

Hướng dẫn lập trình
NGÔN NGỮ PYTHON

(Cho học phần Toán đại cương - Phiên bản 12/2021)



LỜI NÓI ĐẦU

“Ngôn ngữ lập trình phổ biến nhất trên thế giới”

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido Van Rossum. Thiết kế bắt đầu vào cuối những năm 1980 và được phát hành lần đầu tiên vào tháng 2 năm 1991.

Tính năng chính của Python: Ngôn ngữ lập trình đơn giản, dễ học; Miễn phí, mã nguồn mở; Khả năng di chuyển: chạy được trên Windows, Mac OS, Linux. Khả năng mở rộng và có thể nhúng: kết hợp các ngôn ngữ khác vào code Python; Ngôn ngữ thông dịch cấp cao: không phải quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,...; Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến; Hướng đối tượng.

Python được dùng trong: Lập trình ứng dụng web; Khoa học và tính toán (đặc biệt trong Machine Learning, Data mining và Deep Learning); Tạo nguyên mẫu phần mềm (bản chạy thử – prototype); Ngôn ngữ tốt để dạy lập trình.

Tài liệu ngắn này chúng tôi viết dựa theo một số tài liệu trên internet nhằm giúp các bạn sinh viên làm quen với việc sử dụng một ngôn ngữ lập trình để giải các bài toán của học phần Toán đại cương.

LÊ VĂN TUẤN¹

¹ Bộ môn Toán – Trường Đại học Thương mại

MỤC LỤC

Chủ đề 0. Hello World

Chủ đề 1. Tính toán trên trường số thực

Chủ đề 2. Vector, ma trận và định thức

Chủ đề 3. Hệ phương trình tuyến tính

Chủ đề 4. Đồ thị của hàm số

Chủ đề 5. Đạo hàm

Chủ đề 6. Giá trị nhỏ nhất – lớn nhất

Chủ đề 7. Các tham số đặc trưng

Phụ lục: Download, cài đặt và chạy “Hello World”

Chủ đề 0. Hello World

Ví dụ này sẽ thực hiện: in chữ Hello World! và in giá trị của $\sin 0$

```
import math  
  
print("Hello World !")  
print(math.sin(0))
```

- Câu lệnh `import math` để nhập thư viện `math`
- Lệnh `print` để in chữ/số ra cửa sổ kết quả

Chủ đề 1. Tính toán trên trường số thực

1. Các phép toán trên trường số thực là:

cộng (+), trừ (-), nhân (*), chia (/), lũy thừa (**)

2. Các hàm thông dụng trong thư viện math:

```
import math
```

Hàm toán học	Trên Python	Hàm toán học	Trên Python
π	math.pi	$\sin(x)$	math.sin(x)
e	math.e	$\cos(x)$	math.cos(x)
$+\infty$; $-\infty$	math.inf ; -math.inf	$\tan(x)$	math.tan(x)
$\sqrt[2]{x}$	math.sqrt(x)	$\cot(x)$	1/math.tan(x)
x	abs(x)	$\arcsin(x)$	math.asin(x)
$\ln(x)$; $\lg(x)$; $\log_a x$	math.log(x) ; math.log10(x) ; math.log(x,a)	$\arccos(a)$	math.acos(a)
e^x ; x^y	math.exp(x) ; math.pow(x, y)	$\arctan(x)$	math.atan(x)
n!	math.factorial(n)	$\operatorname{arccot}(x)$	math.atan(1/x)

2. Các hàm thông dụng trong thư viện sympy:

```
import sympy
```

Hàm toán học	Trên Python	Hàm toán học	Trên Python
		$\sin(x)$	sympy.sin(x)
		$\cos(x)$	sympy.cos(x)
		$\tan(x)$	sympy.tan(x)
$\sqrt[2]{x}$	sympy.sqrt(x)	$\cot(x)$	sympy.cot(x)
x	sympy.Abs(x)	$\arcsin(x)$	sympy.asin(x)
$\ln(x)$; $\log_a x$	sympy.log(x) ; sympy.log(x,a)	$\arccos(a)$	sympy.acos(a)
e^x	sympy.exp(x)	$\arctan(x)$	sympy.atan(x)
		$\operatorname{arccot}(x)$	sympy.acot(x)

