Scientific Computing with Python Lab

1st session(Jan 23, 2024)

Kaheon Kim

Introduction

Instructor

Kaheon Kim 🇰🇷

- 1st year Ph.D student in ACMS department
- 2017 - 2023: BS in Mathematical Science and Industrial Engineering at UNIST(Ulsan National Institute of Science&Technology)
- Currently doing a research on Statistical Models

: You can see the website through the link appended to my profile in ACMS department website!

Introduction

Office Hour

My office hour

- Time : 9AM ~ 12PM on Thursday
- Location : 205 Crowley(can be switched to Zoom session)
- Other Office hours are operated by Professor Michael, TA Jiabao
- Feel Free to ask question!!!!

Introduction

Lab Session

For this 50 minutes...

- Check the important points dealt with in the class
- Provide the examples based on my coding, TA experience
- Give some of the tips for making a code : making schemes, debugging ... doing assignment!
- Make you good habit for doing an efficient coding

Today,

we are going to deal with

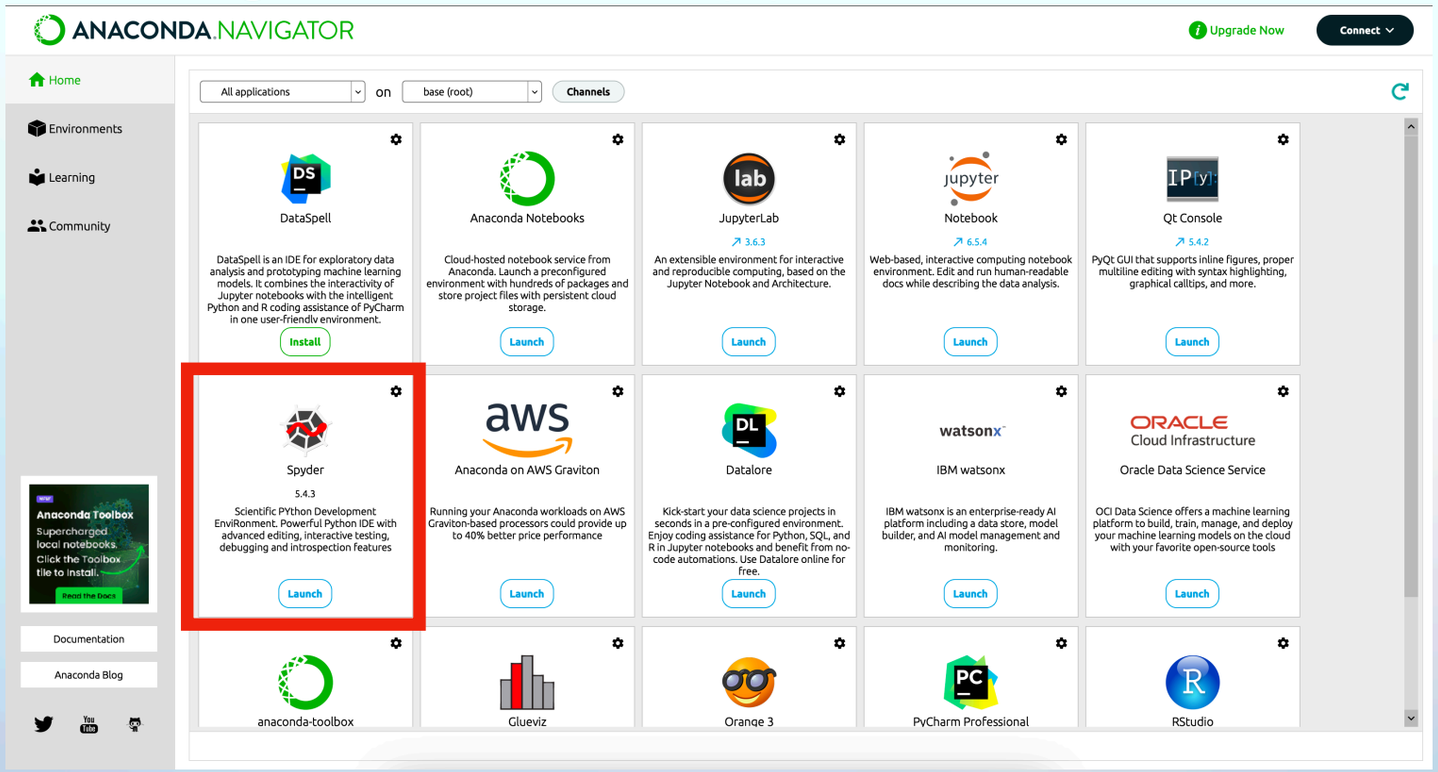
#Installation : Spyder with Anaconda

#Basics : Arithmetics, Print

Installation

Spyder with Anaconda

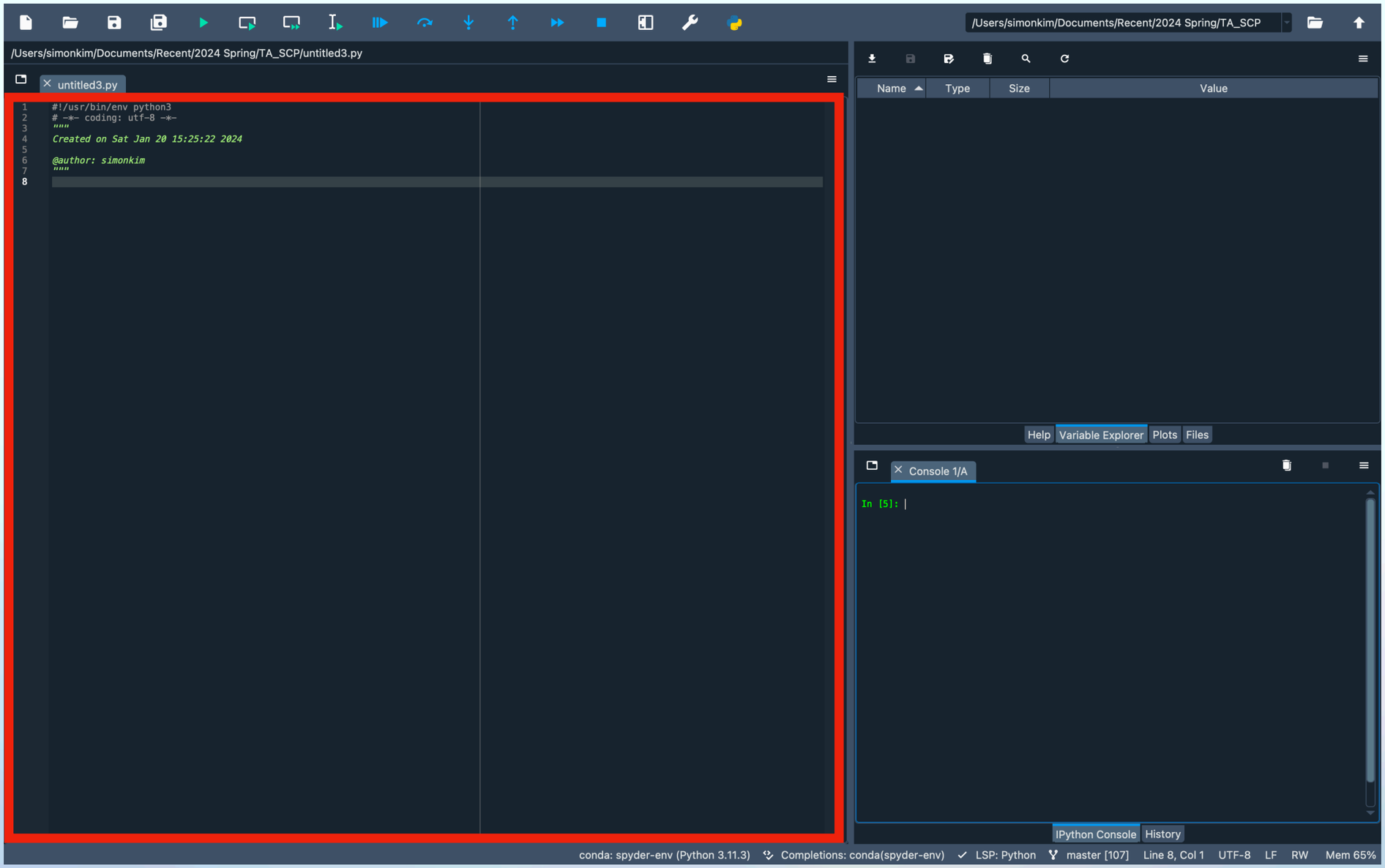
1. Search : anaconda download on Google
2. Install through downloading .exe(Window) or .pkg(Mac) file
3. In anaconda navigator choose Spyder



