LESSON

# 람다(lambda)

# 람다(lambda) 함수



- **이름이 없는 일회용 함수이다.**
- **def를 사용하지 않고 함수를 한 줄로 간단하게 만들어 준다.**
- 함수 이름은 없고 **몸체만 있는 함수(익명 함수)**로 return을 사용하지 않는다.

**lambda 매개변수1, 매개변수2 : 리턴값**

# 람다(lambda) 함수



```
# 일반적인 함수 정의
def add(x,y):
    return x+y

print(add(10,20))
```

```
30
```

```
# lambda를 이용한 함수 정의
add = lambda x, y:x + y
print(add(10, 20))

# lambda를 이용한 함수 정의
print((lambda x, y:x+y)(10, 20))
```

```
30
```

# 도전!

[문제] 아래 일반적인 함수 정의 코드를 lambda 버전으로 만들어 보세요.

```
def check_odd_even(num):
    return "짝수" if num%2==0 else "홀수"

print(check_odd_even(5))
print(check_odd_even(6))
```

LESSON

# map()
# filter()

# map() 함수



- 파이썬 내장 함수.
- 데이터를 처리할 **함수와 iterable데이터 (문자열, 리스트, 튜플 등)을 입력받아** 처리하는 함수.

**map(함수, 반복 가능한 객체)**

내장 함수, 사용자 정의 함수,
**람다함수**, 메소드, 클래스 ...

# map() 함수



```
# 숫자를 입력받아서 최대값을 출력하기
nums=input('정수입력: ').split()
new_nums=[int(num) for num in nums]
print(max(new_nums))


# map()함수 적용
nums=map(int,input('정수 입력: ').split())
print(max(nums))
```

```
정수 입력: 1 2 3 4 5
5
```

# map() 함수



```
# 문자열의 길이를 리스트로 만들기
names=['python','c/c++','java','visual basic']
print(list(map(len,names)))

# 기존 리스트 요소에 10을 더한 값으로 변경
my_list=[1,2,3,4]
my_list=list(map(lambda num:num+10,my_list))
print(my_list)
```

```
[6, 5, 4, 12]
[11, 12, 13, 14]
```

# map() 함수



```python
# 홀수 만들기
nums = [1, 2, 3, 4]
def make_odd(num):
    return num if num % 2 == 1 else num + 1

is_odd = list(map(make_odd, nums))
print(is_odd)
```

```
[1, 3, 3, 5]
```

# filter() 함수



- 파이썬 내장 함수.
- 데이터에서 **지정한 조건에 맞는 요소만 추출하는** 함수.
- if문 사용없이 조건에 맞는 데이터만 내보낼 수 있다. 즉, 반환 값이 참(True)인 것만 추출해서 돌려준다.

**filter(함수, 반복 가능한 객체)**

내장 함수, 사용자 정의 함수,
**람다식**, 메소드, 클래스 ...

# filter() 함수



```
# 80이상인 점수들의 평균
scores=[23,45,66,2,78,44,99,100,56,87,93]
n=list(filter(lambda x:x>=80,scores))
print(f'80이상의 평균:{sum(n)/len(n)}')
```

```
80이상의 평균:94.75
```

# filter() 함수



```
# 5~10 사이 짝수값만 출력하기
def check(num):
    if num>=5 and num<=10 and num%2==0:
        return True
    else:
        return False

nums=[1,2,5,10,4,6,11]
print(list(filter(check,nums)))
```

[10, 6]