3. Các hàm thông dụng trong thư viện numpy:

```
import numpy as np
```

Hàm toán học	Trên Python	Hàm toán học	Trên Python
--------------	-------------	--------------	-------------

π	np.pi	sin(x)	np.sin(x)
e	np.e	cos(x)	np.cos(x)
$+\infty ; -\infty$	np.Inf; np.NINF	tan(x)	np.tan(x)
$\sqrt[2]{x}$	np.sqrt(x)	cot(x)	1/np.tan(x)
x	np.absolute (x)	arcsin(x)	np.arcsin(x)
ln(x) ; log_a x	np.log(x) ; np.log(x)/ np.log(a)	arccos(a)	np.arccos(a)
e^x	np.exp(x)	arctan(x)	np.arctan(x)
		arccot(x)	np.arctan(1/x)

VD1. Tính $\sqrt[2]{5,1}$

Cách 1: Dùng math	Cách 2: Dùng sympy	Cách 3: Dùng numpy
<pre>print(math.sqrt(5.1))</pre>	<pre>x = sympy.Symbol('x') y = sympy.lambdify(x, sympy.sqrt(x)) print(y(5.1))</pre>	<pre>print(np.sqrt(5.1))</pre>

VD2. Tính log₃(4)

Cách 1: Dùng math	Cách 2: Dùng sympy	Cách 3: Dùng numpy
<pre>print(math.log(4,3))</pre>	<pre>x = sympy.Symbol('x') y =sympy.lambdify(x, sympy.log(x,3)) print(y(4))</pre>	<pre>print(np.log(4)/np.log(3))</pre>

VD3. Tính arcsin²(1/2)

Cách 1: Dùng math	Cách 2: Dùng sympy	Cách 3: Dùng numpy
<pre>print((math.asin(1/2))**2)</pre>	<pre>x = sympy.Symbol('x') y = sympy.lambdify(x, (sympy.asin(x))**2) print(y(1/2))</pre>	<pre>print((np.arcsin(1/2))**2)</pre>

Chủ đề 2. Ma trận và định thức

- Nhập thư viện:

```
import numpy as np
```

1. Khai báo biến vector và ma trận

VD: Khai báo vector:

Khai báo vector dòng: A = np.array([1, 2, 3]) print(A) <i>###(xem vector A)</i>	Khai báo vector cột: A = np.array([[1], [2], [3]]) print(A) <i>###(xem vector A)</i>
---	--

VD: Khai báo ma trận cỡ 3x3:

```
B = np.array([[11,4,20], [4,9,8], [3,6,9]])
```

```
print(B)
```

2. Các phép toán trên các phần tử của ma trận

Python đang nhớ các biến A và B được khai báo ở trên, ta khai báo thêm biến C:

```
C = np.array([[0,4,17], [-2,5,8], [3.5,8,-9.2]])
```

Ta có thể thực hiện các phép toán cộng (+), trừ (-), nhân (*)

VD: D=B+C

VD: D=B*C *###(lưu ý các phép toán này là cho từng phần tử)*

3. Phép nhân hai ma trận

VD: print(np.dot(B,C))

4. Phép lũy thừa ma trận

VD: print(np.linalg.matrix_power(B, 3)) *(tính B³)*

5. Ma trận chuyển vị

VD: print(B.T)

6. Tìm hạng của ma trận

VD: print(np.linalg.matrix_rank(B))

7. Tìm ma trận nghịch đảo

VD: `print(np.linalg.inv (B))`

8. Tính định thức (của ma trận vuông)

VD: `print(np.linalg.det (B))`

Chủ đề 3. Hệ phương trình tuyến tính

- Nhập thư viện:

```
import numpy as np
```

VD1: Giải hệ PTTT:
$$\begin{cases} x_1 + x_2 + x_3 = 6 \\ x_1 - x_2 = -1; \\ x_1 + x_2 + 2x_3 = 9 \end{cases}$$

Ta thực hiện như sau:	Màn hình hiện kết quả là:
<pre>a = np.array([[1, 1, 1], [1, -1, 0], [1, 1, 2]]) b = np.array([6, -1, 9]) x = np.linalg.solve(a, b) print(x)</pre>	[1. 2. 3.]

