



# Python Tutoring #4

School of Computing, KAIST  
& 대덕고등학교 빛나리



# 목차

- 1학기 Review

  - Python Shell / File의 활용

  - 변수 정의하기

  - 조건문 if / 반복문 for, while

  - Sequence 자료형

- 참고하면 좋을 자료

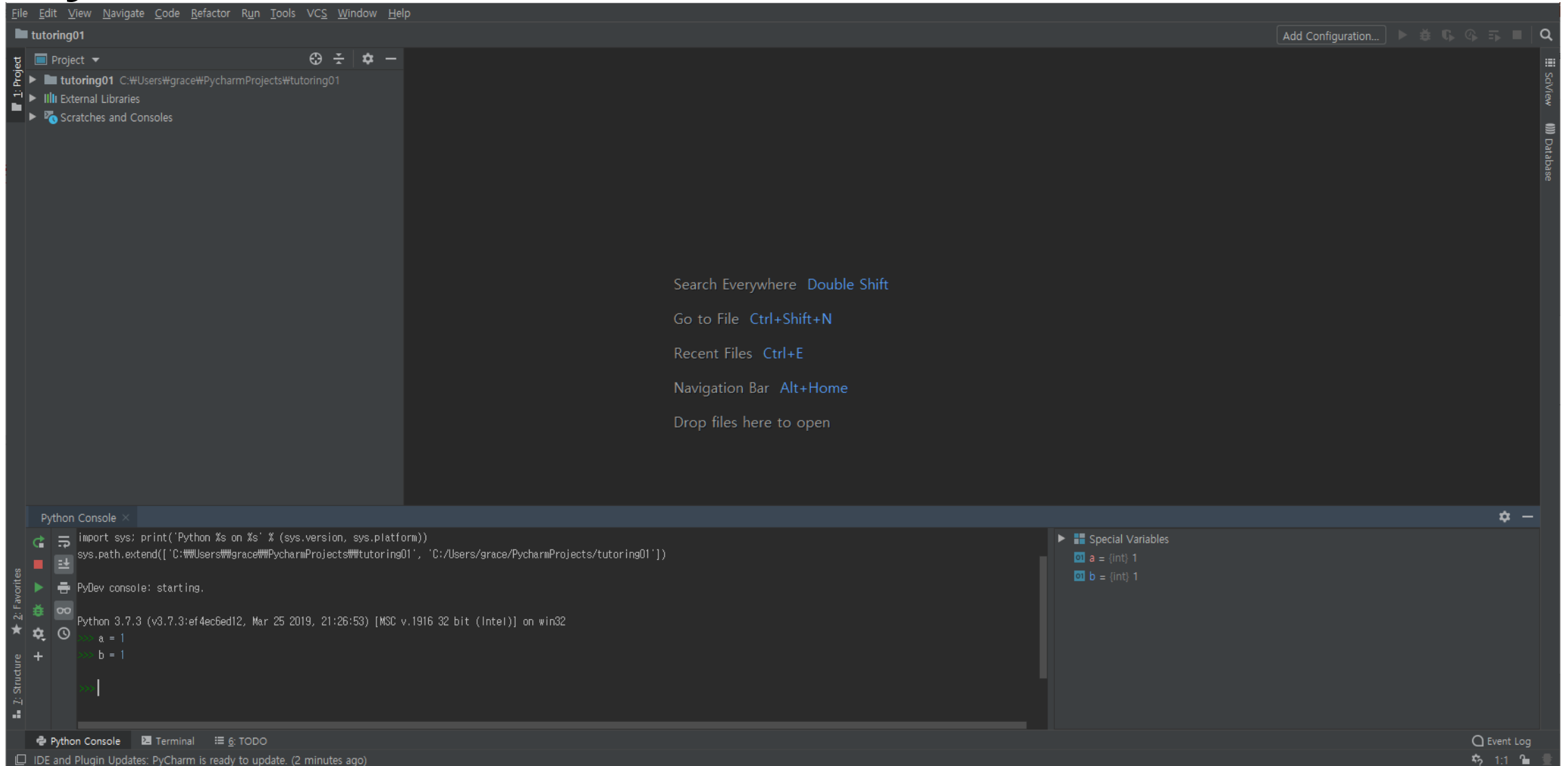
  - Python 설치하기 / 그룹 프로젝트를 위한 공유 방법



# Review



# Python Shell / File의 활용





# Console에서 간단한 연산 해보기

- 기본적인 사칙연산

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

- 나머지 연산자 :  $\%$

e.g.

$8 \% 3 = 2$

- 몫 연산자 :  $//$

e.g.