Let's Install

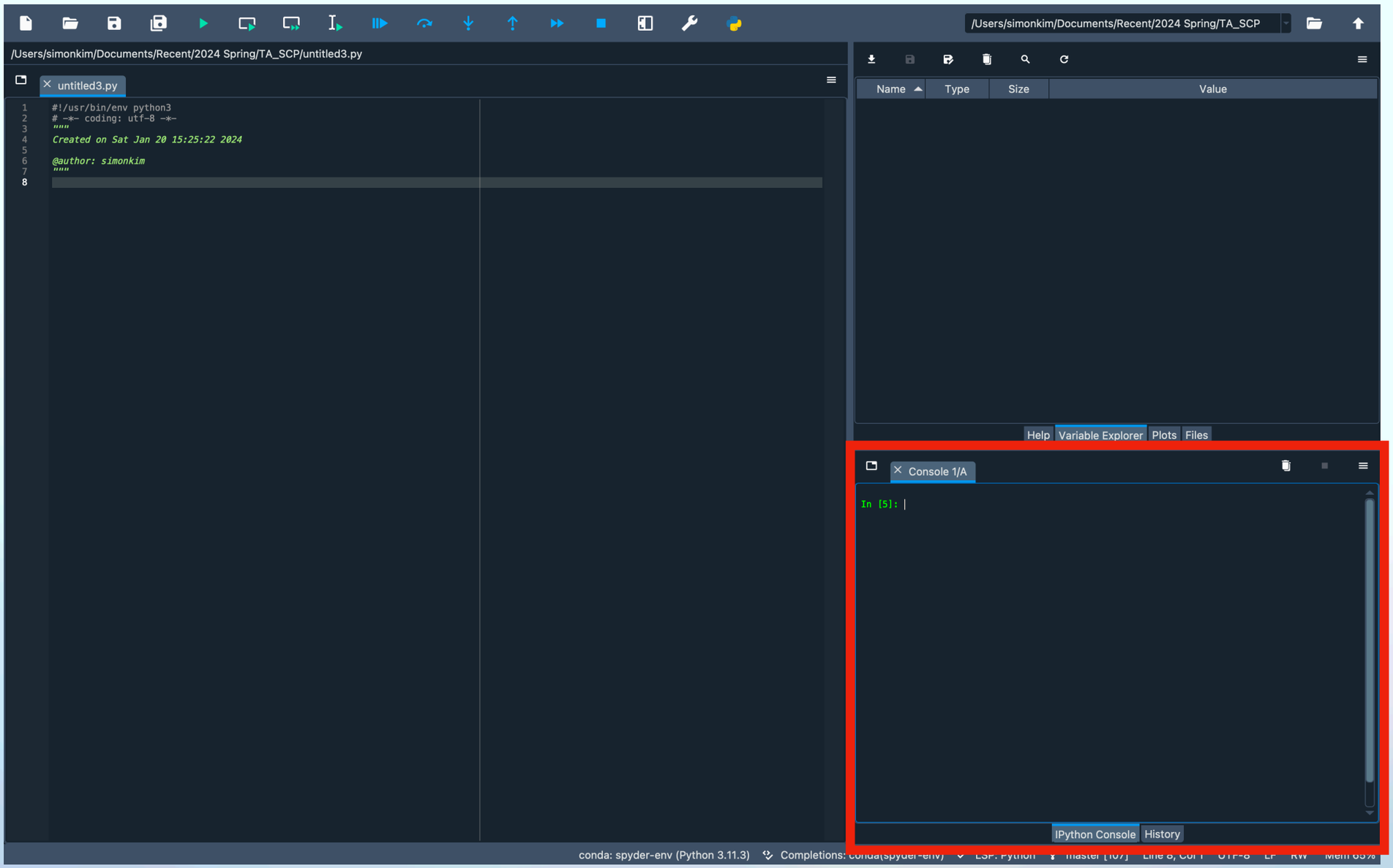
Spyder

Code


/Users/simonkim/Documents/Recent/2024 Spring/TA_SCP
/Users/simonkim/Documents/Recent/2024 Spring/TA_SCP/untitled3.py
untitled3.py
#!/usr/bin/env python3
-*- coding: utf-8 -*-
"""
Created on Sat Jan 20 15:25:22 2024
@author: simonkim
"""
Name
Type
Size
Value
Help
Variable Explorer
Plots
Files
Console 1/A
In [5]:
IPython Console
History
conda: spyder-env (Python 3.11.3)
Completions: conda(spyder-env)
LSP: Python
master [107]
Line 8, Col 1
UTF-8
LF
RW
Mem 65%