Ghi chú: Trong ví dụ này hệ có nghiệm duy nhất: $X=(1, 2, 3)$

VD2: Giải hệ PTTT:
$$\begin{cases} x_1 + x_2 = 6; \\ x_1 + x_2 = 9; \end{cases}$$

Ta thực hiện như sau:	Màn hình hiện kết quả là:
<pre>a = np.array([[1, 1], [1, 1]]) b = np.array([6, 9]) x = np.linalg.solve(a, b) print(x)</pre>	Traceback (most recent call last): numpy.linalg.LinAlgError: Singular matrix

Ghi chú: Trong ví dụ này hệ vô số nghiệm hoặc vô nghiệm.

Chủ đề 4. Đồ thị của hàm số

- Ghi chú: Tham khảo Chủ đề 1 khi cần vẽ các hàm phức tạp.
- Nhập thư viện:

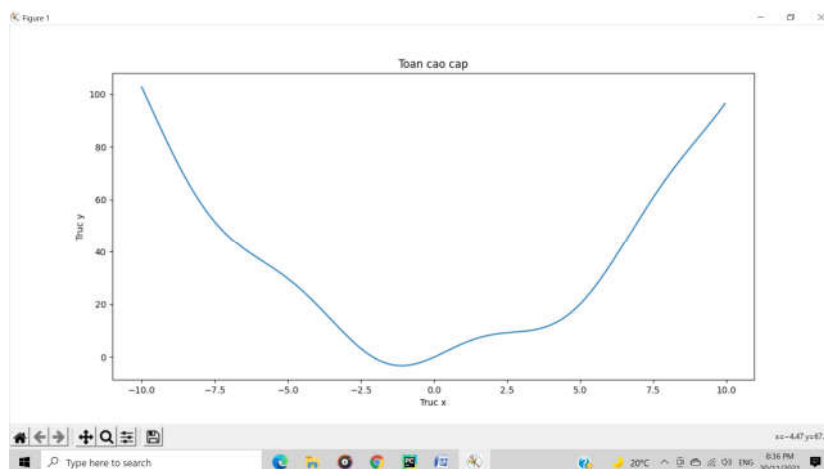
```
import numpy as np
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

1. Hàm 1 biến (2D)

VD: Vẽ đồ thị hàm số: $y = x^2 + 5\sin(x)$ trên đoạn $[-10; 10]$

Thực hiện như sau:

```
x = np.arange(-10, 10, 0.05)
y = x**2 + 5*np.sin(x)
plt.plot(x, y)
plt.xlabel("Trục x")
plt.ylabel("Trục y")
plt.title('Toan cao cap')
plt.show()
```



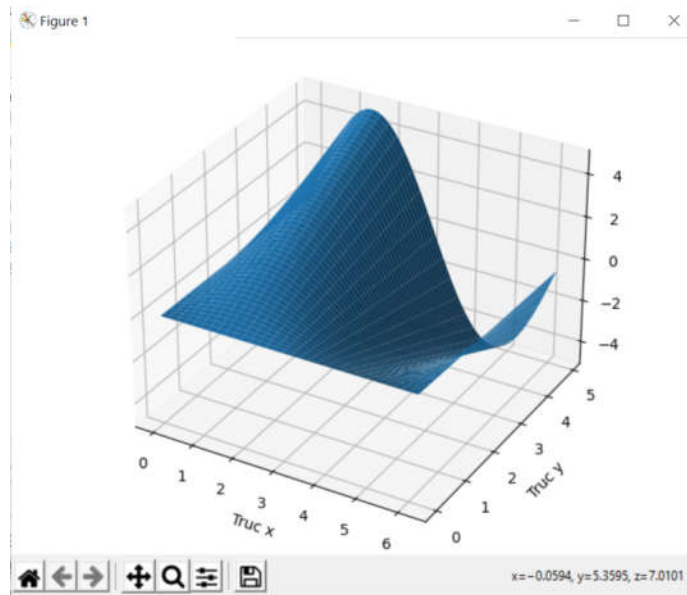
2. Hàm 2 biến (3D)

VD: Vẽ đồ thị hàm $z = \sin(x) * y$ với $x \in [0; 2\pi]$, $y \in [0; 5]$

Thực hiện như sau:

