



Python Tutoring #2

School of Computing, KAIST
& 대덕고등학교 빛나리



목차

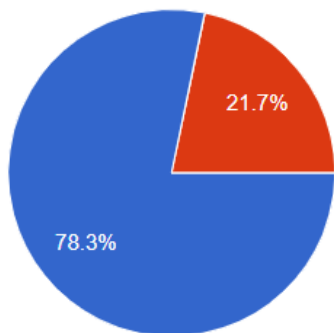
- Review
- Conditional Statement (조건문) : if
- Loop Control (반복문)
 - for, while
- 문제 풀이



설문조사 결과

프로그래밍을 배운 적이 있나요?

응답 23개



동아리와 학교에서 프로그래밍을 배웠습니다.

작년에 이 동아리에서 배웠다

정식적인 첫 경험은 빛나리가 처음이었습니다.

방학때 잠시동안 학원을 다녔습니다.

동아리시간, 정보시간

파이썬, 스크래치로 간단한 프로그래밍들을 해 봄

1학년 때 파이썬을 사용하여 코딩을 배우고 아두이노를 활용하여 코딩도 해봄

혼자 독학으로 해본거여서 간단한 게임정도만 만들어보았습니다.

기초능력과 간단한 알고리즘 등에 대해 학습하였고 소규모 프로젝트를 영재원에서 진행한 적 있습니다.



참고 자료

- 작년 튜터링 자료

<https://github.com/ironore15/Python-Tutoring>

- 점프 투 파이썬

<https://wikidocs.net/book/1>

- 백준

<https://www.acmicpc.net/>



참고 자료

- Leet Code
<https://leetcode.com/>
- Codeforce
<https://codeforces.com/>



Review



Review: Python Shell과 File의 차이점

- Python Shell
- Python File

Review: 파이썬의 기본 연산

- 기본적인 사칙연산

$+$, $-$, $*$, $/$

- 나머지 연산자 : $\%$

e.g.

$8 \% 3 = 2$

- 몫 연산자 : $//$

e.g.

$9 // 4 = 2$

예시

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

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

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

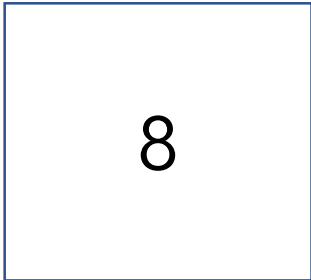
```
>>> 7 / 3
```

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

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


Review: 변수에 값을 저장하기

- x (변수 이름) = 8 (변수에 저장할 값)

X : 

- 후에 변수 이름만으로 안의 값을 바꾸는 연산을 할 수 있다.

예시

```
>>> a = 1 + 2
>>> b = 3 - 6
>>> a
>>> b
>>> a + b
>>> a - b

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

Review: 문자열과 print()

- 숫자가 아닌 값도 대입 가능
- 문자열, 집합, 함수 등등
- 문자열에 대해서도 일부 연산 가능
+, *
- 문자열을 출력하기 위해 print 사용

예시

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

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

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

Review: input()

- 사용자로부터 값을 입력 받을 수 있다.
- Console보다는 File에서 활용하기 좋다.
- Input 문은 괄호 안의 값을 출력하고 사용자가 입력한 값을 반환한다.

예시

```
>>> a = input("ABC")
ABC
>? 3

>>> b = input("CBA")
CBA
>? "hello"

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



예제 1

- `input()`을 이용하여 원의 반지름을 입력 받아서
원의 둘레를 출력하는 프로그램을 작성하라.
단, π 의 근삿값 3.14를 활용하라.
- 힌트 : 원의 둘레 = $2 * \pi * \text{원의 반지름}$



Conditional Statement



Conditional Statement (조건문)

- if에 주어진 조건을 보고
특정 조건을 만족할 때만 실행되는 코드
- Indentation (들여 쓰기)
파이썬 vs. c 언어

```
score = input()

if score >= 60 :
    print("Pass")
else :
    print("Fail")
```



if문 작성하기

- if + 조건문 + : (콜론)
- 조건문 안에서 수행할 문장에는
들여쓰기할 것
- else: 조건문을 만족하지 않는
경우 수행할 문장

```
score = input()

if score >= 60 :
    #조건문을 만족
    print("Pass")
else :
    #조건문을 불만족
    print("Fail")
```



2가지 이상의 상황에 대한 if문 작성하기

- if / else를 if / else 안에 추가할 것

```
score = input()

if score >= 60 :
    if score >= 90:
        print("Awesome")
    else:
        print("Pass")
else :
    print("Fail")
```




2가지 이상의 상황에 대한 if문 작성하기

- elif를 활용할 것

```
grade = input()

if score >= 90 :
    print("Awesome")
elif score >= 60:
    print("Pass")
else :
    print("Fail")
```



조건문과 관련된 연산

- 비교 연산자 : `<`, `>`, `==`, `!=`, `>=`, `<=`
- 논리 연산자 : `and`, `or`, `not`

e.g.

```
if score1 >= 60 and score2 >= 60 :  
    print( "All Pass" )
```



예제 2

- input 함수를 활용하여 성적을 입력 받자.
- 입력 받은 성적에 따라서 아래의 기준으로 "ABCD F"를 출력하자.

A: 90 ~ 100

B: 80 ~ 89

C: 70 ~ 79

D: 60 ~ 69

F: 0 ~ 59



Loop Control



Loop Control

- 같은 코드를 여러 번 반복하여
수행하고 싶을 때: for, while

e.g.

n번 반복하고 싶다면,

for 변수 이름 in range(n):
 수행할 문장

```
print("Hello World!")  
print("Hello World!")  
print("Hello World!")  
print("Hello World!")  
print("Hello World!")
```

```
for i in range(5) :  
    print("Hello World!")
```



For Loop

- 매번 Loop를 돌 때마다
i의 값이 0 -> n-1까지 순차적으로 변화
- : (콜론)과 들여쓰기에 유의할 것

```
for i in range(5) :  
    print( i )
```

```
for i in range(5):  
    for j in range(5):  
        print( i, j )
```

While Loop

- while + 조건문 + : (콜론)
- : (콜론)과 들여쓰기에 유의할 것
- Loop를 빠져나갈 수 있는지 확인 할 것

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



예제 3

- 0부터 1000까지의 수 중 아래의 조건을 만족하는 수를

찾아서 print 해보자.

7로 나눈 나머지가 3

11로 나눈 나머지가 5

13으로 나눈 나머지가 9

- 힌트 : for Loop를 활용할 것

예제 4

- input으로 양의 정수를 하나 입력 받자.
- 입력 받은 정수가 소수인지 판별하여 결과를 print해보자.
- 힌트 :
n이 소수라면,
1부터 n-1으로 n을 나누어도 나머지가 0이 아니다.



추가 예제

- <https://www.acmicpc.net/problem/2557>
print 함수의 활용
- <https://www.acmicpc.net/problem/1000>
input, print 함수의 활용
- <https://www.acmicpc.net/problem/8958>
input, for, print 활용