3. 함수와 람다

1. 함수

```
def coffee_machine(button) :
    print()
    print("#1 뜨거운 물 준비")
    print("#2 종이컵 준비")
    if button==1 :
        print("#3 보통 커피를 탄다")
    elif button == 2 :
        print("#3 설탕 커피를 탄다")
    elif button == 3 :
        print("#3 블랙 커피를 탄다")
    else :
        print("#3 커피 종류 없음")
    print("#4 물을 붓는다.")

#main 시작
coffee=int(input("커피 종류를 입력하세요(1:보통,2:설탕,3:블랙)"))
coffee_machine(coffee)
```

1. 함수

전역 변수,지역변수

```
def func1() :
    global gval #gval 변수는 전역변수로 선언된 변수로 사용함.
    a = 20   #지역변수  func1 함수에서만 사용되는 변수
    b = 1000 #지역변수. func1 함수에서만 사용되는 변수
    gval = 200 #전역변수 gval 값을 100에서 200으로 변경
    print("func1() 함수 호출함",gval,a,b) #200 20 1000. 전역변수,지역변수,지역변수값 출력
def func2() :
    print("func2() 함수에서 func1() 함수를 호출함")
    func1()
    print("전역 변수 gval 값=",gval,a) #전역변수값 출력
    #print(b) #오류

gval = 100  #전역변수 => 모든 함수에서 접근이 가능한 변수
a = 10      #전역변수
if __name__ == '__main__' : #프로그램의 시작. main함
    func2()
```

1. 함수

리턴값을 두개이상 반환 =>리스트로 반환

```
def multi(v1,v2):
    list = []
    res1 = v1+v2  #합
    res2 = v1-v2  #차
    list.append(res1)
    list.append(res2)
    return list  #list[0]:합계, list[1]:차

hap,sub=0,0
list = multi(100,200)
hap = list[0]
sub = list[1]
print("multi 함수의 리턴: %d, %d " % (hap, sub))
```

1. 함수

가변 매개변수 함

```
def multiParam(* p) :  #가변매개변수 : 매개변수의 갯수 지정이 안됨.
    result = 0
    for i in p :
        result += i
    return result

print("multi 함수의 리턴: %d, %d " % (hap, sub))
print("multiParam()=",multiParam())  #0
print("multiParam(10)=",multiParam(10)) #10
print("multiParam(10,20)=",multiParam(10,20))  #30
print("multiParam(10,20,30)=",multiParam(10,20,30)) #60
```

1. 함수

함수의 종료 없이 ,yield로 값 리턴

```
def genFun(num) :
    for i in range(10, num + 10) :
        yield i  #i변수의 값을 전달. 리턴하지 않음.
        print(i, "값 반환")


print(list(genFun(5)))

for data in genFun(5) :
    print("main에서 출력 : " , data)
```

2. 람다

람다식으로 함수 구현

```
def hap(num1,num2) :
    res = num1 + num2
    return res

print(hap(10,20))

hap1=lambda num1,num2:num1+num2  #람다식으로 작성된 함수
print(hap1(10,20))
```

2. 람다

람다 방식으로 곱셈 출력하기

```
mul=lambda num1,num2:num1*num2  #람다식으로 작성된 함수
print(mul(10,20)) #200
```

매개변수의 기본값 설정하기

```
hap2 = lambda num1=0,num2=1:num1+num2
print(hap2())  #1
print(hap2(100))  #num1=100, num2=1 앞자리의 매개변수로 설정됨.
print(hap2(100,200))
```

2. 람다

람다식을 이용한 리스트 처리

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

def add10(num):
    return num+10

for i in range(len(myList)) :
    myList[i] = add10(myList[i])
print(myList)

add=lambda num:num+10
for i in range(len(myList)) :
#     myList[i] = add(myList[i])
    myList[i] = (lambda num:num+10)(myList[i])
print(myList)
```