

PYTHON TUTORING #4

School of Computing, KAIST & 대덕고등학교 빛나리

INTRO

- ① 저번 시간 REVIEW
- ② 함수의 선언과 사용
- ③ 프로그래밍 예제

string REVIEW

string이란 문자들로 구성된 문자열

따옴표로 둘러싸여 있으면 string

string들의 연산은 숫자와 다르게 정의

len(string)은 주어진 string의
길이를 반환하는 기본 기능

예시

```
str1 = "ABC"
str2 = "123"
str3 = "I love Python"
str4 = "내 이름은?"

print(str1 + str2)
print(str3 * 2)
print(str1 == str2)
print(len(str3))
```

string REVIEW

`string[n]`은 $n+1$ 번째 문자를 의미한다.

`str3[0]`: "I"

`str3[1]`: " "

`str3[3]`: "o"

...

`str3[12]`: "\n"

예시

```
str3 = "I love Python"  
length = len(str3)  
  
print(str3[0])  
print(str3[1])  
  
print(str3[3])  
  
print(str3[12])  
print(str3[length - 1])
```

string REVIEW

`string[a:b]`는 `a+1`번째 문자부터 `b`번째 문자까지를 의미한다.

```
str3[2:10] = "ove Pyt"
```

```
str3[:9] = "I love Py"
```

```
str3[4:] = "ve Python"
```

```
str3[:] = "I love Python"
```

예시

```
str3 = "I love Python"
```

```
print(str3[2:10])
```

```
print(str3[:9])
```

```
print(str3[4:])
```

```
print(str3[:])
```

string REVIEW

for 변수이름 **in** "string":

혹은

```
for i in range(len("string")):  
    "string"[i]
```

예시

```
str3 = "I love Python"
```

```
for s in str3:  
    print(s)
```

```
for i in range(len(str3)):  
    print(i)  
    print(str3[i])
```

string REVIEW

`count`는 `string`에 괄호 안의 요소가 등장한 횟수를 반환한다.

`find`는 괄호 안의 요소가 `string`의 몇 번째에 처음으로 등장하는지 반환한다.

예시

```
str3 = "I love Python"

c = str3.count("o")
print(c)

i = str3.find("P")
print(i)
print(str3[i])
```

예제

`input` 함수를 이용해서 문장을 받은 뒤,

한 문장에 등장하는 단어의 개수를 세는 프로그램을 작성하시오.

list REVIEW

list는 하나 이상의 값을 가지는 자료형

대괄호로 감싸져 있으면 list

list 안의 요소는 숫자, string, list
관계 없이 전부 가능

list 사이의 연산도 따로 적용

예시

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

```
empty = []
```

```
word = ["even", "odd"]
```

```
all = [1, 2, "number"]
```

```
print(odd + word)
```

```
print(word * 2)
```

list REVIEW

`list[n]`은 $n+1$ 번째 요소를 의미한다.

`all[0]: 1`

`all[1]: 2`

`all[2]: "number"`

예시

```
all = [1, 2, "number"]
```

```
print(all[0])
```

```
print(all[1])
```

```
print(all[2])
```

```
print(all[len(all)-1])
```

list REVIEW

`list[a:b]`은 $a+1$ 번째 요소부터 b 번째 요소까지를 의미한다.

`all[0]: 1`

`all[1]: 2`

`all[2]: "number"`

예시

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

```
print(all[0])
```

```
print(all[1])
```

```
print(all[2])
```

예제

`input` 함수를 이용하여 문장을 입력 받은 뒤,
문장 중에 중복되는 단어가 있는지 검사하여라.

Hint: 각 단어를 하나의 `list` 안에 넣고,
`i != j and list[i] = list[j]`인 `i, j`를 찾는다.

함수의 선언

```
def 함수이름(인자1, 인자2, ...):  
    수행할 문장1  
    수행할 문장2  
    ...  
    return 결과값
```

함수 내부의 문장들은 반드시 들여쓰기!

예시

```
def squareadd(a, b):  
    c = a * a  
    return c + b
```

함수의 사용

이름 붙힌 함수를 인자와 함께 실행하고

그 결과값을 다른 변수에 저장하거나

다른 함수의 인자로 사용할 수 있다.

예시

```
def squareadd(a, b):  
    c = a * a  
    return c + b  
  
a = squareadd(3, 4)  
print(a)  
print(squareadd(4, 5))
```

함수의 선언

함수는 반드시 첫 번째 **return**만 고려

두 가지 이상의 값을 반환하고 싶다면
`return a, b`를 사용한다.

결과값을 받아올 때도 서로 다른 변수에
각각 대입한다.

예시

```
def squareadd(a, b):  
    c = a * a  
    return c + b  
    return c * b
```

```
def squareadd(a, b):  
    c = a * a  
    return c + b, c * b
```

```
x, y = squareadd(3, 4)
```

함수의 사용

함수 내부에서 print를 하면
함수를 호출할 때마다 print를 같이 실행

다음의 실행 결과를 예측하면?

A -> B -> C -> A -> B

예시

```
def message():  
    print("A")  
    print("B")
```

```
message()  
print("C")  
message()
```


예제

숫자 1개를 받아서

그 숫자의 팩토리얼을 반환하는 함수를 정의하여라.

$$n! = n * n-1 * n-2 * \dots * 3 * 2 * 1$$

예제

1. 숫자 1개를 받아서
그 숫자가 완전수인지 아닌지 반환하는 함수를 정의하여라.
2. 작은 순서대로 3개 이상의 완전수를 찾아라.

완전수: 자기 자신을 제외한 약수들의 합이 자기 자신인 수

Ex) $6 = 3 + 2 + 1$, $28 = 14 + 7 + 4 + 2 + 1$

예제

string 2개를 입력 받아서,

각 string을 구성하는 문자들이 동일한지
확인하는 프로그램을 작성하시오.

Ex) checker(“stressed”, “desserts”)

예제

숫자로 된 `list` 1개를 입력 받아서,

짝수만 포함되어 있는 `list`를 반환하시오.

Hint: 새로운 빈 `list`를 선언한 후, `append` 함수를 사용