

PYTHON TUTORING #1

School of Computing, KAIST & 대덕고등학교 빛나리

INTRO

- ① 강사진 소개
- ② 프로그래밍이란?
- ③ Python이란?
- ④ PyCharm을 이용한 python 맛보기

강사진 소개

카이스트!

이준수 : 전산학부

서동평 : 전산학부

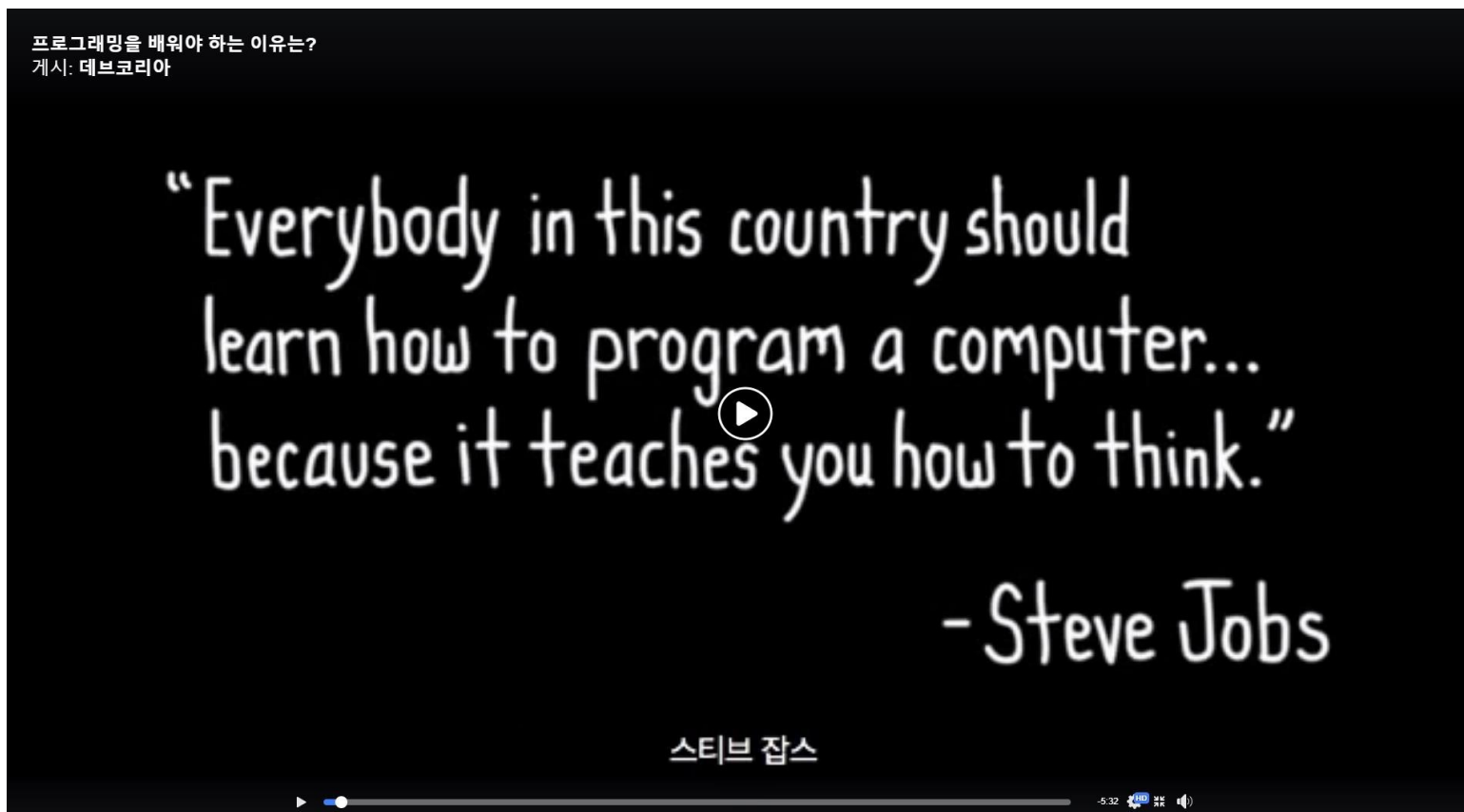
안승현 : 전기 및 전자공학부

설문조사

앞으로의 수업 방향성을 위한 설문조사!

