```

그래서..! 이런식으로 축약해서 쓸 수 있다!



```
1 li = [i for i in range(1,n+1)]
```

한 줄 코드로 실습 해봅시다!

**for문을 사용하여, 0~20범위의 2의 배수를 담은
리스트를 만들어 출력해봅시다.**

range() 말고 다른 것도 쓸 수 있음




```
1 for i in (여기에 어떻게 들어갈 수 있을까) :  
2     print(i)
```

range() 말고 다른 것도 쓸 수 있음



```
1 for i in (여기에 어떻게 들어갈 수 있을까) :  
2     print(i)
```

**인덱싱이 되는 자료형이면
이런식으로 쓸 수 있음**
ex) list, tuple, dict, str, set...



```
1 li = [1,2,3]
2 for i in li:
3     print(i)
4
5 _str = '123'
6 for i in _str:
7     print(i)
```

for문, 실습 해봅시다!

**for문을 사용하여 문자열 'sunrin'에서
모음을 빼고 출력해봅시다.**

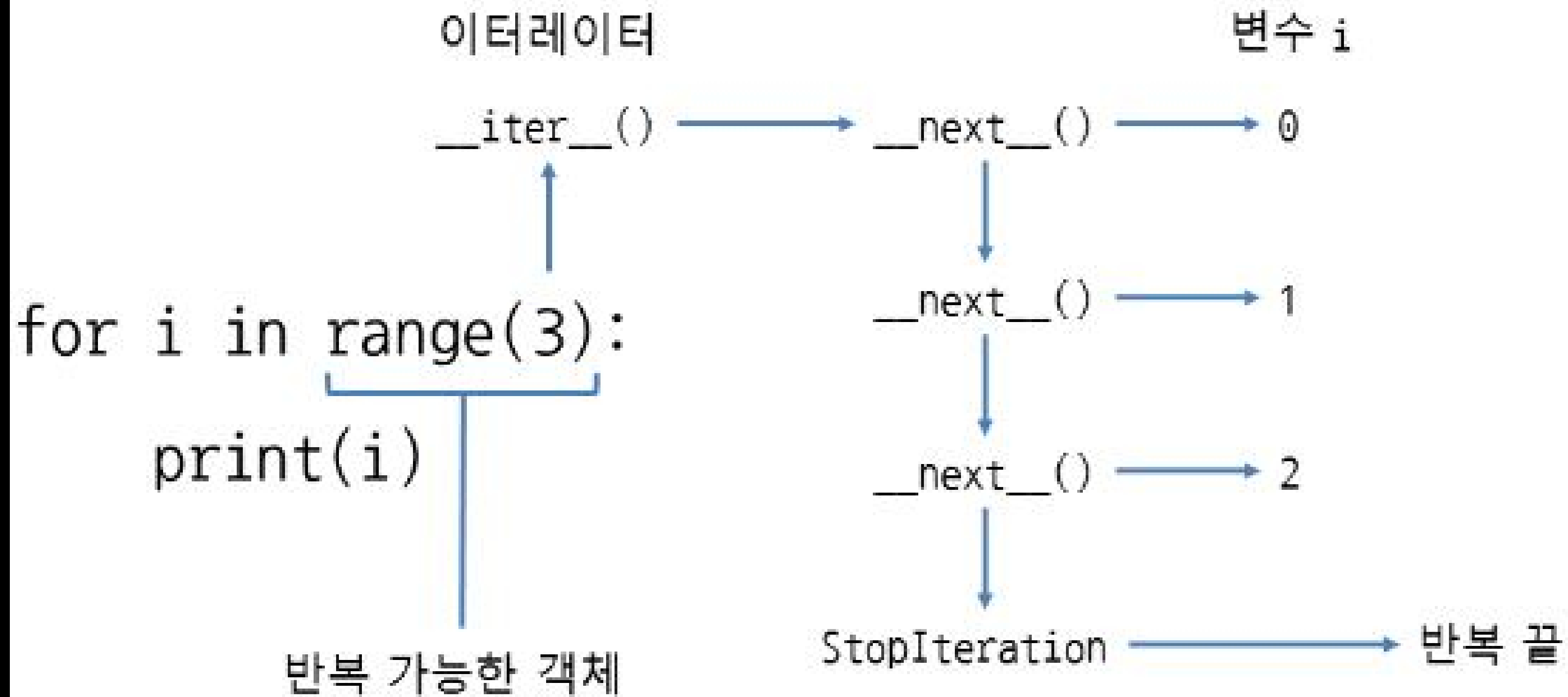
iterable(이터러블)은 무엇인가요?

“반복할수 있는” 이라는 뜻.

이 속성이 있는 객체는 반복문에 사용할 수 있다.

ex) list, tuple, dict, set s...

for문의 활용



for문의 활용