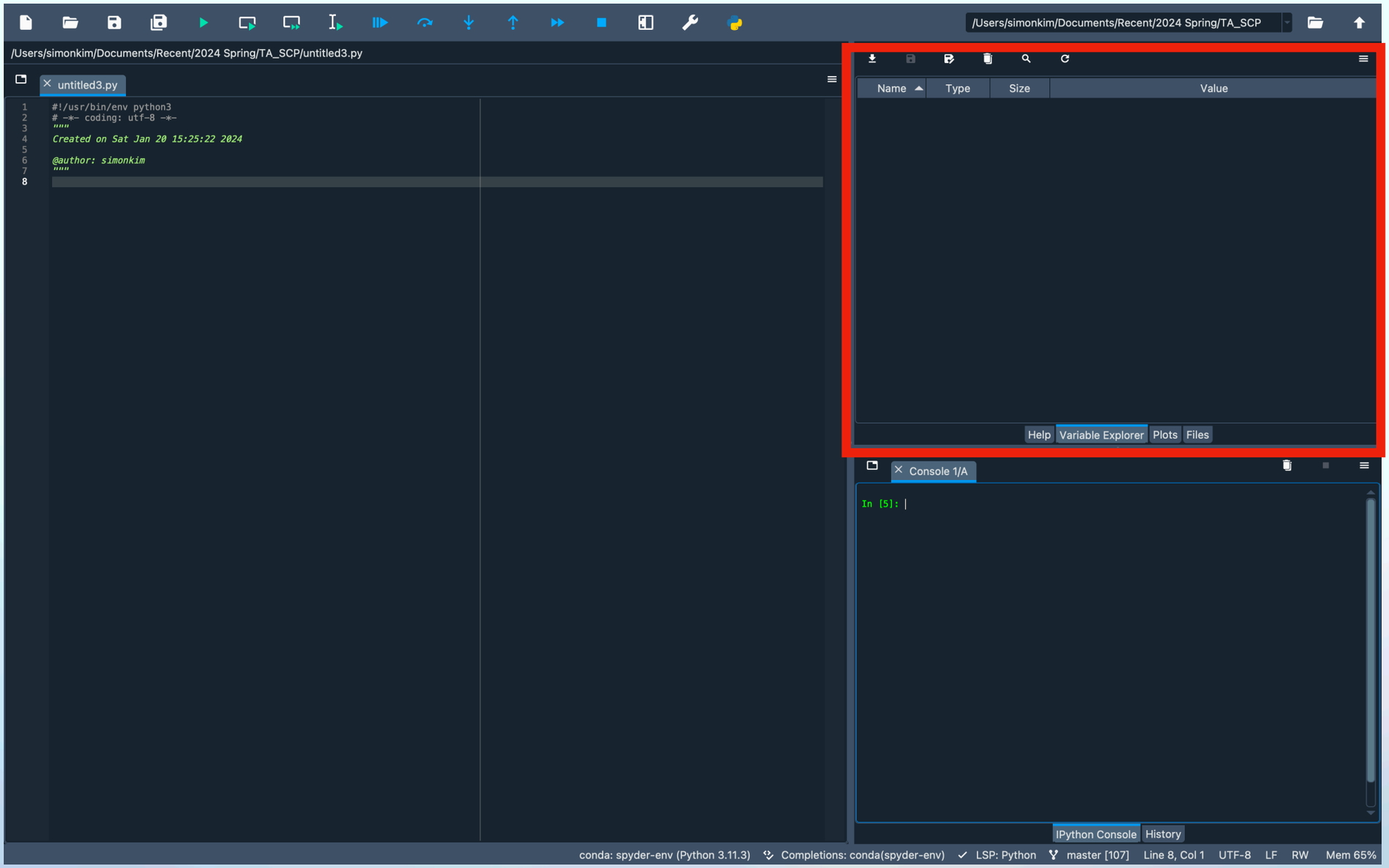
Spyder

Console : We can see the print output value



Spyder

Variable Explorer



Start Coding

Really Important! Why?

1. Make clear Directory

- Clear Location + Clear Hierarchy between folders(Standard)

ex) Chapter, Assignments, Exams

2. Set the directory location for each .py file

- Clear naming rule between files

3. Save it

Let's make coding settings

Variable

Input

- Direct Assignment

```
a = 2
```

- Input function

```
c = input("enter the code : ")
```

- Update the value assignment

```
a = a+1
```

Variable

Data Type

1.Integer(int) : integer ex) 1

2.Float(float) : real numbers ex) 1.0, 1.2

3.String(str) : an array of letters ex) ‘hello’, ‘a’, ‘1’

Variable

Difference between integer and Float

float : Numerical Calculation Friendly

```
In [9]: f(x) = -3/2*ln(x/2)-3/10*x^2+2*x
        p_i = 2.0
        i =1

while abs(p_i - f(p_i)) > 10^(-8):
    print("At step", i, "we have", p_i)
    p_i = f(p_i)
    i += 1
```

```
At step 1 we have 2.00000000000000
At step 2 we have 2.80000000000000
At step 3 we have 2.74329164506818
At step 4 we have 2.75487155146057
At step 5 we have 2.75261246765400
At step 6 we have 2.75305741357807
At step 7 we have 2.75296993997349
At step 8 we have 2.75298714302695
At step 9 we have 2.75298376002026
At step 10 we have 2.75298442530308
At step 11 we have 2.75298429447272
```

