LÀM QUEN VỚI PYTHON

1. Mục tiêu

Tài liệu này là để hướng dẫn sinh viên cài đặt môi trường làm việc với ngôn ngữ lập trình Python và viết code với IPython Notebook.

Chuẩn bị môi trường:

* -  OS: Windows hoặc Ubuntu
* -  Anaconda2 with Python 2.7
* -  Editor: Notepad++ hoặc Sublime Text

Cài đặt Anaconda2

Vào trang *https://www.continuum.io/downloads* và tải về bản Anaconda2 phù hợp với hệ điều hành đang sử dụng, lưu ý phải chọn bản Python 2.7.

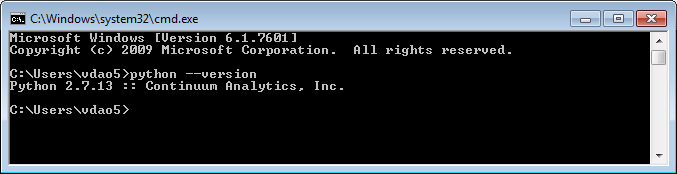
Sau khi hoàn tất, chúng ta có thể kiểm tra version python bằng cách mở cửa sổ command line và thực hiện câu lệnh sau:

***python –version <Enter>***

2. Chươngtrìnhđầutiên a. Hello World

Chúng ta sẽ viết và chạy chương trình “Hello World” từ màn hình dòng lệnh. Từ màn hình command line gõ lệnh ***python*** <Enter>

C:\> python



Sau đó từ dấu nhắc >>>\_ gõ lệnh print(*‘Hello World!’*) <Enter>, kết quả sẽ như sau:

Để kết thúc phiên làm việc python, gõ lệnh ***exit()*** <Enter>

Như minh họa ở trên, python cho phép chúng ta gõ lệnh trực tiếp vào cửa sổ command line và thực hiện từng dòng lệnh một. Tuy nhiên, trong đa số trường hợp, ta sẽ đặt các lệnh python vào các file \*.py, đây là file mã nguồn (source code) của chương trình python.

Mở trình soạn thảo (Notepad++) và gõ vào nội dung như sau:

print(*‘Hello World’*)  
Lưu tập tin trên với tên hello.py, quay trở lại màn hình command line

thực hiện lệnh sau:

python hello.py <Enter> Kết quả:

Hello World!

3. Các phép toán

Cácphép toánsốhọc:+-\*/%  
Các phép toán luận lý: ==, !=, or, and, not, &, |

Giá trị luận lý có giá trị True hoặc False

|  |
| --- |
| >>> print(‘Hello World!’) Hello World! |
| >>> |

|  |
| --- |
| >>> 1 + 1 2 |
| >>> 2 \* 3 6 |

|  |
| --- |
| >>> 1==0 False >>> not (1==0) True |
| >>> (2==2) and (2==3) False >>> (2==2) or (2==3) True |

4. Sử dụng biến và kiểu dữ liệu

Biến là một vùng nhớ chứa dữ liệu, với Python bạn có thể khai báo và sử dụng biến như sau:

Note: bạn có thể sử dụng hàm type(varA) để biết xem biến varA có kiểu là gì, ví dụ:

Bạn cũng có thể dùng lệnh dir(varA) để liệt kê các hàm mà mình có thể thao tác với biến tương ứng, ví dụ sau sẽ cho biết các hàm có thể gọi với kiểu dữ liệu float.

5. Làm việc với chuỗi

Tạo một chuỗi sử dụng ký tự định dạng:

Kết quả in ra là:

Gia tri cua so pi la 3.1415900

6. Cáccấutrúcđiềukhiển  
a. Cấu trúc if .. elif .. else

|  |
| --- |
| a = 10 x = 1.25 |
| s = “Hello world!” print “a = ”, a |
| print “x = ”, x print “Say: ”,s |