<https://tinyurl.com/lightnari>

프로그래밍이란?



프로그래밍이란?

문제들을 해결하기 위한 계획(ex. 파일 찾기)

컴퓨터는 몇 억 번의 단순계산을 1초만에 할 수 있음



BUT 컴퓨터는 단순 작업만 가능

프로그래밍

= 실제 문제 해결 과정을
컴퓨터가 처리할 수 있는
수준으로 나눠서 정리한 것

프로그래밍 언어란?

프로그램을 작성하기 위해 필요한 특정 형식, 도구



Python을 사용하는 이유

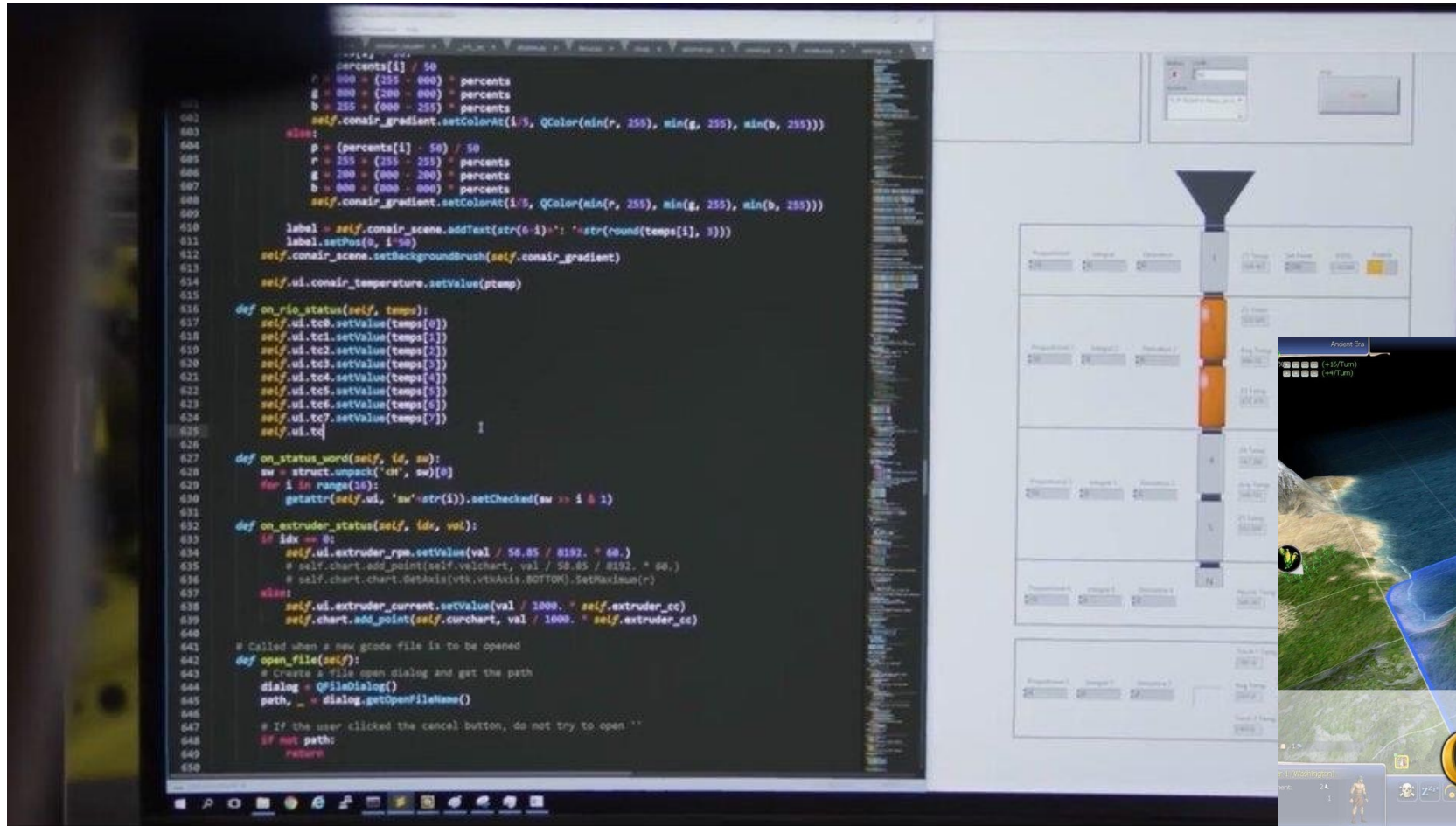
① 쉽고 간결한 문법

ex) `if 4 in [1,2,3,4]: print("4가 있습니다")`

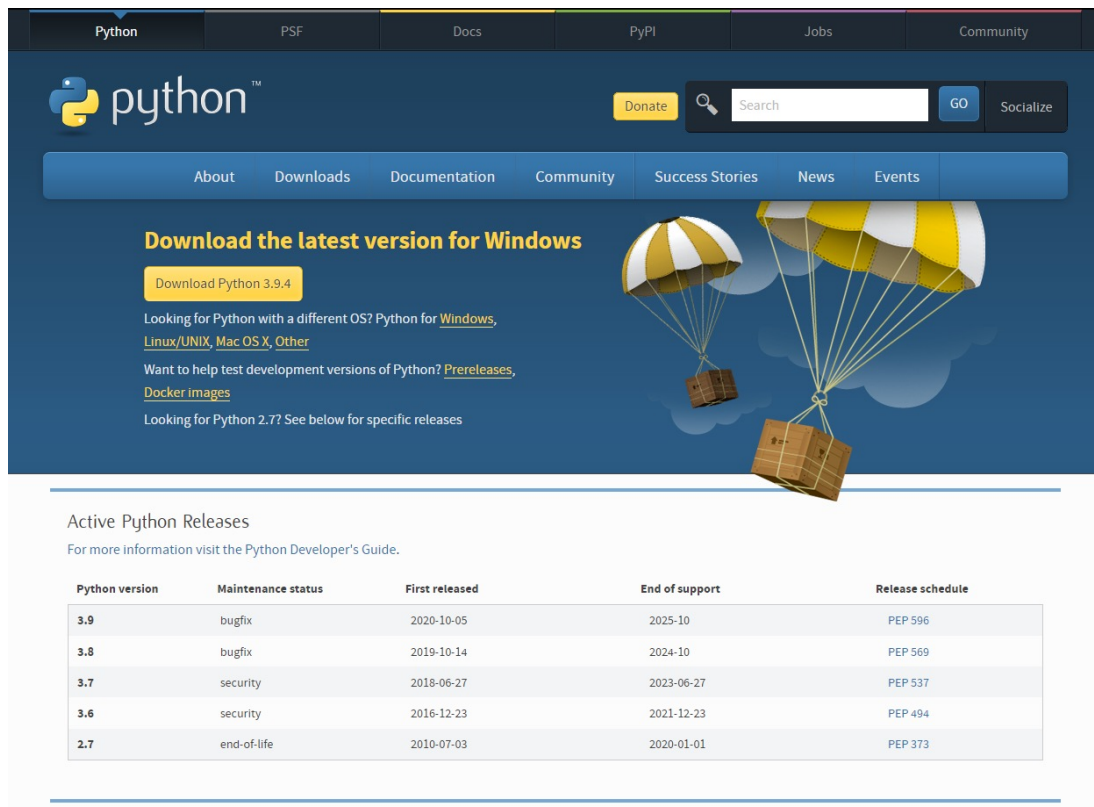
② 개발자를 위한 다양한 도구들(찾아서 가져다쓰기)