```
x = np.arange(0, 2*np.pi, 0.1)
y = np.arange(0, 5, 0.1)
```

```
X, Y = np.meshgrid(x, y)
z = np.sin(X)*Y
fig = plt.figure()
ax = Axes3D(fig)
ax.plot_surface(X, Y, z)
plt.xlabel("Truc x")
plt.ylabel("Truc y")
plt.title('Toan cao cap')
plt.show()
```



Chủ đề 5. Đạo hàm

- Ghi chú: Tham khảo Chủ đề 1 khi cần tính đạo hàm của các hàm phức tạp.
- Nhập thư viện:

```
import sympy
```

1. Hàm một biến

VD 1: Tính đạo hàm cấp 1 và cấp 2 của hàm số $f(x) = x^3 + \sin(x)$

Đạo hàm cấp 1	Đạo hàm cấp 2
<pre>x = sympy.Symbol('x') y = x**3 + sympy.sin(x) y1 = y.diff(x) print(y1)</pre>	<pre>x = sympy.Symbol('x') y = x**3 + sympy.sin(x) y1 = y.diff(x) y2 = y1.diff(x) print(y2)</pre>
Kết quả: $3*x**2 + \cos(x)$	Kết quả: $6*x - \sin(x)$

VD 2: Tính đạo hàm của hàm số $f(x) = x^3 + \sin(x)$ tại $x = 2$.

```
x = sympy.Symbol('x')
y = x**3 + sympy.sin(x)
y1 = y.diff(x)
y1 = sympy.lambdify(x, y1)
print(y1(2))
```

2. Hàm nhiều biến

VD1: Tính các đạo hàm riêng cấp 1 và cấp 2 của hàm số $z = x^3 + \sin(xy)$

Đạo hàm cấp 1:

z'_x	z'_y
<pre>x, y = sympy.symbols('x y', real=True) z = x**3 + sympy.sin(x*y) z_x = sympy.diff(z, x) print(z_x)</pre>	<pre>x, y = sympy.symbols('x y', real=True) z = x**3 + sympy.sin(x*y) z_y = sympy.diff(z, y) print(z_y)</pre>

Đạo hàm cấp 2:

z''_{xx}	z''_{xy}	z''_{yy}
<pre>x, y = sympy.symbols('x y', real=True) z = x**3 + sympy.sin(x*y)</pre>	<pre>x, y = sympy.symbols('x y', real=True) z = x**3 + sympy.sin(x*y)</pre>	<pre>x, y = sympy.symbols('x y', real=True) z = x**3 + sympy.sin(x*y)</pre>

<pre>z_x = sympy.diff(z, x) z_xx = sympy.diff(z_x, x) print(z_xx)</pre>	<pre>z_x = sympy.diff(z, x) z_xy = sympy.diff(z_x, y) print(z_xy)</pre>	<pre>z_y = sympy.diff(z, y) z_yy = sympy.diff(z_y, y) print(z_yy)</pre>
---	---	---

VD2: Tính các đạo hàm riêng cấp 1 và cấp 2 của hàm số $z = x^3 + \sin(xy)$ tại (1,2)

Đạo hàm cấp 1:

z'_x	z'_y
<pre>x, y = sympy.symbols('x y', real=True) z = x**3 + sympy.sin(x*y) z_x = sympy.diff(z, x) z_x = sympy.lambdify((x, y), z_x) print(z_x(1,2))</pre>	<pre>x, y = sympy.symbols('x y', real=True) z = x**3 + sympy.sin(x*y) z_y = sympy.diff(z, y) z_y = sympy.lambdify((x, y), z_y) print(z_y(1,2))</pre>

Đạo hàm cấp 2:

z''_{xx}	$z''_{xy} = z''_{yx}$	z''_{yy}
<pre>x, y = sympy.symbols('x y', real=True) z = x**3 + sympy.sin(x*y) z_x = sympy.diff(z, x) z_xx = sympy.diff(z_x, x) z_xx = sympy.lambdify((x, y), z_xx) print(z_xx(1,2))</pre>	<pre>x, y = sympy.symbols('x y', real=True) z = x**3 + sympy.sin(x*y) z_x = sympy.diff(z, x) z_xy = sympy.diff(z_x, y) z_xy = sympy.lambdify((x, y), z_xy) print(z_xy(1,2))</pre>	<pre>x, y = sympy.symbols('x y', real=True) z = x**3 + sympy.sin(x*y) z_y = sympy.diff(z, y) z_yy = sympy.diff(z_y, y) z_yy = sympy.lambdify((x, y), z_yy) print(z_yy(1,2))</pre>

Chủ đề 6. Giá trị nhỏ nhất – lớn nhất

- Ghi chú: Tham khảo Chủ đề 1 khi cần tính cho các hàm phức tạp.
- Nhập thư viện:

```
import math

from scipy.optimize import fmin
```

1. Hàm một biến

VD: Tìm giá trị nhỏ nhất, lớn nhất của hàm $y = \sin^2(x) + x^2 + 1$

Nhỏ nhất (min)	Lớn nhất (max)
<pre>def f(x): return (math.sin(x))**2+x**2+1 min_loc = fmin(f, 1.0) print(min_loc) print(f(*min_loc))</pre>	<pre>def f(x): return -((math.sin(x))**2+x**2+1) min_loc = fmin(f, 1.0) print(min_loc) print(-f(*min_loc))</pre>
<p>Kết quả:</p> <p>Optimization terminated successfully.</p> <p>Current function value: 1.000000</p> <p>Iterations: 17</p> <p>Function evaluations: 34</p> <p>[-8.8817842e-16]</p> <p>1.0</p>	<p>Kết quả:</p> <p>Warning: Maximum number of function evaluations has been exceeded.</p> <p>[6.338253e+28]</p> <p>4.0173451106474974e+57</p>
<p><i>Ghi chú.</i> Hàm số đạt min tại $x = -8.8817842e-16 \approx 0$</p> <p>Giá trị min: 1.0</p>	<p><i>Ghi chú.</i> Giá trị max vượt quá giá trị lớn nhất của Python (hàm số tiến ra vô cùng)</p>

2. Hàm nhiều biến

VD: Tìm giá trị nhỏ nhất, lớn nhất của hàm $f = x^2 + (y-1)^2$

Nhỏ nhất (min)	Lớn nhất (max)
<pre>def f(x,y): return x**2 + (y-1)**2 min_loc = fmin(lambda vec: f(vec[0],</pre>	<pre>def f(x,y): return -(x**2 + (y-1)**2) min_loc = fmin(lambda vec: f(vec[0],</pre>

<pre>vec[1]), [0.1, 0.1]) print(min_loc) print(f(*min_loc))</pre>	<pre>vec[1]), [0.1, 0.1]) print(min_loc) print(-f(*min_loc))</pre>
<p>Kết quả:</p> <p>Optimization terminated successfully.</p> <p>Current function value: 0.000000</p> <p>Iterations: 51</p> <p>Function evaluations: 94</p> <p>[-1.03073585e-05 9.99952728e-01]</p> <p>2.340897791033299e-09</p>	<p>Kết quả:</p> <p>Warning: Maximum number of function evaluations has been exceeded.</p> <p>[1.02325529e+43 -1.21372471e+43]</p> <p>2.520179052869857e+86</p>
<p><i>Ghi chú.</i> Hàm số đạt min tại $(x, y) = (-1.03073585e-05, 9.99952728e-01) \approx (0, 1)$</p> <p>Giá trị min: $2.340897791033299e-09 \approx 0$</p>	<p><i>Ghi chú.</i> Giá trị max vượt quá giá trị lớn nhất của Python (hàm số tiến ra vô cùng)</p>

Chủ đề 7. Các tham số đặc trưng

- Nhập thư viện:

```
import numpy as np
```

VD: Quan sát chiều cao X (cm) của 10 sinh viên, ta được :

[172, 154, 165, 165, 160, 165, 160, 155, 170, 160.]

Tính kỳ vọng, phương sai và độ lệch chuẩn.

```
X = [172, 154, 165, 165, 160, 165, 160, 155, 170, 160]
print(np.mean(data))
print(np.var(data))
print(np.std(data))
```


Phụ lục: Download, cài đặt và chạy “Hello World”

1. Download và cài đặt

Để lập trình python cần cài đặt: python và PyCharm (Pycharm là một công cụ để soạn thảo và chạy python (IDE), có thể sử dụng IDE khác hoặc không cần dùng IDE)

Bước 1:

@ Download python tại: <https://www.python.org/downloads/>

@ Cài đặt như các phần mềm khác

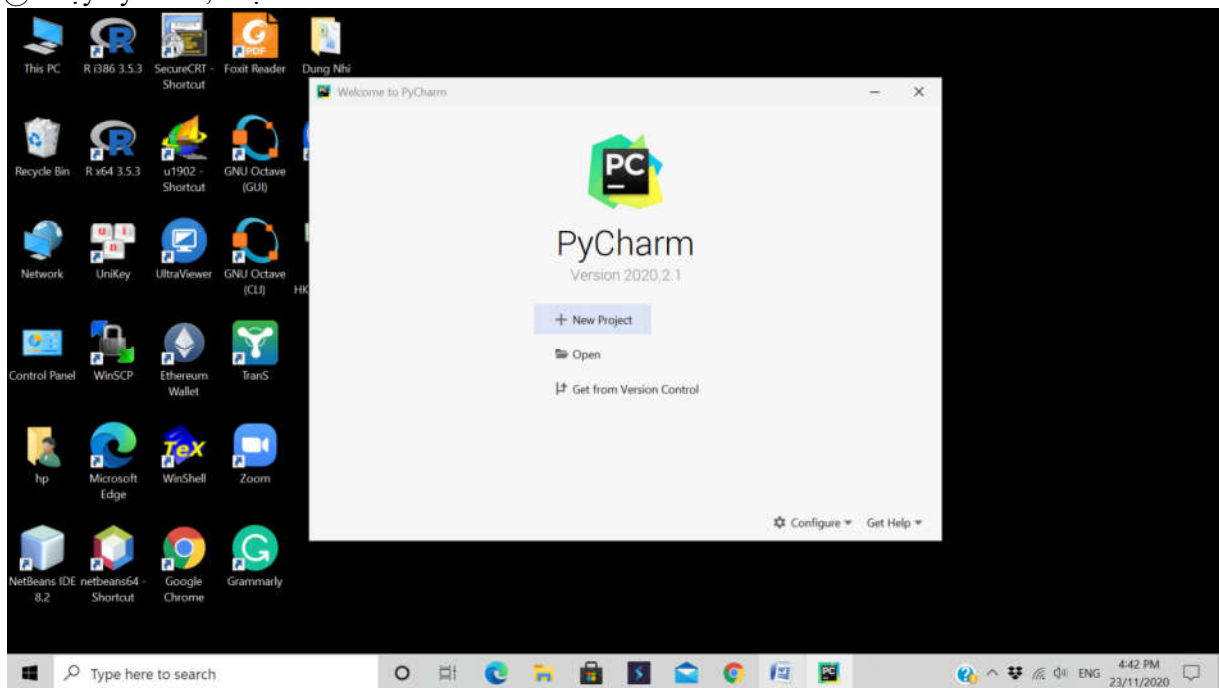
Bước 2:

@ Download PyCharm tại: <https://www.jetbrains.com/pycharm/download/>
(chọn bản Community)

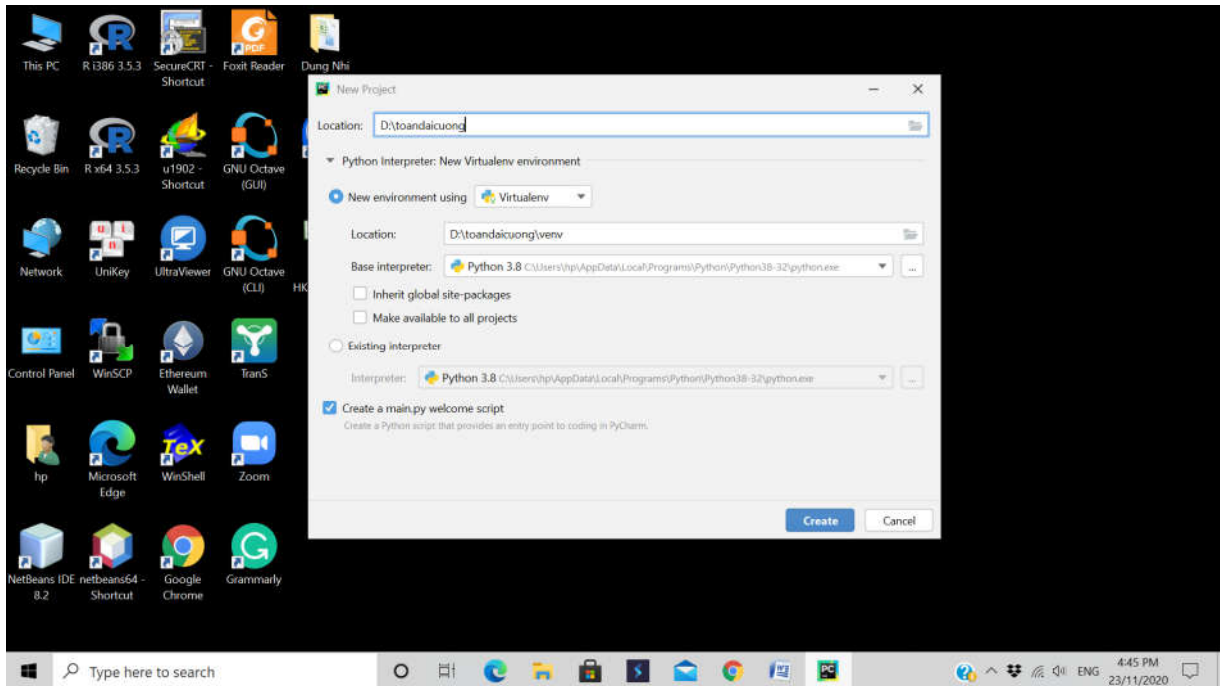
@ Cài đặt như các phần mềm khác

2. Chạy “Hello World”

@ Chạy Pycharm, được cửa sổ:



@ click vào **+ New Project**; Tại **Location** gõ: D:\toandaicuong; click nút **Create**

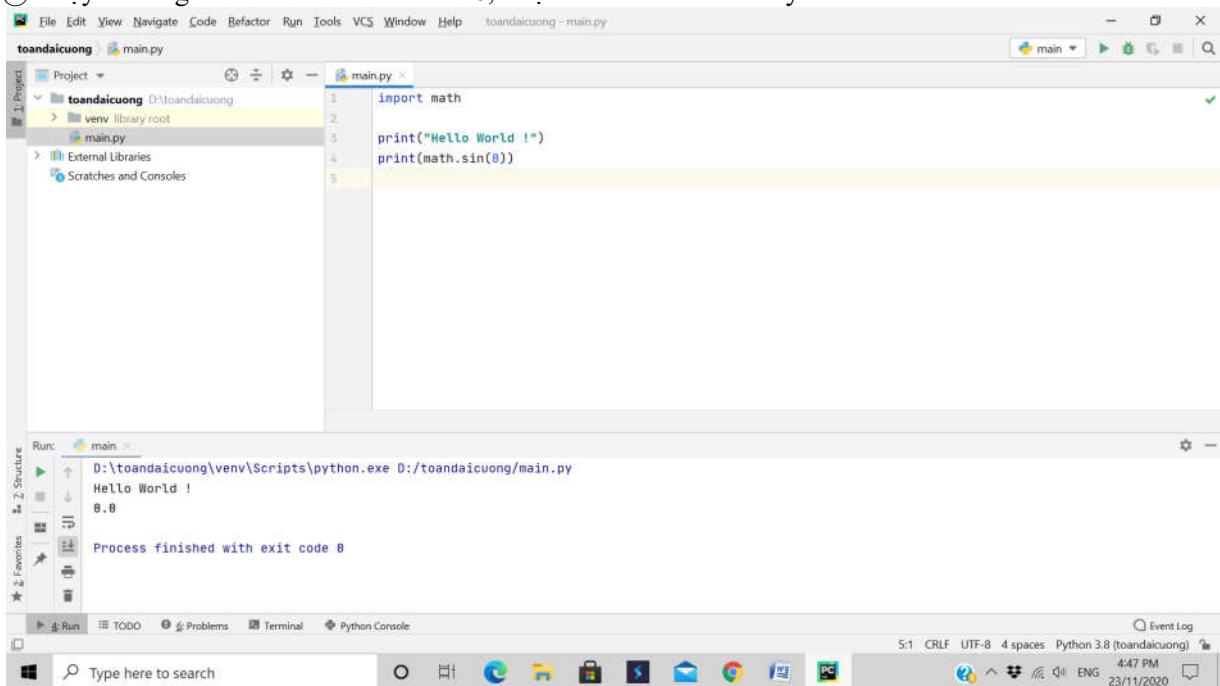


@ Trong cửa sổ bên phải, đây chính là file main.py; xóa hết các câu lệnh trong file này, gõ các câu lệnh:

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

```
print(math.sin(0))
```

@ Chạy chương trình: nhấn nút Shift+F10, được cửa sổ như dưới đây



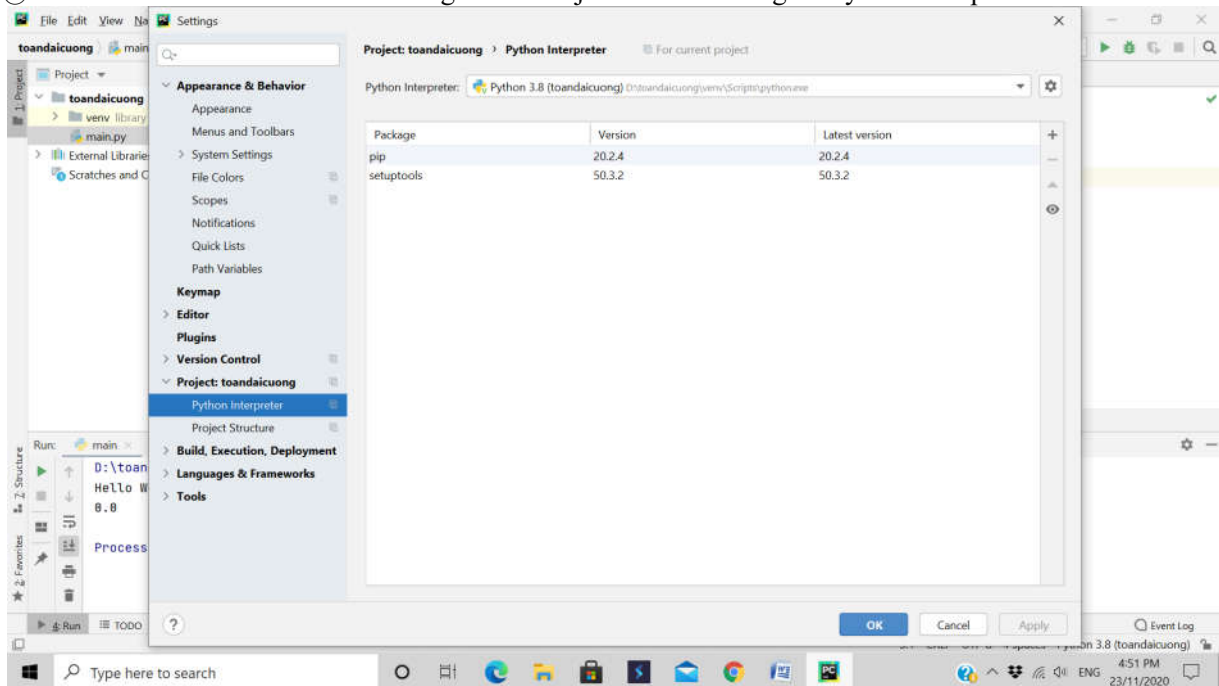
Kết luận: Bạn đã chạy thành công “Hello World”

3. Cài đặt thư viện

Để chạy được hết các ví dụ trong tài liệu này, cần phải cài đặt các thư viện: numpy, scipy, sympy, matplotlib

Mô tả cài numpy (các thư viện khác cài tương tự)

@ Trên thanh Menu vào File → Settings... → Project: toandaicuong → Python Interpreter



@ click vào dấu **+** ở bên phải; gõ numpy và click **Install Package**