```
1  li = [1,2,3,4,5]
2  li_iter = iter(li)
3
4  print('li_iter의 타입 : ',type(li_iter))
5  print('li_iter의 값 : ',li_iter)
6
7  print(li_iter.__next__())
8  print(li_iter.__next__())
9  print(li_iter.__next__())
10 print(li_iter.__next__())
11 print(li_iter.__next__())
```

함수

맨날 함수 함수 거리는데..
이건 어떻게 쓰나요??!?!!!!

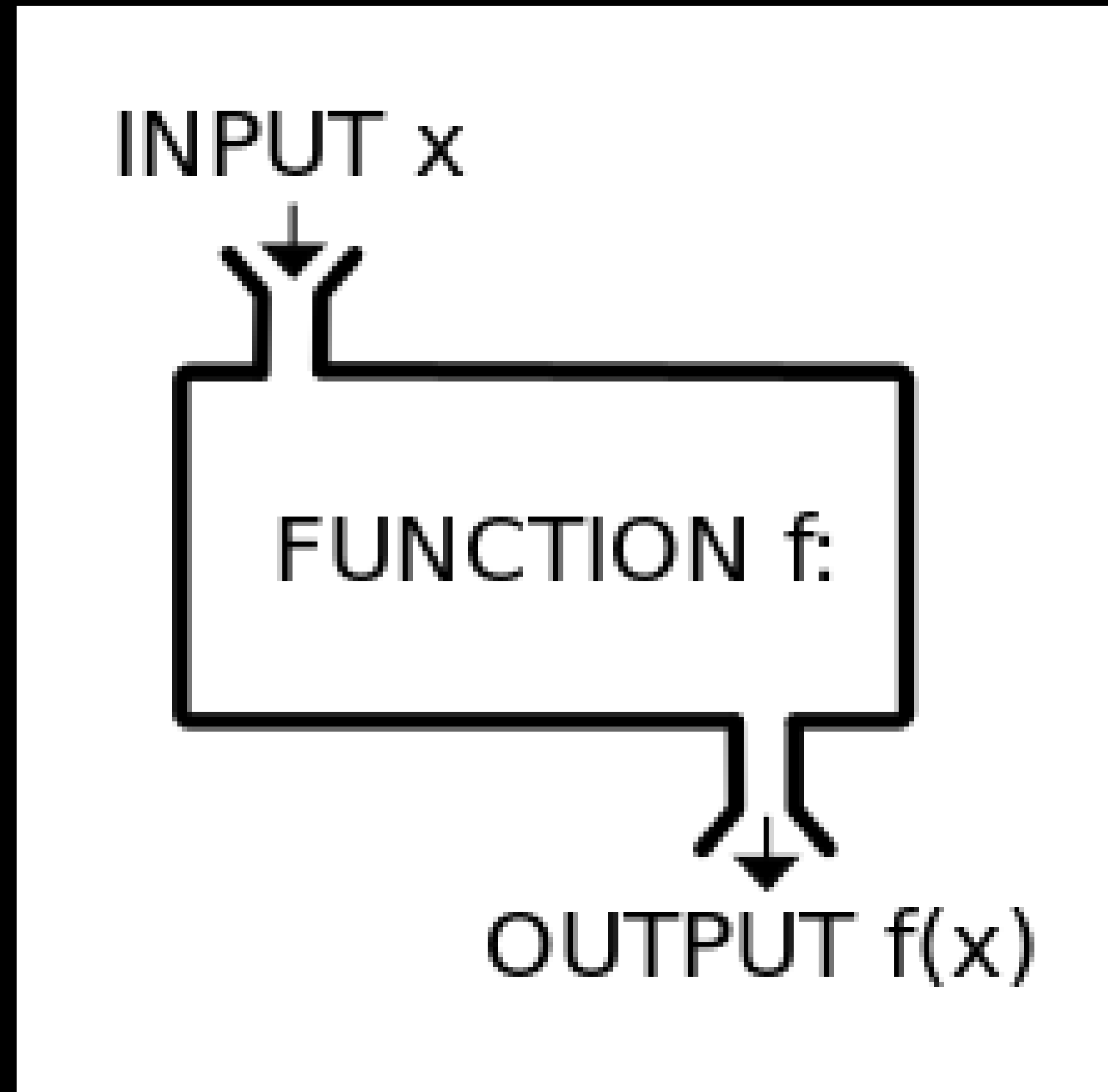
함수

이미 모두 다 쓰고 있습니다.

함수

네!?!?!?!!

함수



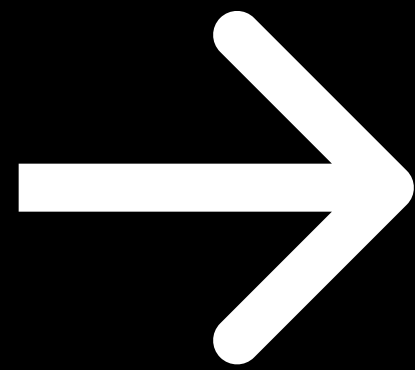
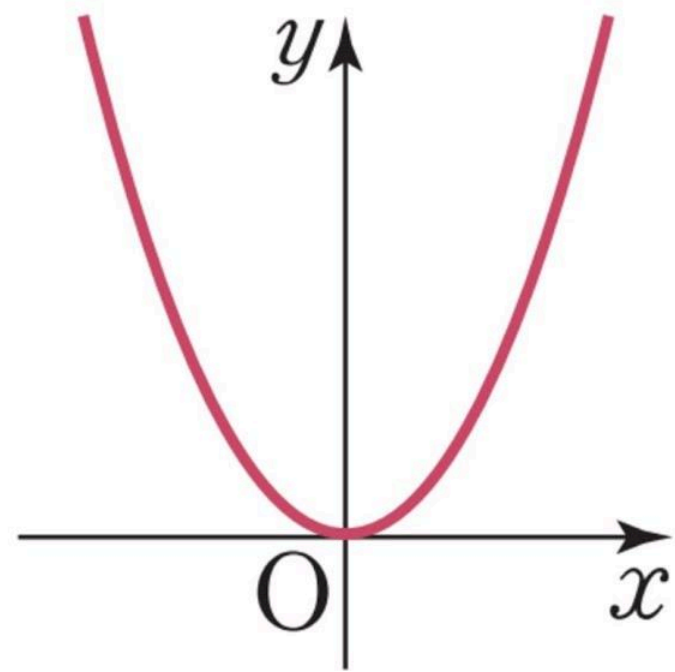
**함수는 입력과 출력이
존재한다.**

함수

이차함수

$$y = x^2$$

의 그래프



#파이썬에서의 이차함수