③ ***"Life is short, You need python."***

Python을 사용하는 이유



Python 설치



The screenshot shows the Python.org homepage. The navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. Below the navigation bar, there's a search bar and a 'Donate' button. The main content area features a large banner for downloading the latest version for Windows, with a 'Download Python 3.9.4' button. Below the banner, there are links for other operating systems and development versions. At the bottom, there's a table titled 'Active Python Releases'.

| Python version | Maintenance status | First released | End of support | Release schedule |
|----------------|--------------------|----------------|----------------|------------------|
| 3.9 | bugfix | 2020-10-05 | 2025-10 | PEP 596 |
| 3.8 | bugfix | 2019-10-14 | 2024-10 | PEP 569 |
| 3.7 | security | 2018-06-27 | 2023-06-27 | PEP 537 |
| 3.6 | security | 2016-12-23 | 2021-12-23 | PEP 494 |
| 2.7 | end-of-life | 2010-07-03 | 2020-01-01 | PEP 373 |

python.org -> Downloads 탭

```
Last login: Thu Apr  7 17:43:07 on ttys003
junsoolee ~ python3
Python 3.9.10 (main, Jan 15 2022, 11:40:36)
[Clang 13.0.0 (clang-1300.0.29.3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 1 + 1
2
>>> 5 * 8
40
>>> a = 2
>>> a
2
>>> 
```


Python shell

Python IDE(통합 개발 환경)

- 수 천 줄의 코드를 Python Shell로 작성하는 것은 비효율적임! (재활용 등)
- IDE: 더 쉽게, 더 빠르게 개발할 수 있도록 도와줌(자동 완성 등)
- 코딩, 디버그, 컴파일, 배포 등을 한 프로그램 안에서 모두 처리할 수 있음

-> Pycharm


Pycharm 설치


개발자 도구 팀 도구 학습 도구 솔루션 지원

교육

교육 도구 다운로드

[Java](#)
[Kotlin](#)
[Python](#)
[Scala](#)
[JavaScript](#)
[C/C++](#)
[Rust](#)
[Go](#)
[PHP](#)



버전: 2021.3.4
 빌드: 213.7172.45
 2022년 3월 31일
[릴리스 노트](#)

[시스템 요구 사항](#)
[설치 안내](#)
[기타 버전](#)

PyCharm Edu는 무료로 사용할 수 있습니다.

IDE 및 번들로 포함된 대부분의 플러그인은 오픈 소스를 기반으로 빌드되었으며 Apache 2 라이선스에 따라 제공됩니다.


[다운로드](#)

.dmg (Intel)

or

[EduTools 플러그인 설치](#)

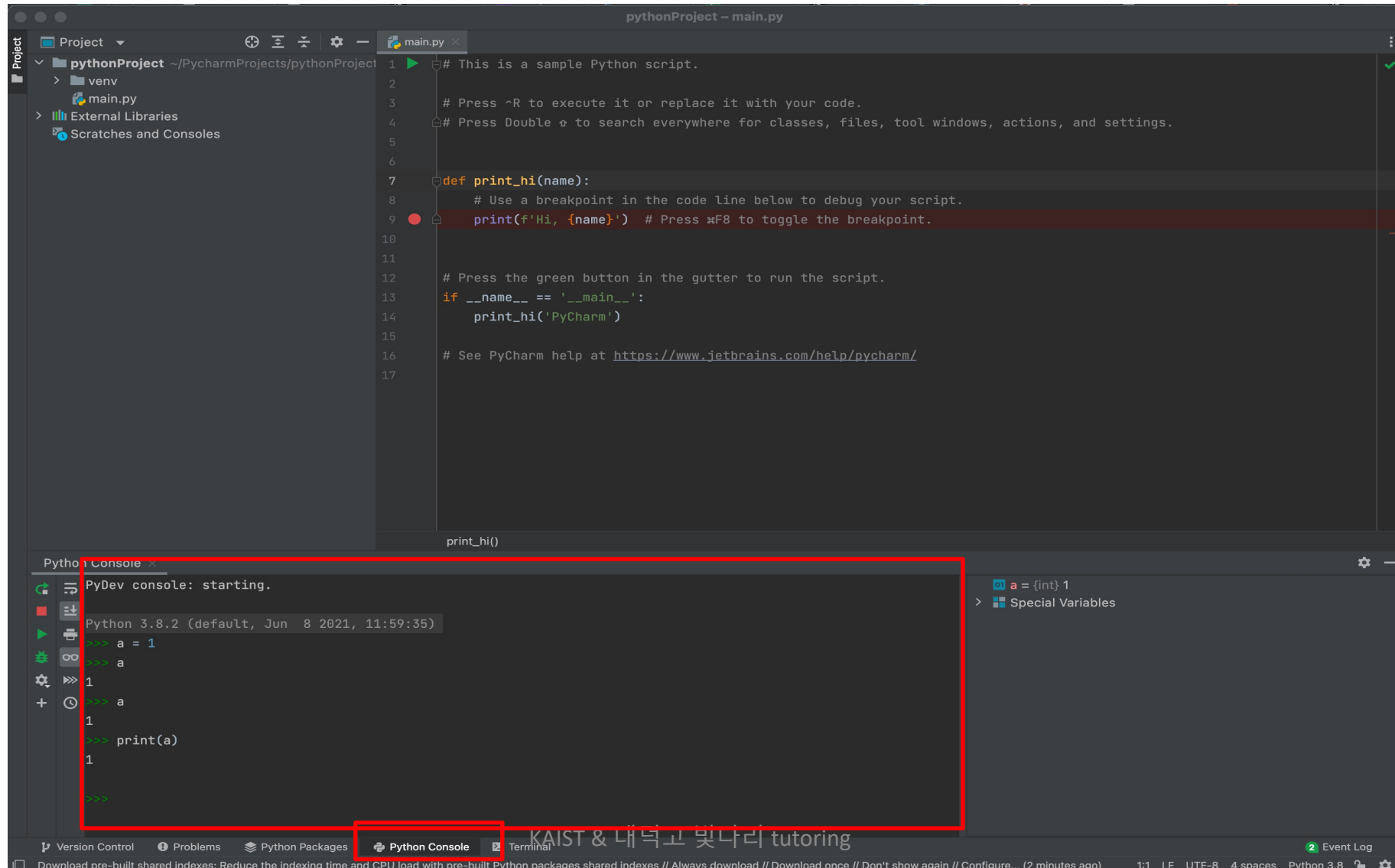
PyCharm Edu Professional 또는 Community를 이미 설치한 경우



JetBrains Academy에서 Python을(를) 배워 보세요
 실제 작동하는 애플리케이션을 제작하며 프로그래밍을 배워보세
 요. 자세한 내용은 [JetBrains Academy 가이드](#)를 참조하세
 요.

<https://bit.ly/3r9EEUi>

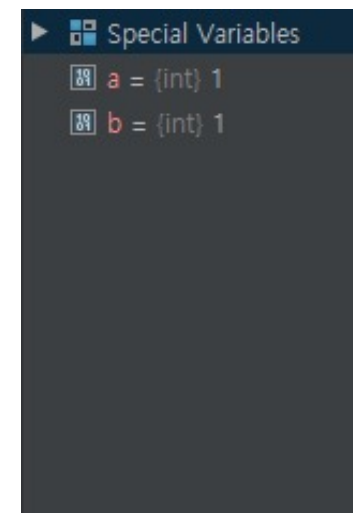
Pycharm 실행



Pycharm

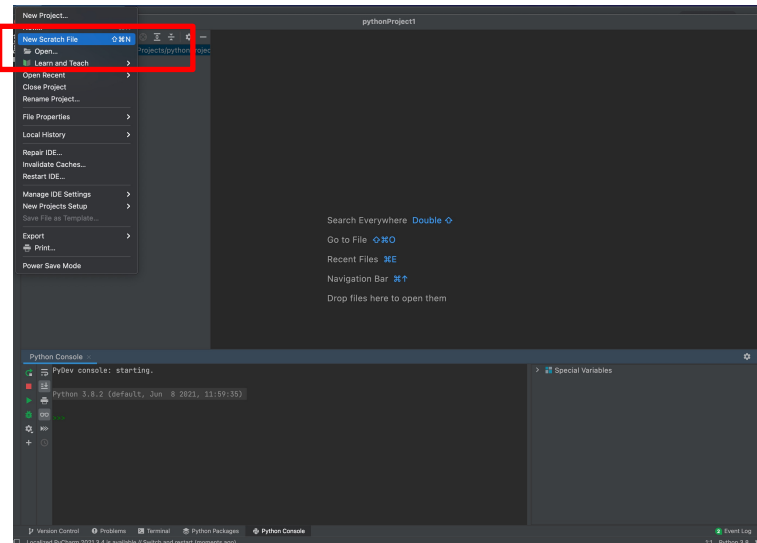
① Python Shell

- ① 짧은 파이썬 명령어를 입력하고, 그 결과 값을 바로 받아볼 수 있다.
- ② Python과 완전히 같은 문법으로 실행된다.
- ③ 특정 파이썬 문법을 확인할 경우에 자주 사용한다.
- ④ PyCharm을 사용할 경우, 현재 정보를 오른쪽에 표현해준다.

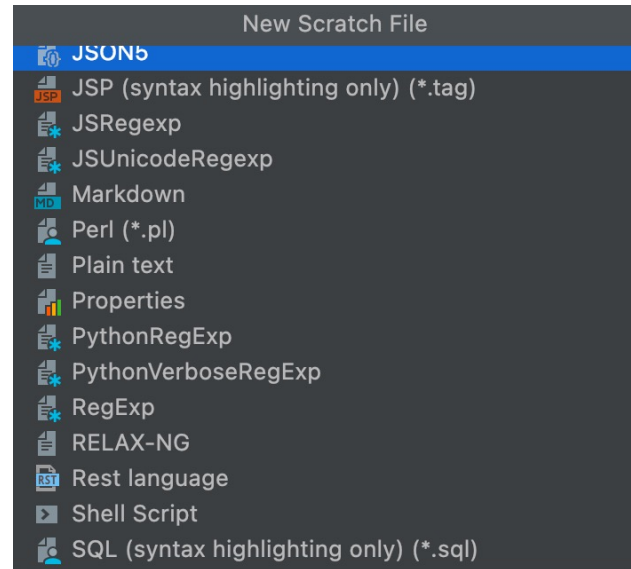


Pycharm

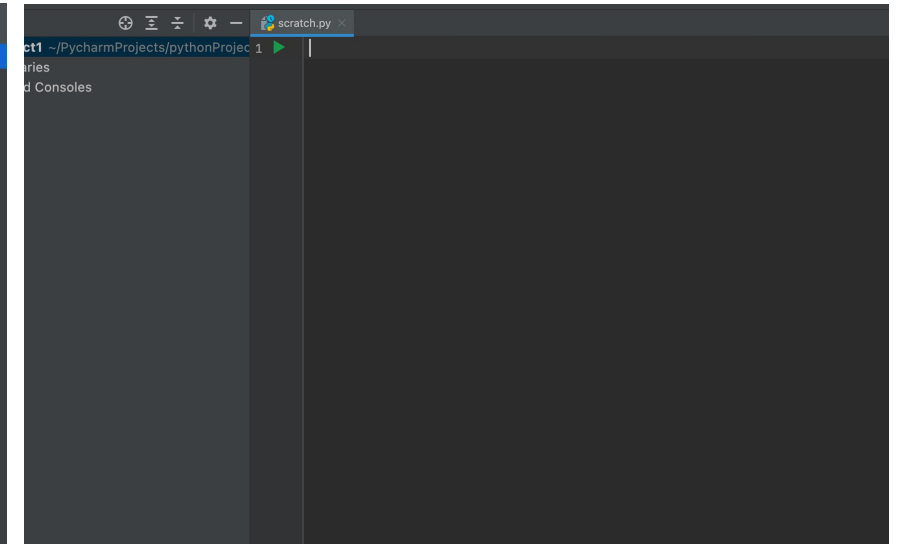
1. New scratch file



2. New Scratch File 화면에서 Python을 클릭



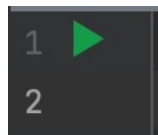
3. scratch.py에서 코딩 시작



Pycharm

② Python File

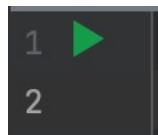
- ① 실행해야 할 많은 양의 명령어들을 파일로 저장해 재활용 할 수 있다.
- ② 초록색 재생버튼을 누르면 파일 전체가 실행된다.



Pycharm

② Python File

- ① 실행해야 할 많은 양의 명령어들을 파일로 저장해 재활용 할 수 있다.
- ② 초록색 재생버튼을 누르면 파일 전체가 실행된다.



Python 맛보기 시작!

Python의 모든 데이터는 object(객체)로 이뤄져 있어요

- 숫자(int, float)
 - “abcd”, “안녕” 같은 문자열(str)
 - True, False를 나타내는 Bool
 - [1,2,3,4,5] 같이 여러 값들이 묶여있는 list
- 등등.....

일단 오늘은 먼저 숫자와 문자열, Bool 에 대해 배워보아요

Python 에서 사칙연산하기

덧셈, 뺄셈, 곱셈, 나눗셈

곱셈과 나눗셈은 각각 *와 / 로 약속

//는 나눗기의 몫

%는 나눗기의 나머지

예시

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

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

```
>>> 2.5 * 7
```

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

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

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

Python 숫자 + 변수

변수는 값을 저장하는 공간

= 기호를 통해서 변수에 값을 저장,
추후에 바꿀 수 있음

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
>>> print(a)
```

Python 문자열 + 변수

변수는 숫자가 아닌 객체들도 대입 가능

문자열, 리스트, 함수 등

문자열에 대해서도 **일부** 연산 적용 가능
(- 는 안돼요)

예시

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

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

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

Python Bool

두 수나 문자열 등이 같은지 비교(==)

수가 크거나 같은지 비교(<, <=, >, >=)

여러개도 사용 가능($1 < 2 < 3$)

예시

```
>>> var1 = "Hello"
```

```
>>> var2 = "KAIST"
```

```
>>> var1 == var2
```

```
>>> 1 < 3
```

```
>>> 1 == 1
```

```
>>> 1 <= 2
```

Python 맛보기

사용자로부터 입력을 받기 위해서는

input 함수 사용

input 함수는 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 1

사용자로부터 숫자를 입력 받아서

100보다 큰지 비교하고

크면 True, 작으면 False를 출력하는 프로그램 작성

Hint: `int(input())`

예제 2

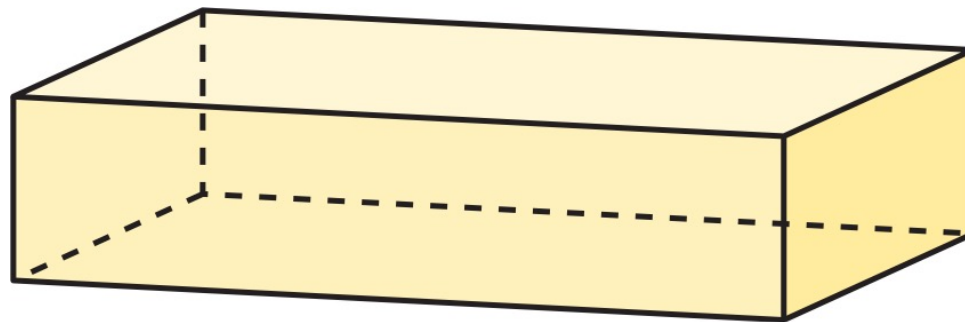
사용자로부터 숫자 2개를 입력 받아,

두 수에 대한 사칙연산의 결과를 출력하는 프로그램을 작성

연산: +, -, *, /, //, %

예제 3

사용자로부터 직육면체의 세 변의 길이를 입력 받아,
직육면체의 부피를 출력하는 프로그램을 작성



예제 4

사용자로부터 이름과 나이를 입력 받아,

“??? 님의 나이는 XX세입니다.”

를 출력하는 프로그램 작성