

PYTHON TUTORING #2

School of Computing, KAIST & 대덕고등학교 빛나리

INTRO

- ① 저번 시간 REVIEW
- ② Conditional Statement
- ③ Loop Control
- ④ 예제 프로그래밍

Python REVIEW

덧셈, 뺄셈, 곱셈, 나눗셈

곱셈과 나눗셈은 각각 * 와 / 로 약속

//는 나눗기의 몫

%는 나눗기의 나머지

예시

```
>>> 1 + 2
```

```
>>> 3 - 6
```

```
>>> 2 * 7
```

```
>>> 7 / 3
```

```
>>> 7 // 3
```

```
>>> 7 % 3
```

Python REVIEW

변수는 값을 저장하는 공간

= 기호를 통해서 변수에 값을 저장!

변수 이름 = 변수에 저장할 값

후에 같은 이름의 변수가 사용되면,
대입된 값으로 명령을 수행하게 된다.

예시

```
>>> a = 1 + 2
```

```
>>> b = 3 - 6
```

```
>>> a
```

```
>>> b
```

```
>>> a + b
```

```
>>> a - b
```

```
>>> c = a * b
```

```
>>> a
```

Python REVIEW

변수는 숫자가 아닌 것도 대입 가능

문자열, 집합, 함수 등

문자열에 대해서도 일부 연산 적용 가능

문자열을 출력하기 위해서는 `print` 문 사용

예시

```
>>> var1 = "Hello"
>>> var2 = "KAIST"

>>> print(var1)
>>> print(var2)

>>> var = var1 + var2
>>> print(var)
```

Python REVIEW

사용자로부터 입력을 받기 위해서는

input 문 사용

input 문은 괄호 안에 있는 값을 출력하고,
사용자가 입력한 값을 반환한다.

예시

```
>>> a = input("ABC")
```

```
ABC
```

```
>? 3
```

```
>>> b = input("CBA")
```

```
CBA
```

```
>? "hello"
```

```
>>> print(a)
```

```
>>> print(b)
```

예제

`input` 함수를 이용하여 원의 반지름을 입력 받아서,
원의 둘레와 넓이를 출력하는 프로그램을 작성하시오.

단, $\pi \cong 3.14$

Conditional Statement

현재 조건을 판단하여

특정 조건을 만족할 때만 실행되는 코드

예시

```
grade = 0
score = input()

if score >= 60:
    print("PASS")
    grade = grade + 1
else:
    print("FAIL")

print(grade)
```


Conditional Statement

if 다음에는 어떤 조건 하에서 실행할지 조건문을 이용하여 표현해줘야 한다.

비교연산자	설명
$x < y$	x가 y보다 작다
$x > y$	x가 y보다 크다
$x == y$	x와 y가 같다
$x != y$	x와 y가 같지 않다
$x \geq y$	x가 y보다 크거나 같다
$x \leq y$	x가 y보다 작거나 같다

연산자	설명
$x \text{ or } y$	x와 y 둘중에 하나만 참이면 참이다
$x \text{ and } y$	x와 y 모두 참이어야 참이다
$\text{not } x$	x가 거짓이면 참이다

예시

```
if 조건문:
    수행할 문장1
    수행할 문장2
    수행할 문장3

if x > y or x < y:
    print("not same")
    x = y
    print("now same!")
```

Conditional Statement

if + 조건문 다음에는 반드시 콜론 :

수행할 문장들은 반드시 **들여쓰기**

else: 뒤에는
조건문을 만족하지 않을 경우의 코드

예시

```
if 조건문:
    ____수행할 문장1
    ____수행할 문장2

else:
    ____수행할 문장3
    ____수행할 문장4
```

Conditional Statement

셋 이상의 분기문을 만들고 싶다면,

예시

```
if 조건문1:  
    수행할 문장1  
else:  
    if 조건문2:  
        수행할 문장2  
    else:  
        수행할 문장3
```

Conditional Statement

셋 이상의 분기문을 만들고 싶다면,
elif를 이용하여 코드를 줄일 수 있다.

예시

```
if 조건문1:  
    수행할 문장1  
elif 조건문2:  
    수행할 문장2  
else:  
    수행할 문장3
```

예제

input 함수를 이용하여 성적을 입력 받아서,

성적에 따른 "ABCD F"를 출력하는 프로그램을 작성하시오.

A : 90 ~ 100

B : 80 ~ 89

C : 70 ~ 79

D : 60 ~ 69

F : 0 ~ 59

Loop Control

같은 코드를 여러 번 반복시키고 싶을 때,
for, while을 이용한다.

n번 반복하고 싶다면

for 변수이름 in range(n):

예시

```
print("hello")  
print("hello")  
print("hello")  
print("hello")
```

```
for i in range(4):  
    print("hello")
```

Loop Control

매 코드 실행 시마다

i 의 값이 0부터 9까지 변화

if와 마찬가지로 콜론 :, **들여쓰기** 중요!

예시

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

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

Loop Control

while은 조건문과 함께 사용

for보다 덜 사용하지만,
복잡한 프로그램을 작성할 때 자주 사용

마찬가지로 콜론 :, **들여쓰기** 중요!

예시

```
변수 초기화
while 조건문:
    반복할 문장1

i = 0
while i < 10:
    print(i)
    i = i + 1
```


예제

0 ~ 1000 이하의 수 중,

7로 나누면 나머지가 3

11로 나누면 나머지가 5

13으로 나누면 나머지가 9

가 되는 수를 찾아라!

예제

input 함수를 이용하여 입력 받은 숫자가,

소수인지 판별하여 결과를 출력하는 프로그램을 작성하시오.

Hint: 숫자 n 이 소수인지 확인하려면,

n 을 2부터 루트 n 까지의 숫자로 나눴을 때 나누어 떨어지면 안됨!