

문제: 함수과제 1

함수 과제 1

```
In [95]: def check_prime(num):  
        for div in range(2, num):  
            if num % div == 0:  
                return False  
        return True  
  
        def main():  
            a = 13  
            b = 15  
  
            if check_prime(a):  
                print(str(a)+'는 소수입니다.')  
            else:  
                print(str(a)+'는 소수가 아닙니다.')  
  
            if check_prime(b):  
                print(str(b)+'는 소수입니다.')  
            else:  
                print(str(b)+'는 소수가 아닙니다.')  
  
        main()
```

13는 소수입니다.
15는 소수가 아닙니다.

문제: 함수과제 2

함수 과제 2

```
In [160]: def add_comma(val):
str_val = list(str(val))
str_rev = str_val[::-1]
#print(str_rev)
mod_rev = ""
for i in range(0, len(str_rev)):
    mod_rev += str_rev[i]
    if( i%3 == 2):
        mod_rev+=", "

    if mod_rev[-1] == ", ":
        mod_rev = mod_rev[:len(mod_rev)-1]
#print(mod_rev)
return mod_rev[::-1]

def main():
    comma_added_1234 = add_comma(1234)
    comma_added_12345678 = add_comma(12345678)
    comma_added_12 = add_comma(12)

    print(comma_added_1234)
    print(comma_added_12345678)
    print(comma_added_12)

main()

1,234
12,345,678
12
```

문제: 함수 과제 3

함수 과제 3

```
In [143]: def tokenize(trg, N=1):
each_list = trg.split()
if N==1:
    return each_list
else:
    length = len(each_list)
    lst = []
    #print(length)
    for i in range(0, length-N+1):
        tmp = ""
        for j in range(0, N):
            tmp += (each_list[i+j] + " ")
        tmp = tmp[:len(tmp)-1]
        lst.append(tmp)
        #print(tmp)
    return lst

def main():
    a = "There was a farmer who had a dog ."
    print(tokenize(a))
    print(tokenize(a, 2))

main()
```

```
['There', 'was', 'a', 'farmer', 'who', 'had', 'a', 'dog', '.']
```

```
['There was', 'was a', 'a farmer', 'farmer who', 'who had', 'had a', 'a dog', 'dog .']
```

문제: 함수 과제 4

함수 과제 4

```
In [157]: def mean_and_var(*val):
            avg_x = 0; b_x=0
            avg_y = 0; b_y=0
            #평균
            for each in val:
                avg_x += each[0]
                avg_y += each[1]
            avg_x /= len(val)
            avg_y /= len(val)
            #분산
            for each in val:
                b_x += (each[0]-avg_x)**2
                b_y += (each[1]-avg_y)**2
            b_x /= len(val)
            b_y /= len(val)

            return ([avg_x, avg_y], [b_x, b_y])

v1 = (0,1)
v2 = (0.5, 0.5)
v3 = (1,0)
m, var = mean_and_var(v1,v2,v3)
print(m, var)

[0.5, 0.5] [0.16666666666666666, 0.16666666666666666]
```

In []: