

Cf. 디버깅

- 디버깅

- 컴퓨터 프로그램 혹은 전자 회로에서 버그(결함)을 찾아내고 없애는 체계적인 과정

- 일반적인 프로그램 디버깅 순서

- 1. 문제가 되는 상황을 재현

- 2. 문제를 재현 하면서 문제의 원인이 되는 코드 위치 (버그 위치)를 찾는다.

- 3. 문제를 발생하는 코드를 수정

- 그 위치 찾는 방법

- 1. Print debugging

- Printf문을 적절한 장소에 삽입, 변수의 변화를 화면에 출력

- Printf문이 출력되면 프로그램이 그 라인을 실행했다는 것을 알수있음

- 그 라인에서의 관심 있는 변수 값을 출력시켜 관찰 할 수 있음

- 2. 디버깅 도구를 이용

- 디버깅 도구를 이용하여 프로그램의 상태(변수 값)를 관찰

- **디버깅 Tool**

- 1. **파이썬 튜터**

- <http://pythontutor.com/>
 - 간단하게 디버깅 돌리기 편함

- 2. **Pycharm**

- 파이썬 프로그램을 쉽게 개발할 수 있도록 하는 IDE
 - <https://www.jetbrains.com/>
 - 단점 - 프로젝트가 커질수록 속도가 느림, 무료 플러그인은 부족, 유료는 성능이 좋음

- 3. **Spyder**

- Anaconda 배포본에 포함되어 있는 IDE
 - <https://www.anaconda.com/>

- 4. **Jupyter**

- IPython 웹을 이용해서 Python 코드가 포함된 동적문서를 작성하는 웹어플리케이션

- 파이썬 튜터

Write code in Python 3.6

```
1 n = int(input('Enter the even number (1~30) : '))
2 while n < 30:
3     n = n + 1
4     if n % 2 == 0:
5         print(n)
6
```

Help improve this tool by completing a [short user survey](#).

Visualize Execution

Live Programming Mode

- 파이썬 튜터

Python 3.6

```
➔ 1 n = int(input('Enter the even number (1~30) : '))  
2 while n < 30:  
3     n = n + 1  
4     if n % 2 == 0:  
5         print(n)
```

[Edit this code](#)

➡ line that has just executed

➔ next line to execute

Click a line of code to set a breakpoint; use the Back and Forward buttons to jump there.



<< First

< Back

Enter user input below:

Forward >

Last >>

Enter the even number (1~30) : 20

Submit

Help improve this tool by clicking whenever you learn something:

I just cleared up a misunderstanding!

I just fixed a bug in my code!

Frames

Objects

- 파이썬 튜터

Python 3.6

```
→ 1 n = int(input('Enter the even number (1~30) : '))
→ 2 while n < 30:
  3     n = n + 1
  4     if n % 2 == 0:
  5         print(n)
```

[Edit this code](#)

→ line that has just executed

→ next line to execute

Click a line of code to set a breakpoint; use the Back and Forward buttons to jump there.



<< First

< Back

Step 2 of 37

Forward >

Last >>

Help improve this tool by clicking whenever you learn something:

I just cleared up a misunderstanding!

I just fixed a bug in my code!

Print output (drag lower right corner to resize)

Enter the even number (1~30) : 20

Frames

Objects

Global frame

n 20

• 파이썬 튜터

Python 3.6

```
1 n = int(input('Enter the even number (1~30) : '))
2 while n < 30:
  → 3     n = n + 1
  → 4     if n % 2 == 0:
5         print(n)
```

[Edit this code](#)

→ line that has just executed

→ next line to execute

Click a line of code to set a breakpoint; use the Back and Forward buttons to jump there.



<< First

< Back

Step 4 of 37

Forward >

Last >>

Help improve this tool by clicking whenever you learn something:

I just cleared up a misunderstanding!

I just fixed a bug in my code!

Print output (drag lower right corner to resize)

Enter the even number (1~30) : 20

Frames

Objects

Global frame

n 21

• 파이썬 튜터

Python 3.6

```
1 n = int(input('Enter the even number (1~30) : '))
→ 2 while n < 30:
3     n = n + 1
→ 4     if n % 2 == 0:
5         print(n)
```

[Edit this code](#)

→ line that has just executed

→ next line to execute

Click a line of code to set a breakpoint; use the Back and Forward buttons to jump there.



<< First

< Back

Step 5 of 37

Forward >

Last >>

Help improve this tool by clicking whenever you learn something:

I just cleared up a misunderstanding!

I just fixed a bug in my code!

Print output (drag lower right corner to resize)

Enter the even number (1~30) : 20

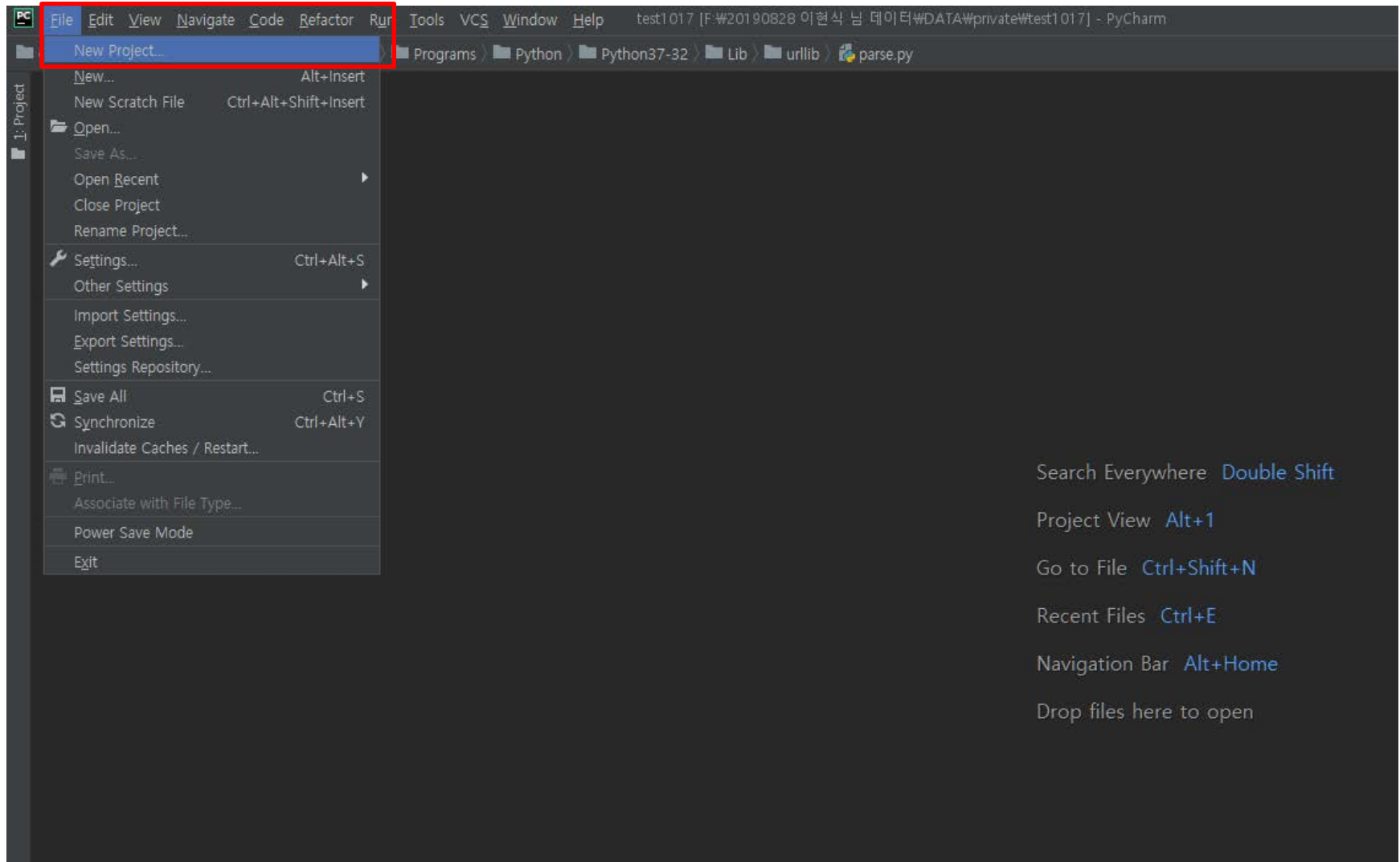
Frames

Objects

Global frame

n 21

- pycharm (프로젝트 생성)



Search Everywhere **Double Shift**

Project View **Alt+1**

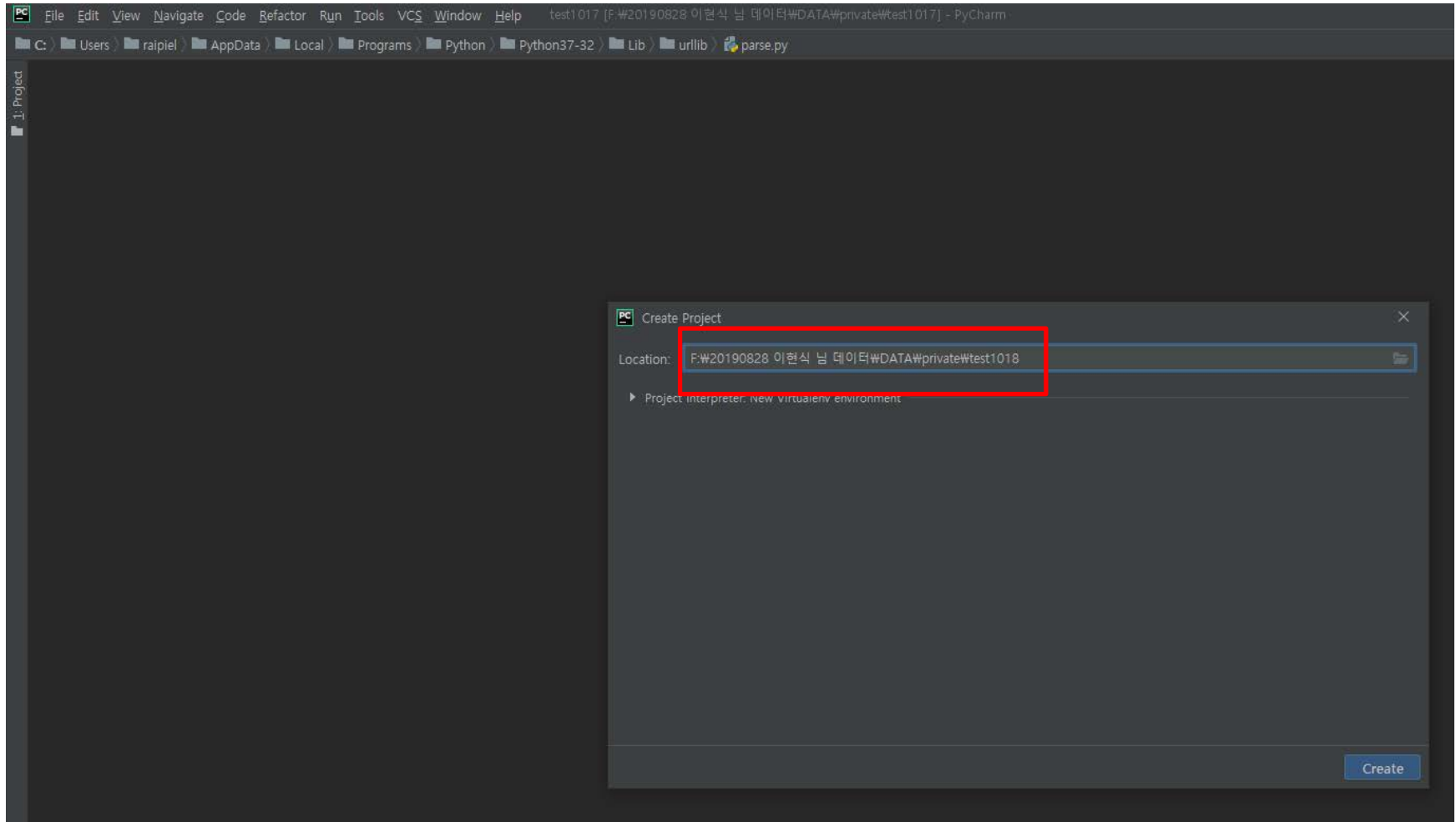
Go to File **Ctrl+Shift+N**

Recent Files **Ctrl+E**

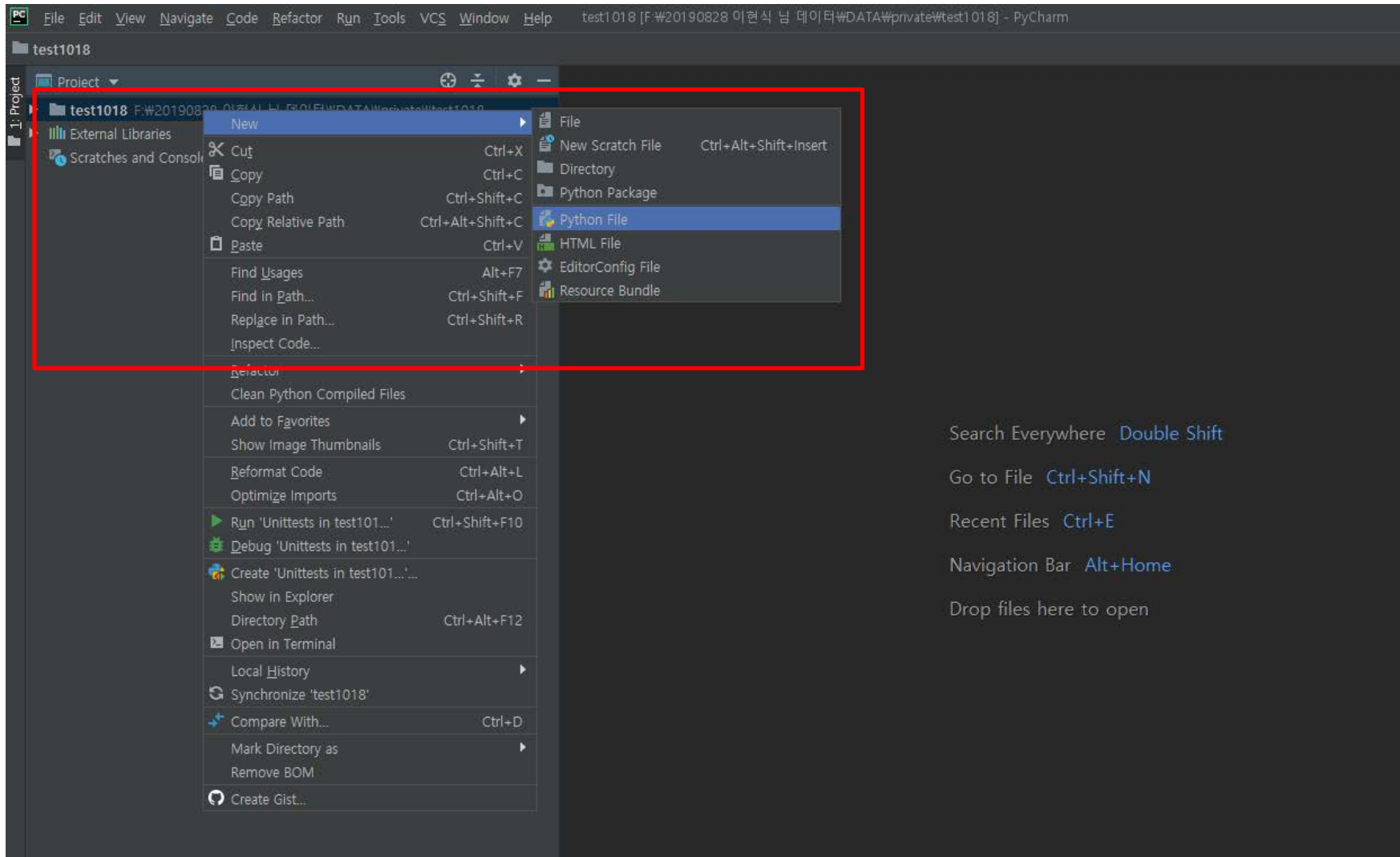
Navigation Bar **Alt+Home**

Drop files here to open

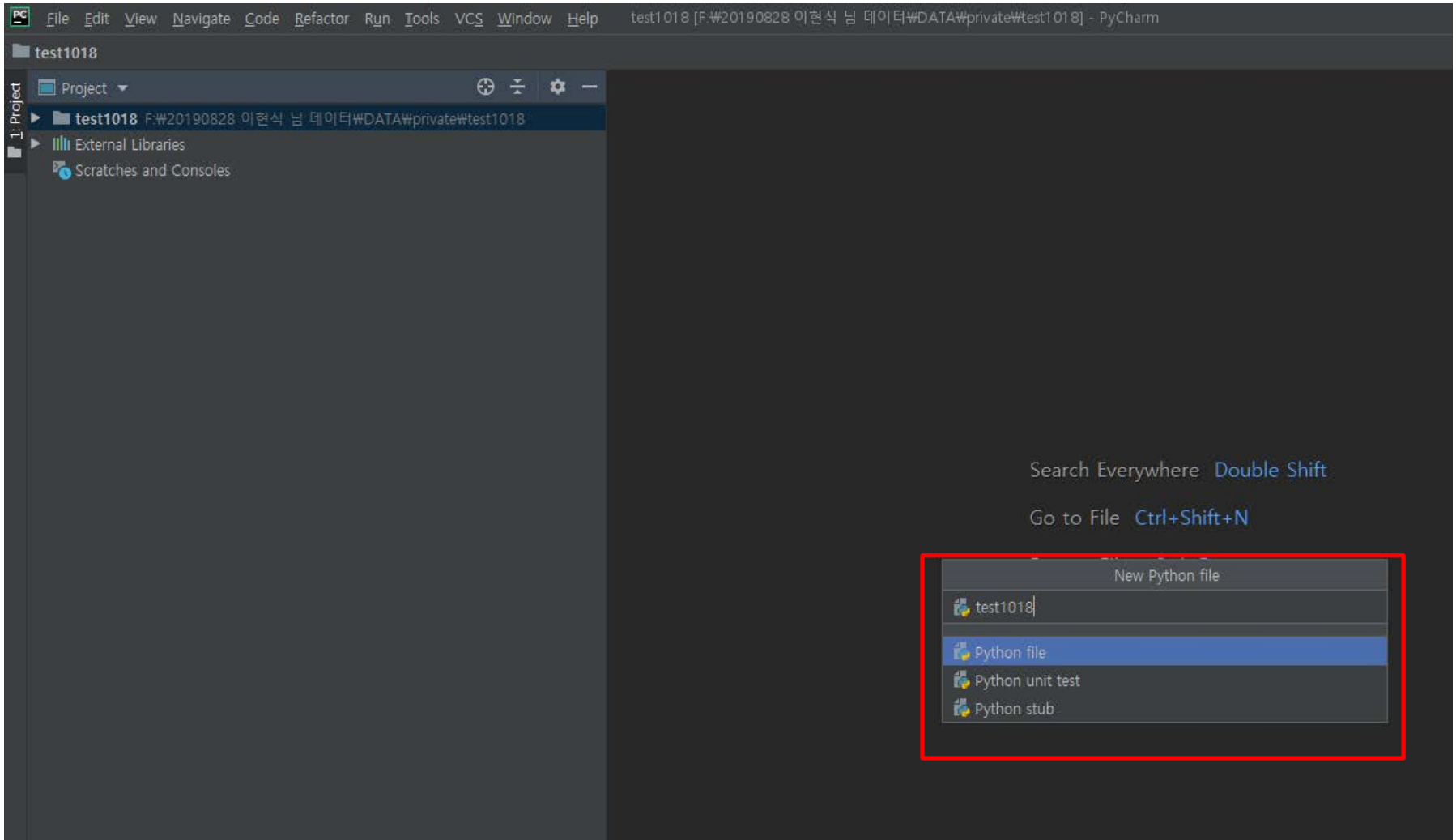
- pycharm (프로젝트 생성)



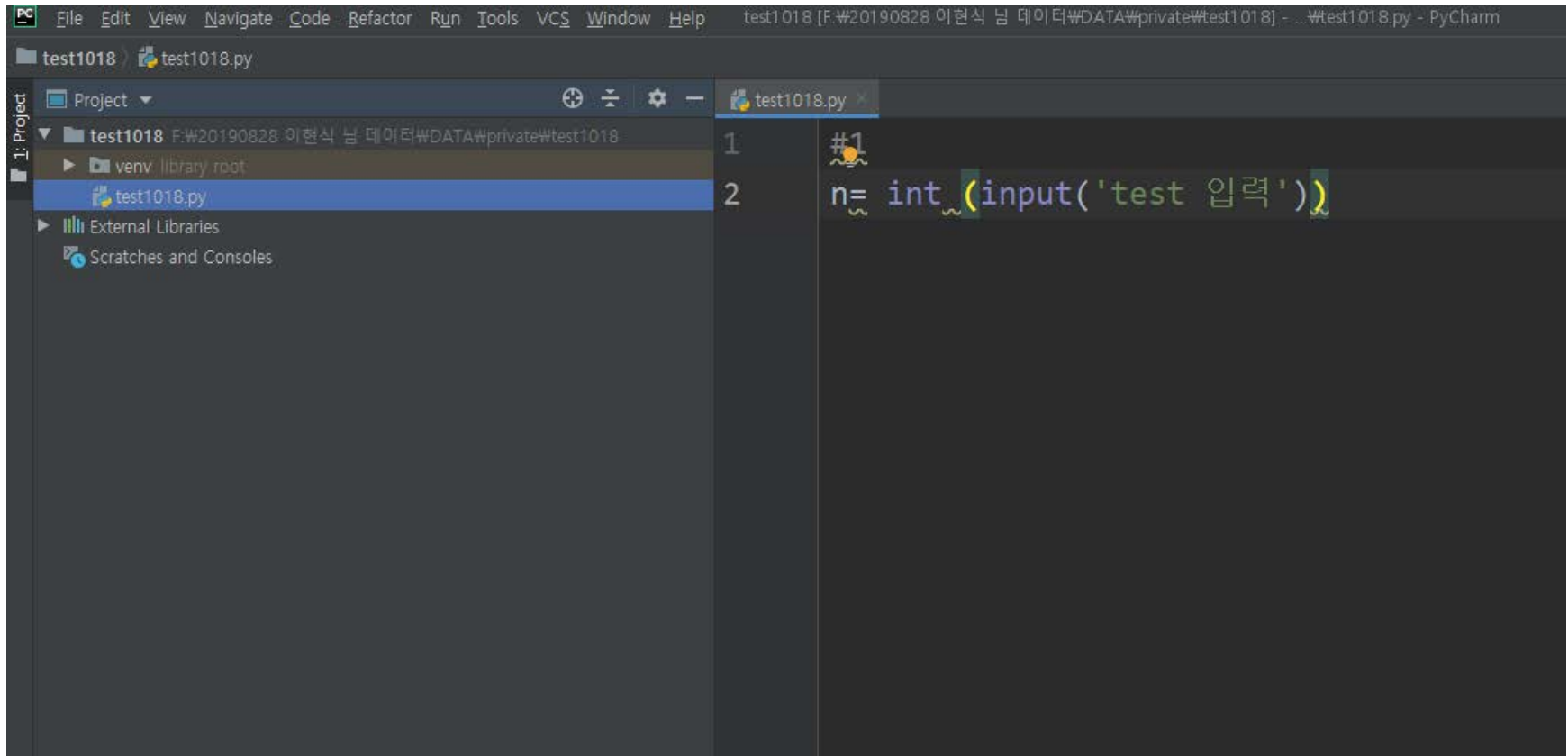
- pycharm (파일 생성) : 커서는 test1018에 두고 마우스 우클릭



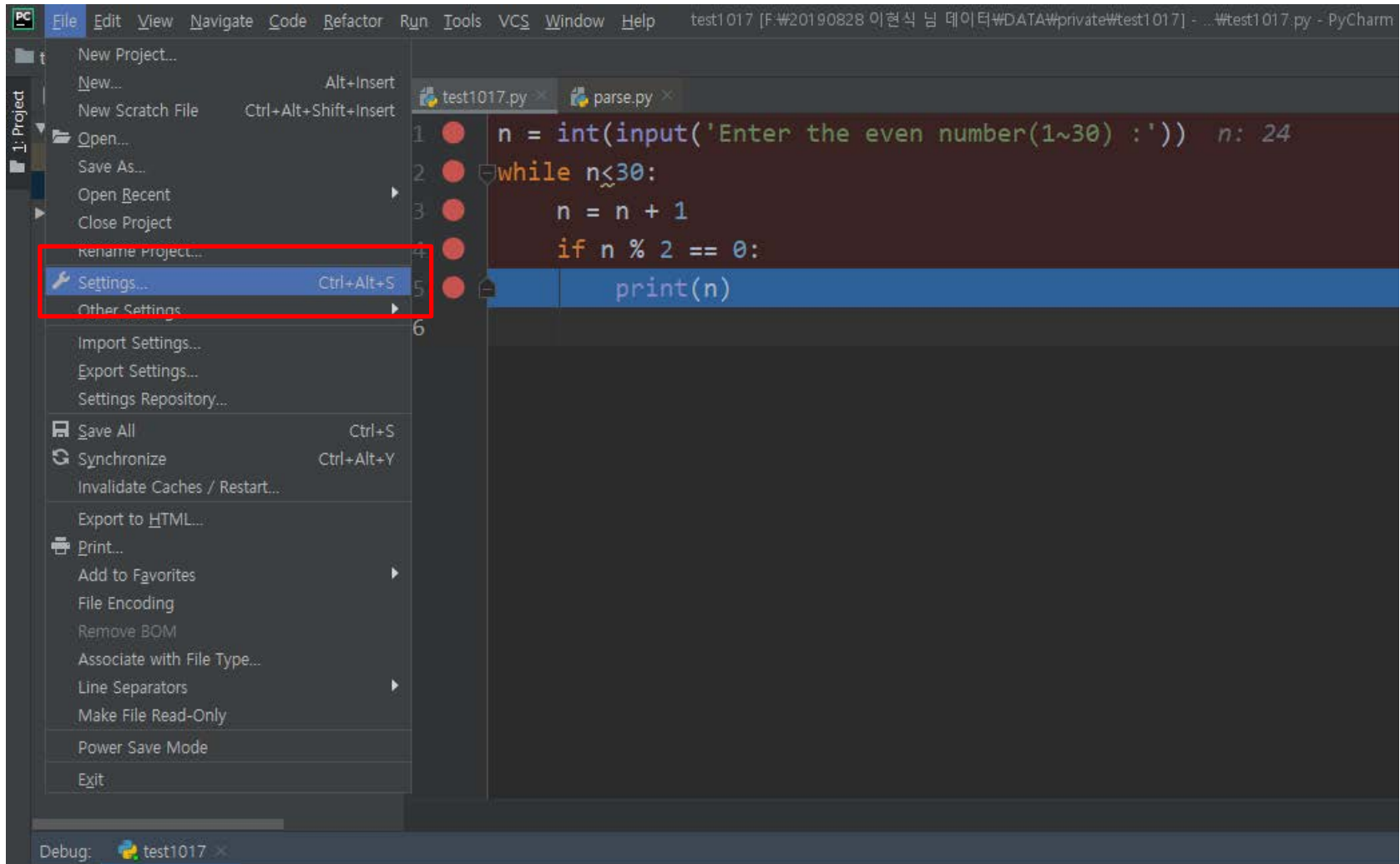
- pycharm (파일 생성)



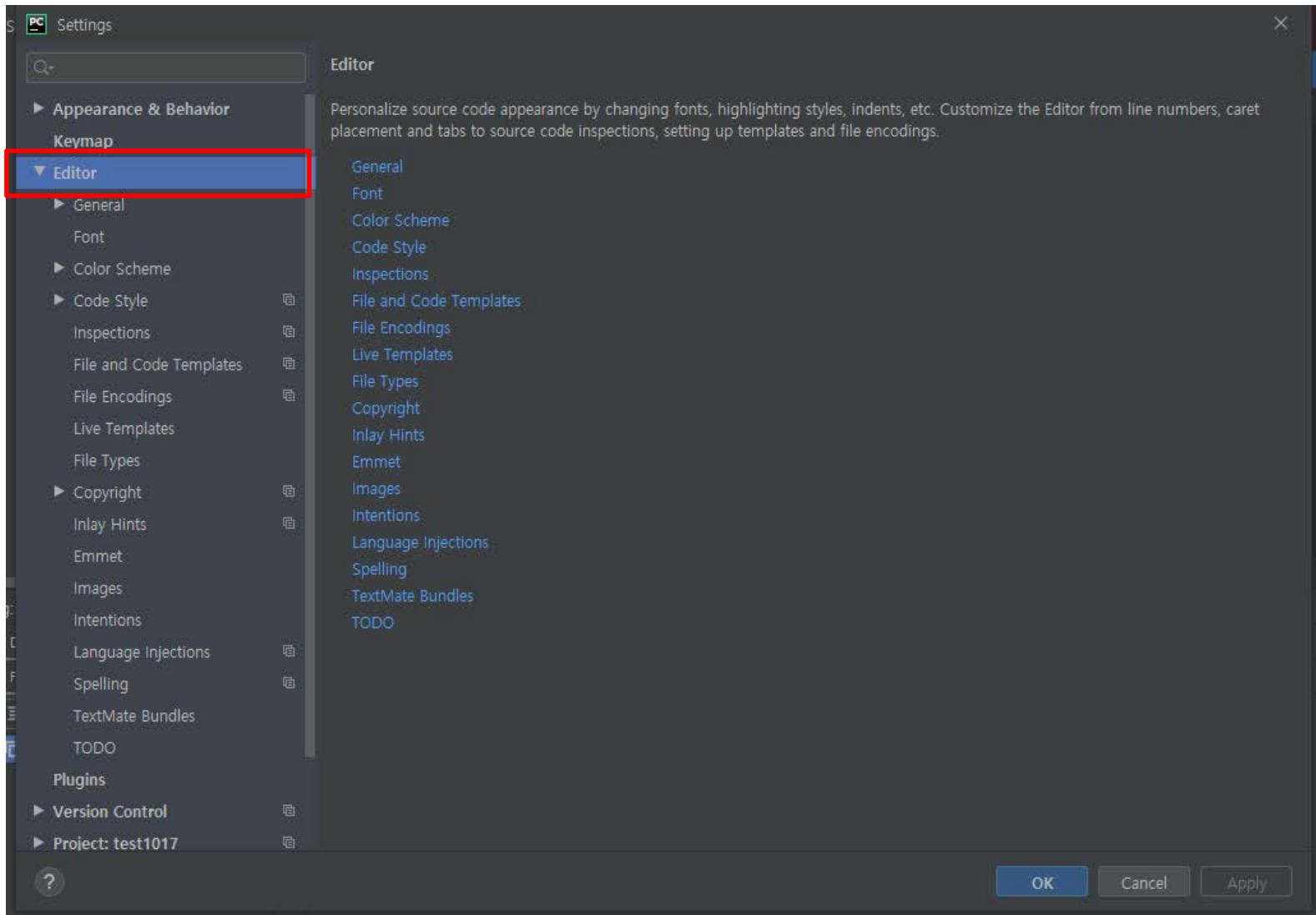
- pycharm (코드 작성)



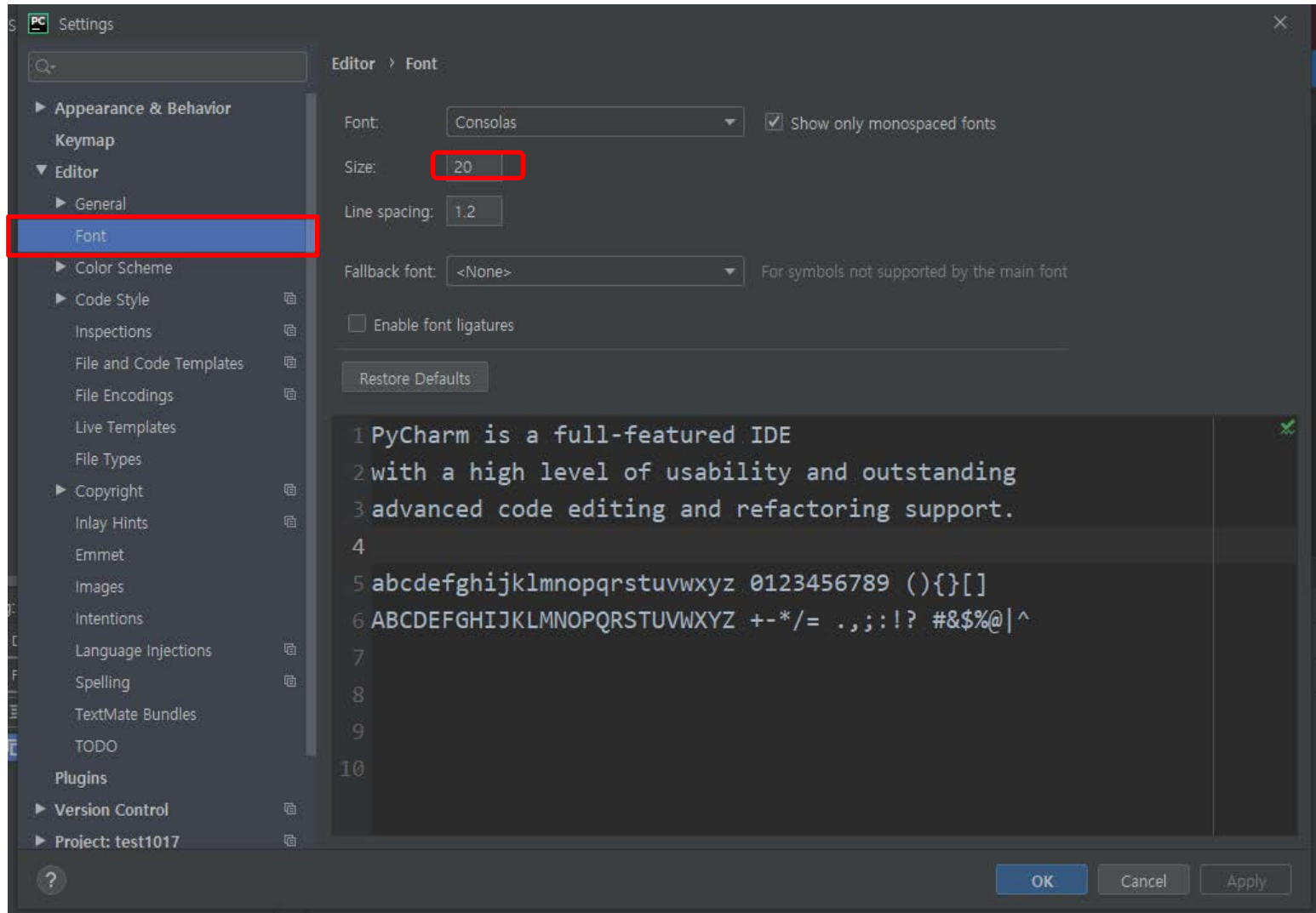
- pycharm (폰트 크기 조절)



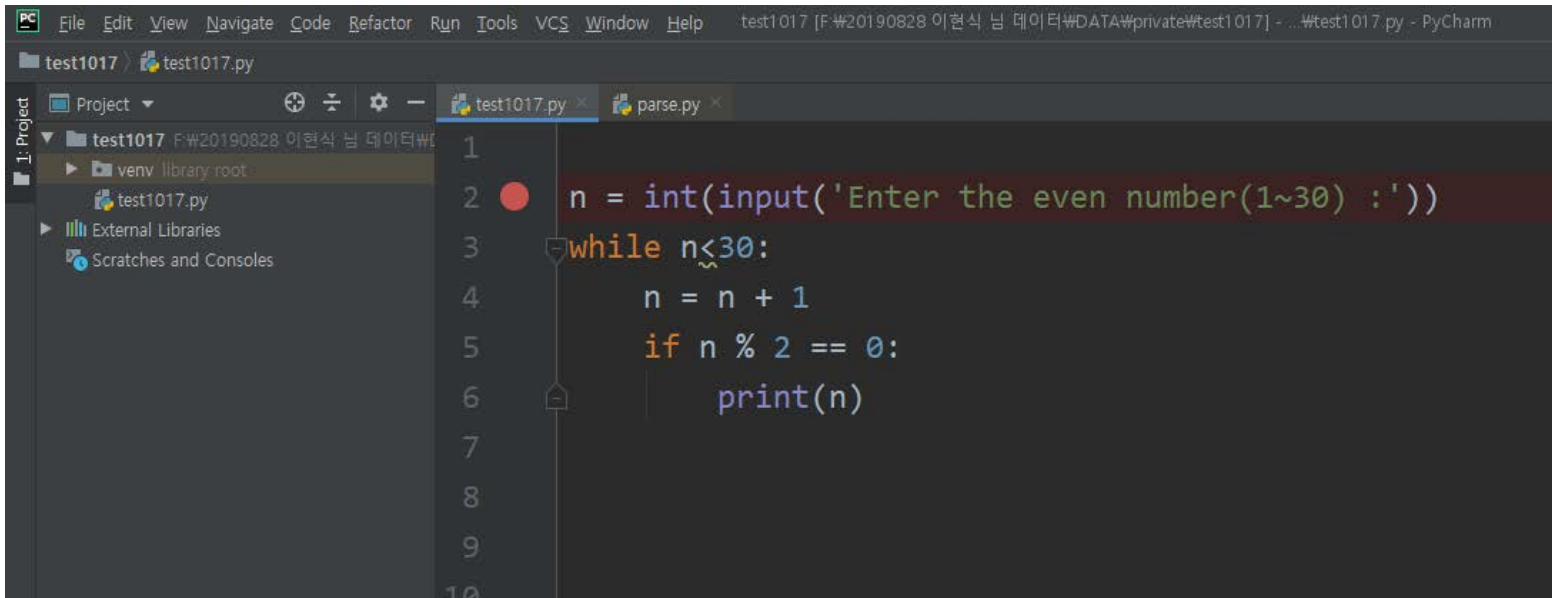
- pycharm



- pycharm

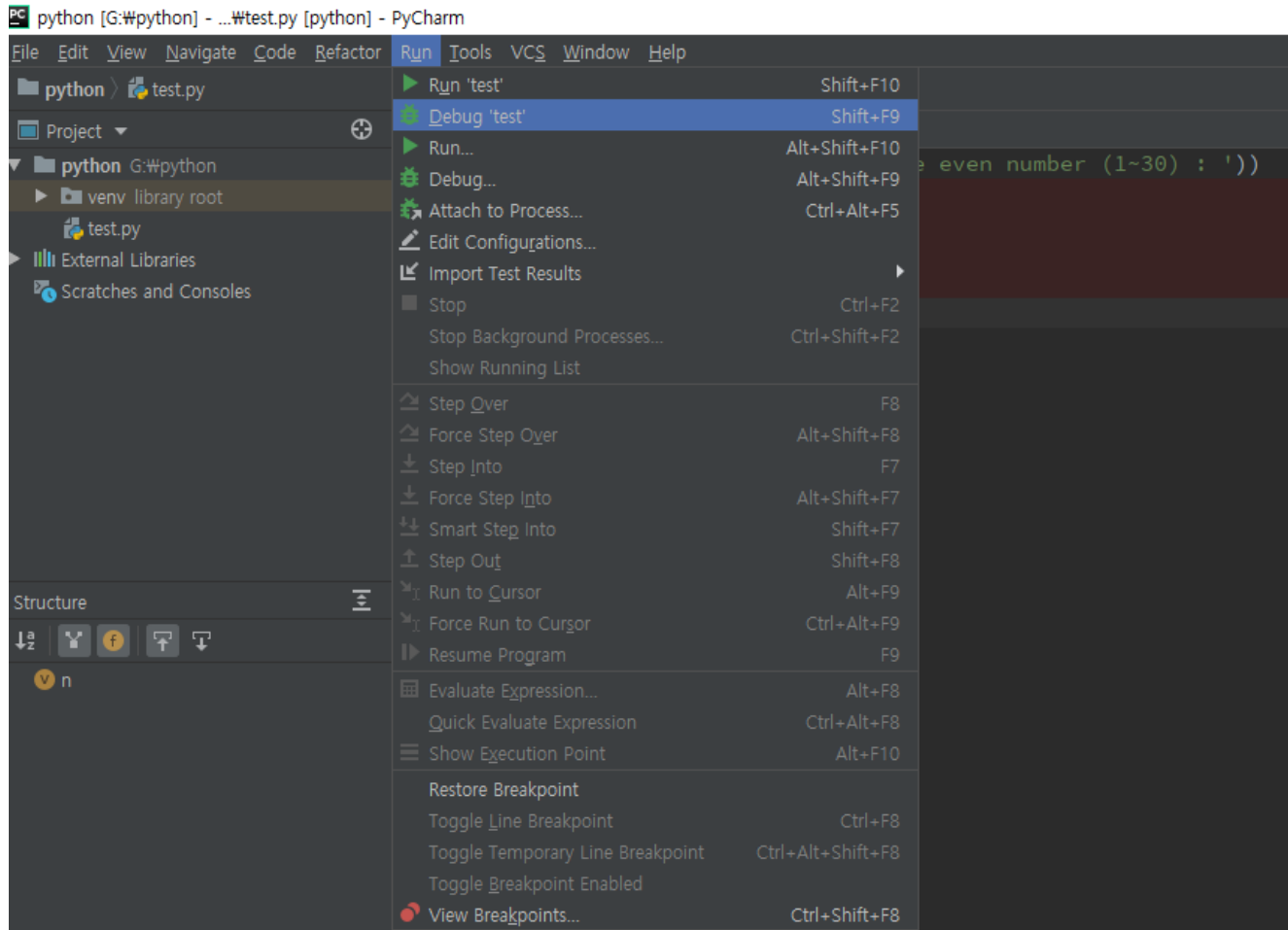


- pycharm (breakpoint)

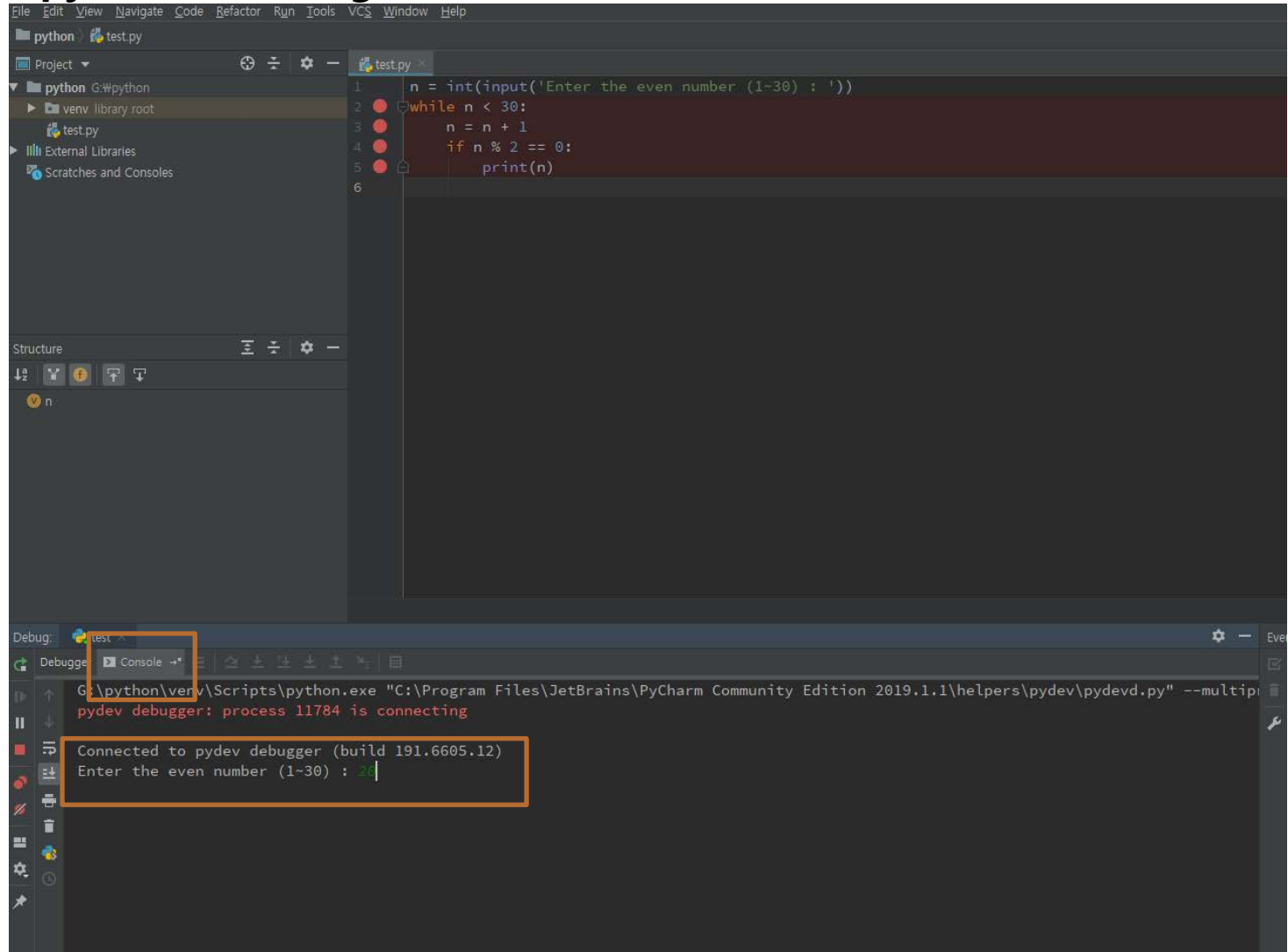


- 행 번호 옆에 좌클릭 하면 breakpoint(중단점) 추가 가능

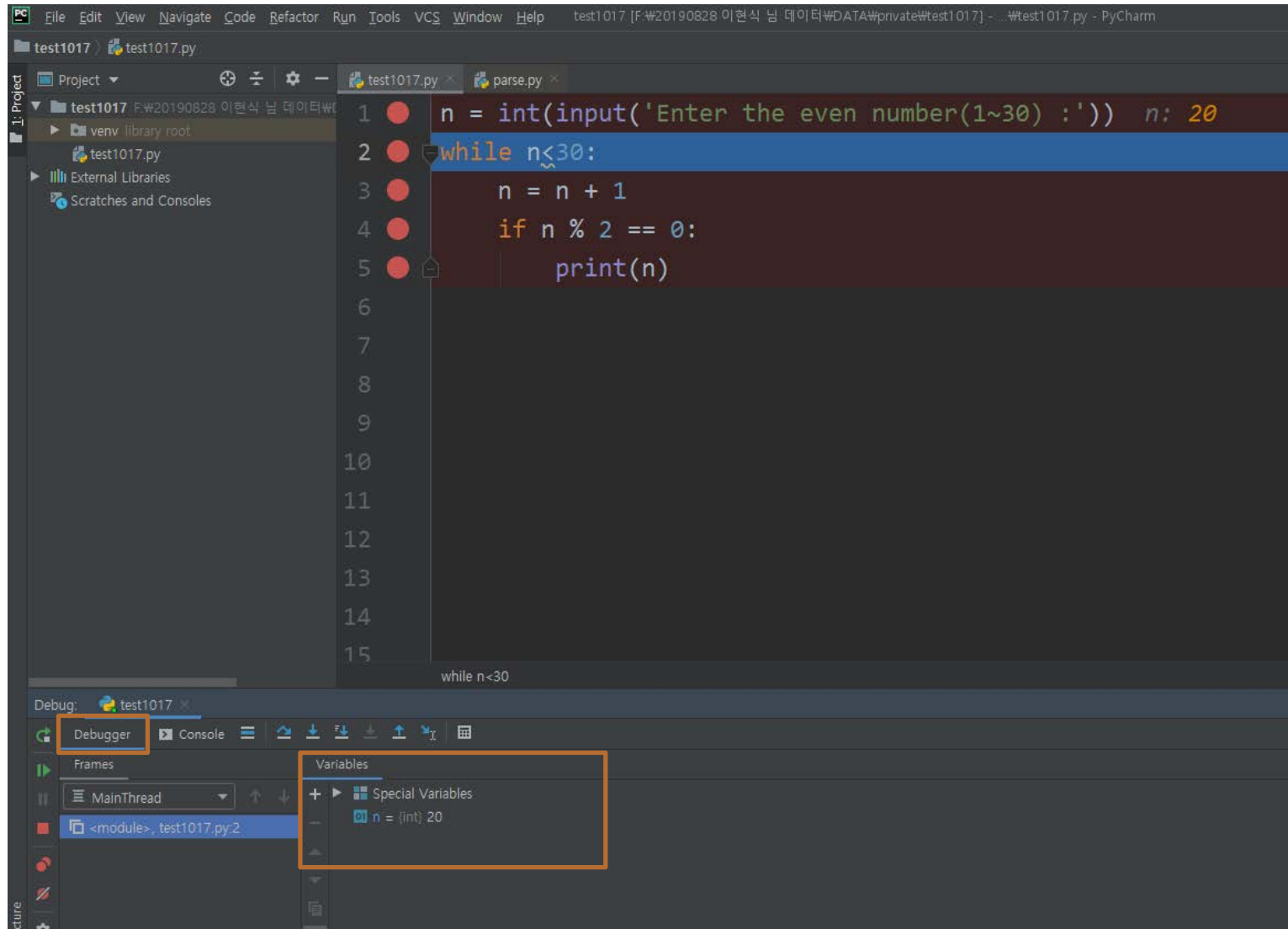
- pycharm (Debug)

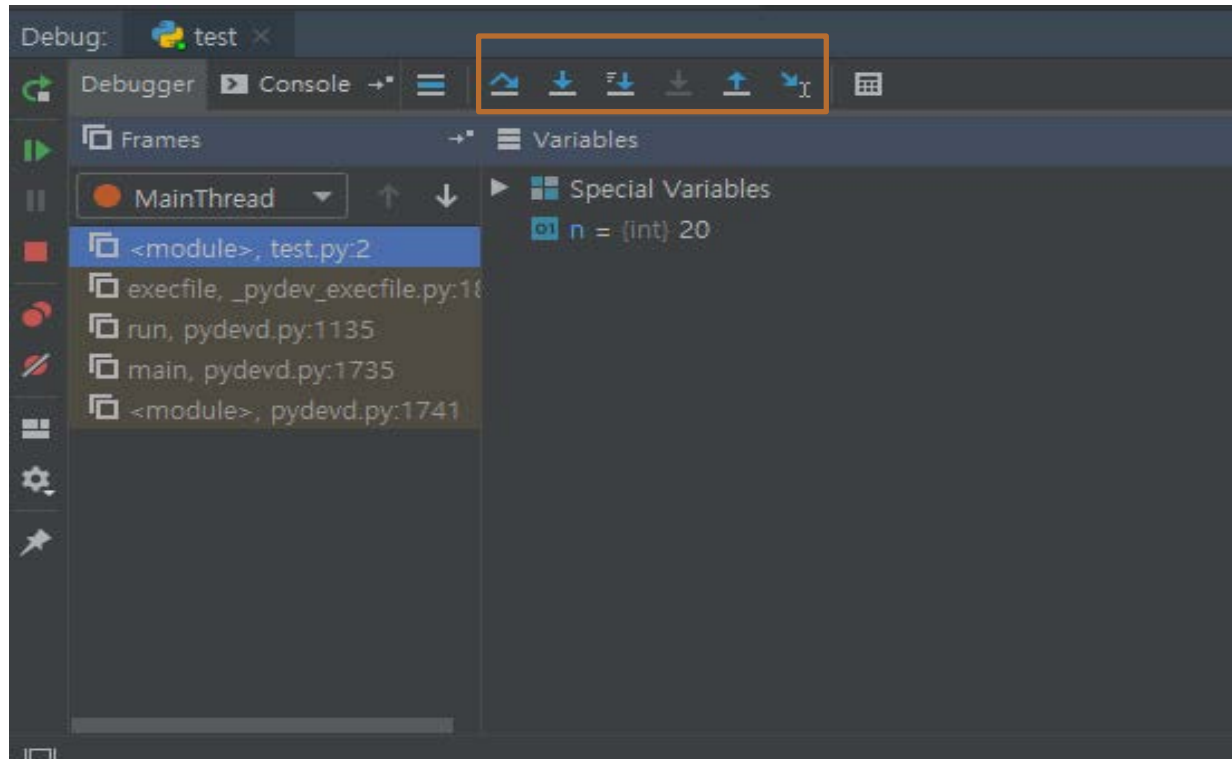


- pycharm (Debug)



- pycharm (Debug)





1. Step over : 중단점이 설정된 파일 안에서 중단점 이후의 코드를 한줄씩 실행
2. Step into : 프로젝트 전체에서 중단점 이후의 코드를 한줄씩 실행
3. Step into My code : 다음 중단점까지 프로젝트 전체에서 내가 작성한 코드에 한해서 중단점 작동
4. Step out : 중단점 이후 실행되는 부분으로 이동
5. Run to Cursor : 현재 커서가 가리키고 있는 라인까지 이동