$9 // 4 = 2$

예시

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

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

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

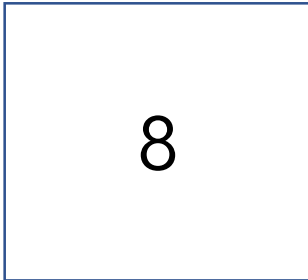
```
>>> 7 / 3
```

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

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

# 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
```

# 사용자로부터 입력 받기

- 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)
```



# 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")
```





# 2가지 이상의 상황에 대한 if문 작성하기

- elif를 활용할 것

```
grade = input()

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



# 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
```



# Conditional Statement (조건문)

- if에 주어진 조건을 보고

특정 조건을 만족할 때만 실행되는 코드

- Indentation (들여 쓰기)  
파이썬 vs. c 언어

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

# 예제 1

- 이차방정식의 해가 몇 개인지 판별하는 프로그램을 짜보자.

Step 1 : A, B, C의 정수를 입력 받자

Step 2 : A가 0이면 "2차방정식이 아니다."라고 print한다.

Step 3 : 판별식의 값에 따라 해가 몇 개인지 판별하자.

힌트: 판별식은  $D = B * B - 4 * A * C$  와 같다.





# 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
```



# 시퀀스(Sequence) 자료형

e.g.

```
string1 = "ABCDEF"
```

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>
0	1	2	3	4	5

Index(순서)로 접근할 수 있다. (Index는 0부터 시작)

Indexing 예시 : `print(string1[1])` >> B



# Slicing(자르기)

e.g.

```
string1 = "ABCDEF"
```

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>
0	1	2	3	4	5

Slicing(자르기)를 하면 원래 자료형의 일부가 나온다.

Slicing 예시 : `print(string1[1:3])` >> BC (index 1부터 3이전까지)



# 시퀀스형 자료에 대한 Loop Control

- 형식 :

```
for s in sequence1:  
    명령문1
```

```
for i in range(len(sequence1)):  
    명령문2
```

```
string1 = "I LOVE PYTHON"
```

```
for s in string1:  
    print(s)
```

```
for i in range(len(string1)):  
    print(string1[i])
```

# String의 연산

연산	결과	예시
string1 + string2	두 개의 문자열을 이어준 것	"A" + "B" >> "AB"
string * n	하나의 문자열을 n번 반복한 것	"A" * 3 >> "AAA"
string1 == string2	두 개의 문자열이 같은가	"A" == "A" >> True
len(string)	문자열의 길이 반환	len("ABC") >> 3



# List의 연산

연산	결과	예시
<code>list1 + list2</code>	두 개의 list를 이어준 것	<code>[1, 2] + [3] &gt;&gt; [1, 2, 3]</code>
<code>list1 * n</code>	하나의 list를 n번 반복한 것	<code>[1] * 3 &gt;&gt; [1, 1, 1]</code>
<code>list1 == list2</code>	두 개의 list의 각각의 원소가 같은가	<code>[1, 2] == [1, 2] &gt;&gt; True</code>
<code>len(list1)</code>	list의 길이 반환	<code>len([1, 2, 3]) &gt;&gt; 3</code>





# Dynamic programming: 피보나치 수

- N 번째 피보나치 수를 구하는 방법?

```
N = int(input())
numbers = [0] * (N+2) # N개의 수를 저장할 공간 + 여유공간
numbers[0] = 1
numbers[1] = 1
for i in range(2, len(numbers)):
    numbers[i] = numbers[i-1] + numbers[i-2]
print(numbers[N-1])
```

# 예제: 구구단 출력 프로그램

- Input : n

>> n단의 구구단을 리스트에 저장하고 출력

1. 사용자 함수 input()을 사용하여 n을 입력 받는다.
2. 입력 받은 n단의 구구단을 리스트에 저장한다.  
2 >> 0 | 2 | 4 | 6 | ... | 18
3. 저장한 리스트를 출력한다.

힌트: input(), print(), for loop를 활용하세요.



# Python 설치하기



# Python 설치 여부 확인하기

1. (Windows 기준) 명령 프롬프트 켜기
2. `python --version`

명령창에 위와 같이 입력하면 아래와 비슷하게 출력됨

```
>>Python 3.7.3
```



# Python 설치하기

- Python 설치하기 (3.7.? 설치를 권장)

<https://wikidocs.net/20339>

- IDLE 사용하기

<https://wikidocs.net/20341>

- Pycharm 설치하기 : 권장 (가상환경 구성이 쉽다.)

<https://wikidocs.net/20343>



# 설치 없이 Python 사용하기

- 구글 Colab 사용하기

<https://theorydb.github.io/dev/2019/08/23/dev-ml-colab/>

머신 러닝에 관심이 많은 학생들에게 추천

.py : Python 파일 형식

.ipynb : 쥬피터 노트북 형식



# 그룹 프로젝트를 위한 공유 방법



# Github.com

- <https://github.com/>
- Github.io 블로그 만들기  
<https://zoomkoding.github.io/gitblog/2019/08/15/git-blog-1.html>
- Repository 만들고 공유하기  
git 명령어로 만들기 : <https://smallcandy.tistory.com/12>  
github 홈페이지에서 만들기 : <https://coding-factory.tistory.com/244>





# Google Colab

- Jupyter Notebook
  - Markdown : HTML과 비슷하게 문서 작성 가능한 언어
  - Shell 명령어 : linux 명령어 (e.g. wget)를 !와 함께 사용
  - Python : python shell처럼 사용 가능
- GPU 무료 액세스
- 데이터 분석, 머신러닝 관련 라이브러리 제공



감사합니다 :)