|  |
| --- |
| A = 3.5 type(A) |
| <type ‘float’> |

|  |
| --- |
| A = 3.5 |
| dir(A) [‘\_abs\_’, ‘\_class\_’, .... ] |

|  |
| --- |
| >>> 'artificial' + "intelligence" |
| 'artificialintelligence' |

pi = 3.14159  
str = “Gia tri cua so pi la %f ” % (pi) **print** str

page3image2563090208

if exp1: statement1 statement2

page3image2563094144

statement 3

Với cấu trúc if ở trên, exp1 là biểu thức luận lý (True/False), các bạn phải chú ý sau biểu thức luận lý là dấu 2 chấm, nếu exp1 có giá trị True thì 2 câu lệnh statement1 và statement2 sẽ được thực hiện. Xin chú ý là không sử dụng dấu mở và đóng khối lệnh đối với cấu trúc của Python, thay vào đó, chúng ta phải thụt vào đầu dòng 2 dấu cách (space), cấu lệnh statement3 trong ví dụ trên không thuộc khối if do không indent 2 dấu cách.

Nếu cấu trúc của chúng ta có nhiều nhánh thì sử dụng cấu trúc if đầy đủ như cú pháp trên.

Ví dụ:

|  |
| --- |
| if exp1: |
| statement1  statement2 |
| elif exp2: statement3  else: |
| statement4 |

|  |
| --- |
| a = **input**("Nhập số nguyên a: ") b = **input**("Nhập số nguyên b: ") |
| **if** a>b: **print** "a lon hon b " |
| **elif** a==b: **print** "a bang b"  **else**: **print** "a nho hon b" |

b. Cấu trúc while

Ví dụ:

c. Cấu trúc for

|  |
| --- |
| while expression: |
| statement(s) |

|  |
| --- |
| x = "Hello World!" |
| i=0 **while** i < 10:  **print** x, " ", i i += 1 |

In chuỗi “Hello” 3 lần ra màn hình:

7. Một số cấu trúc dữ liệu tập hợp a. List

Khai báo list:

**x[2:4]** là phép toán lấy danh sách con của danh sách x từ phần từ thứ 0 đến phần tử thứ 3 (tính từ 0, tức là 4 - 1).

Trong ví dụ sau, ta sẽ lấy giá trị tương ứng của phần tử trong một list, mỗi phần tử trong list này là một cặp (name:value).

Một số hàm thông dụng làm việc với List

|  |
| --- |
| for i in range(0,3): |
| print “Hello” |
| Hello Hello |
| Hello |

|  |
| --- |
| x = [-3, 0, 3, 9, 7] s = [“car”,“bike”, “bus”, “plane”] |
| print x[3] 9 |
| print x[2:4] [3, 9] |

|  |
| --- |
| lst = [‘Honda’:29, ‘Suzuki’:16, ‘Toyota’:38] print lst[‘Honda’] for itm in lst: |
| print itm[1] 29  16 38 |

|  |  |  |
| --- | --- | --- |
| Tên hàm  page5image2563212144page5image2563259376page5image2563260736page5image2563261248 | Ý nghĩa | Ví dụ  page5image2563265648 |
| len(a)  page5image2563270432page5image2563271008 | Trả về số phần tử trong list a | a = [9, 3, 5, 7] print len(a) 4  page5image2563281408 |
| cmp(lst1, lst2)  page5image2564413552page5image2564416208 | So sánh phần tử giữ 2 list lst1 và lst2 Trả về: -1 nếu lst1 < lst2  0 nếu lst1 = lst2 1 nếu lst1 > lst2 | A = [4, 5, -1] B = [2, 9, 0] cmp(A, B) 1  page5image2564422496page5image2564423072 |

page6image2564585968

Bạn có thể dùng lệnh dir(list) để liệt kê các hàm của kiểu dữ liệu List. b. Tubes

Kiểu cấu trúc này tương tự như List ngoại trừ bạn không thể thay đổi được giá trị của các phần tử trong một tube.

page6image2564571936page6image2564560224

|  |
| --- |
| >>> pair = (3,5) |
| >>> pair[0] 3 |
| >>> x,y = pair >>> x |
| 3 >>> y 5 |
| >>> pair[1] = 6 TypeError: object does not support item assignment |

c. Sets

Set là cấu trúc tập hợp không thứ tự và các phần không trùng nhau.