```
def 이차함수(x): #입력 x
```

```
    return x**2 #x의 제곱을 반환
```

```
print(이차함수(2))
```

애네들도 썩다 함수 였던거임



```
1 print()
```

```
2 input()
```

```
3 int(); float(); str()
```

```
4 range()
```


이런식으로 함수를 선언할 수 있습니다.



```
1 def 함수이름(입력값1, 입력값2):  
2     반환값 = 입력값1+입력값2  
3     return 반환값
```

구구단 함수, 실습 해봅시다!

입력으로 N을 받아 N단을 출력하는 함수를 만들어보자

$0 < N < 10$

입력이 잘못되었으면 '오류' 출력

도전과제

오늘의 도전과제

BOJ 1620

“나는야 포켓몬 마스터 이다솜”

이거 푼 사람 나와야지 집 갑니다ㅅㄱ

**배우지 않은 문법도 있을 수 있으니
집단지성으로 푸세요**

BOJ 1620

힌트1

'문자열'.isdemical()은 문자열이 숫자로 이루어졌을 때
True, 그렇지 않을 때 **False**를 반환

BOJ 1620

힌트2

일명 '빠른 입출력'을 사용해야합니다.


이건 모르겠으면 질문하세요.

BOJ 1620

힌트3

문자열의 좌우의 공백을 없애고 싶으면
`'문자열'.strip()`

도전과제



```
import sys
_input = sys.stdin.readline #_input()

n,m = map(int,input().split())

nameKey = {}
numKey = {}

for i in range(1,n+1):
    pkm = _input().rstrip()
    nameKey[pkm] = str(i)
    numKey[str(i)] = pkm

for i in range(m):
    ipt = _input().rstrip()
    if ipt.isdecimal():
        print(numKey[ipt])
    else:
        print(nameKey[ipt])
```