Variable

Difference between integer and Float

int : Algebraic Calculation Friendly

```
In [10]: f(x) = -3/2*ln(x/2)-3/10*x^2+2*x
         p_i = 2
         i =1

         while abs(p_i - f(p_i)) > 10^(-8):
             print("At step", i, "we have", p_i)
             p_i = f(p_i)
             i += 1
```

```
At step 1 we have 2
At step 2 we have 14/5
At step 3 we have -3/2*log(7/5) + 406/125
At step 4 we have -3/625000*(375*log(7/5) - 812)^2 - 3*log(7/5) - 3/2*log(-3/4*log(7/5) + 203/125) + 812/125
At step 5 we have -3/3906250000000*(3*(375*log(7/5) - 812)^2 + 1875000*log(7/5) + 937500*log(-3/4*log(7/5) + 203/125)
- 4060000)^2 - 3/312500*(375*log(7/5) - 812)^2 - 6*log(7/5) - 3/2*log(-3/1250000*(375*log(7/5) - 812)^2 - 3/2*log(7/
5) - 3/4*log(-3/4*log(7/5) + 203/125) + 406/125) - 3*log(-3/4*log(7/5) + 203/125) + 1624/125
At step 6 we have -3/1525878906250000000000000000*(3*(3*(375*log(7/5) - 812)^2 + 1875000*log(7/5) + 937500*log(-3/4*lo
g(7/5) + 203/125) - 4060000)^2 + 37500000*(375*log(7/5) - 812)^2 + 23437500000000*log(7/5) + 5859375000000*log(-3/125
0000*(375*log(7/5) - 812)^2 - 3/2*log(7/5) - 3/4*log(-3/4*log(7/5) + 203/125) + 406/125) + 11718750000000*log(-3/4*lo
g(7/5) + 203/125) - 50750000000000)^2 - 3/1953125000000*(3*(375*log(7/5) - 812)^2 + 1875000*log(7/5) + 937500*log(-3/
4*log(7/5) + 203/125) - 4060000)^2 - 3/156250*(375*log(7/5) - 812)^2 - 12*log(7/5) - 3/2*log(-3/7812500000000*(3*(375
*log(7/5) - 812)^2 + 1875000*log(7/5) + 937500*log(-3/4*log(7/5) + 203/125) - 4060000)^2 - 3/625000*(375*log(7/5) - 8
12)^2 - 3*log(7/5) - 3/4*log(-3/1250000*(375*log(7/5) - 812)^2 - 3/2*log(7/5) - 3/4*log(-3/4*log(7/5) + 203/125) - 4
6/125) - 3/2*log(-3/4*log(7/5) + 203/125) + 812/125) - 3*log(-3/1250000*(375*log(7/5) - 812)^2 - 3/2*log(7/5) -
```

Variable

Data Type

1.Integer(int) : integer ex) 1

2.Float(float) : real numbers ex) 1.0, 1.2

3.String(str) : an array of letters ex) ‘hello’, ‘a’, ‘1’

Converting the types

```
a = float(a)

a = str(a)

a = int(a)
```

Let’s do exercise

Arithmetics

Operations

1. Addition : +
2. Subtraction : -
3. Multiplication : *
4. Division : /
5. Power : **
6. Quotient : //
7. Remainder : %

```
# Arithmetic

## 1. Addition
d = 5 + 3

## 2. Subtraction
e = 5 - 3

## 3. Multiplication
f = 5 * 3

## 4. Division
g = 5 / 3

## 5. Power
h = 5 ** 3

## 6. Quotient
i = 5 // 3

## 7. Remainder
j = 5 % 3
```

Arithmetics

Advanced

1. Brackets for more complicated calculations

eg) $((2+4)\times 5)^3$

```
: ((2+4)*5)**3
```

2. math package for several functions

Installation :

```
In [19]: pip install math
```

eg) π, exp, log, cos, sin, tan, arccos, arcsin, arctan

Arithmetics

Advanced

1. Brackets for more complicated calculations

eg) $((2+4)\times 5)^3$

```
((2 + 4) * 5) ** 3
```

Order : Same as mathematical calculation!

power → division, multiplication, quotient, remainder →addition, subtraction

2. math package for several functions

Installation :

```
In [19]: pip install math
```

eg) π, exp, log, cos, sin, tan, arccos, arcsin, arctan

Let’s do exercise

Print functions

print options

1. Align values

print(a, b, c)

where a, b, c is integer, float or string

2. F-string

print('I want to print %d,%f, and %s'%(a, b, c))

- integer : **%d** / float : **%f** / string : **%s**

Print functions

More details on F-string on float

1. integer, string are finite sequence, but float is infinite sometimes
2. %.**3**f : **three** decimals

```
print("2 divided by 3 is %.3f"%(2/3))
```

Print functions

More details on F-string on float

1. integer, string are finite sequence, but float is infinite sometimes
2. %.**3**f : **three** decimals

```
print("2 divided by 3 is %.3f"%(2/3))
```

Output : 0.667

```
2 divided by 3 is 0.667
```

Let’s do exercise