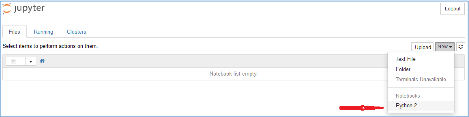
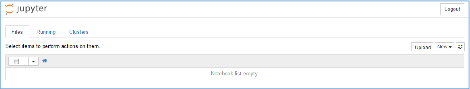
8. LậptrìnhPythonvớiIPython

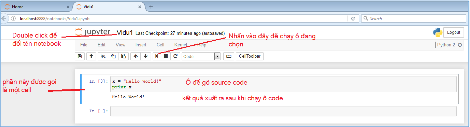
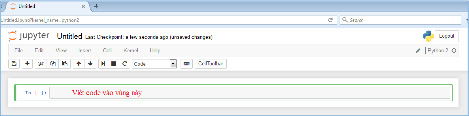
Từ màn hình console gõ lệnh sau:

jupyter notebook

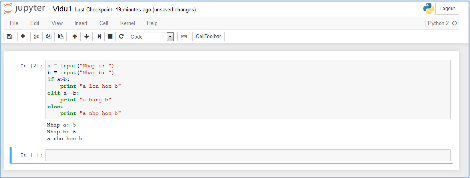
Giao diện jupyter notebook sẽ được load lên giao diện web như sau:

Để tạo mới một notebook ta chọn menu New ở phía bên phải và chọn “Python 2”





Ví dụ:



Bài tập nộp trên Moodle

Chú ý: Với bài tập 1, 2, 3 các file source code \*.py đã được chuẩn bị sẵn trong file tutorial.zip, sinh viên chỉ cần thêm code của mình vào các vị trí "\*\*\* YOUR CODE HERE \*\*\*" để hoàn thành bài làm.

1. File addition.py có nội dung như sau:

def add(a, b):

*"Return the sum of a and b"*

**"\*\*\* YOUR CODE HERE \*\*\*"**

return 0

page8image2563192064page8image2563192464

Sinh viên hãy thêm code vào vị trí **"\*\*\* YOUR CODE HERE \*\*\*"** để hàm add trảvề tổng của avà b.

2. FilebuyLotsOfFruit.pycónộidungnhưsau:

|  |
| --- |
| page8image2563166096page8image2564957200  fruitPrices = {'apples':2.00, 'oranges': 1.50, |
| 'pears': 1.75,  'limes':0.75, 'strawberries':1.00} |
| def buyLotsOfFruit(orderList): *"""*  *orderList: List of (fruit, numPounds) tuples* |
| *Returns cost of order* |
| *"""*  totalCost = 0.0  **"\*\*\* YOUR CODE HERE \*\*\*"**  return totalCost |
| *# Main Method*  if \_\_name\_\_ == '\_\_main\_\_': |
| "This code runs when you invoke the script from the command line"  orderList = [ ('apples', 2.0), ('pears', 3.0), |
| ('limes', 4.0) ]  print 'Cost of', orderList, 'is', buyLotsOfFruit(orderList)  page8image2566143952page8image2566144896 |

Sinh viên hãy thêm code tại vị trí **“\*\*\* YOUR CODE HERE \*\*\*”** để hàm byLotsOfFruit trả về tổng số tiền của danh sách truyền vào. Chú ý là tham số truyền vào là một danh sách gồm các cặp (fruit, <giá>).

3. Sinh viên hãy thêm code vào file shopSmart.py tại vị trí **“\*\*\* YOUR CODE HERE \*\*\*”** trong hàm shopSmart(orders, shops) để trả về FruitShop sao cho giá trị của hóa đơn là nhỏ nhất.

Tham số của hàm shopSmart(orders, shops) như sau:  
Orders: danh sách trái cây cùng với số lượng mà bạn muốn đặt. Shops: dánh sách các shop có bán các loại trái cây này  
Hàm trả về shop có tổng giá trị giá bán là thấp nhất.

Ví dụ, với dữ liệu như sau:

|  |
| --- |
| orders1 = [('apples',1.0), ('oranges',3.0)] orders2 = [('apples',3.0)] dir1 = {'apples': 2.0, 'oranges':1.0} |
| shop1 = shop.FruitShop('shop1',dir1) dir2 = {'apples': 1.0, 'oranges': 5.0} |
| shop2 = shop.FruitShop('shop2',dir2) shops = [shop1, shop2] |

Với orders1 thì shop1 sẽ được chọn mua, với orders2 thì shop2 sẽ được chọn mua.

page9image2563966832

shopSmart.shopSmart(orders1, shops) == shop1 shopSmart.shopSmart(orders2, shops) == shop2