



IME-USP

SPYDER

um “ambiente de desenvolvimento integrado”
(IDE*) para Python

Prof. Carlos Hitoshi Morimoto
DCC – IME – USP
Março, 2016

* IDE = Integrated Development Environment



Esta obra está licenciada com uma licença [Creative Commons Atribuição 4.0 Internacional](http://creativecommons.org/licenses/by/4.0/).
Para ver uma cópia da licença visite: <http://creativecommons.org/licenses/by/4.0/>.



IME-USP

Pré-requisitos

Embora não 100% necessário, recomendamos que você já tenha alguma experiência com Python para entender melhor essa aula. Para isso, antes de prosseguir faça os exercícios online das aulas 1 a 5 em:

[Aulas de Introdução à Computação em Python](#)



IME-USP

Por que “ambiente”?

- O desenvolvimento de um programa requer o uso de várias ferramentas como:
 - um “editor de texto” para escrever o programa fonte
 - um “interpretador Python” para “rodar” o programa
 - um “terminal” onde o programa é “rodado” e permite a entrada e saída dos dados
 - um “depurador de programas”, que te ajuda a encontrar erros
 - etc.
- O IDE (Spyder) permite que você trabalhe em um só ambiente e não fique trocando (entrando e saindo) de uma ferramenta para outra.
- Outros possíveis ambientes gratuitos para o desenvolvimento de programas em Python são o IDLE e o Eclipse (usando PyDev)



IME-USP

Como instalar o spyder

- O spyder vem com o Anaconda

<https://www.continuum.io/downloads>

se você já instalou o Anaconda, pode prosseguir. Caso contrário, lembre-se de instalar a versão para o Python 3.x

- Há também outras formas de instalar

<https://pythonhosted.org/spyder/installation.html>



IME-USP

Como chamar o spyder

- Em vários sistemas operacionais como Windows, OS X, e Linux, você pode chamar o spyder de um terminal como abaixo:

```
hitoshi@xps: ~  
hitoshi@xps:~$ spyder
```

The image shows a terminal window with a dark background. The title bar at the top reads 'hitoshi@xps: ~'. The prompt 'hitoshi@xps:~\$' is followed by the command 'spyder' and a cursor. A mouse cursor is visible in the lower-left area of the terminal.



Tela inicial do spyder

The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script with the following content:

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8
```

The IPython console is open, showing the following output:

```
Python 3.5.1 |Anaconda 2.5.0 (64-bit)| (default, Dec 7 2015, 11:16:01)
Type "copyright", "credits" or "license" for more information.

IPython 4.0.3 -- An enhanced Interactive Python.
?      -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help   -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
%gui?  -> A brief reference about the graphical user interface.

In [1]:
```

The Object Inspector and Variable Explorer are also visible, showing a "Usage" dialog box with the following text:

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Object Inspector*.

New to Spyder? Read our [tutorial](#)



Partes Principais da Interface

IME-USP

Barra de Ferramentas

Explore a barra para

- criar um novo arquivo
- abrir um arquivo
- salvar os arquivos
- depurar o programa

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8
```

Usage

Here you can get help of any object by pressing **Ctrl+H** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Object Inspector*.

New to Spyder? Read our [tutorial](#)

Object inspector Variable explorer File explorer

IPython console

```
Python 3.5.1 [Anaconda 2.5.0 (64-bit)] (default, Dec 7 2015, 11:16:01)
Type "copyright", "credits" or "license" for more information.

IPython 4.0.3 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
%gui?     -> A brief reference about the graphical user interface.

In [1]:
```

Console History log IPython console

Permissions: RW End-of-lines: LF Encoding: UTF-8 Line: 1 Column: 1 Memory: 24 %



Partes Principais da Interface

IME-USP

Janelas de edição dos programas.
O tab mostra o nome do arquivo.
Você pode editar vários arquivos (células) ao mesmo tempo

```
1 """
2 """
3 Spyder Editor
4 """
5 This is a temporary script file.
6 """
7
8
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Object Inspector*.

New to Spyder? Read our [tutorial](#)

Object inspector Variable explorer File explorer

IPython console

Console 1/A

Python 3.5.1 |Anaconda 2.5.0 (64-bit)| (default, Dec 7 2015, 11:16:01)
Type "copyright", "credits" or "license" for more information.

IPython 4.0.3 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
%gui? -> A brief reference about the graphical user interface.

In [1]:

Permissions: RW End-of-lines: LF Encoding: UTF-8 Line: 1 Column: 1 Memory: 24 %



Partes Principais da Interface

IME-USP

The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script named 'temp.py' with the following content:

```
1 # -*- coding: utf-8 -*-  
2 """  
3 Spyder Editor  
4  
5 This is a temporary script file.  
6 """  
7  
8
```

Annotations include:

- A green box on the left contains the text: "Clique na aba desejada: - Object Inspector - Variable Explorer - File Explorer". An arrow points from this box to the tabs of the IPython console.
- A large green oval highlights the Object Inspector panel, which contains the text: "Área para inspeção de objetos, variáveis e arquivos".

The IPython console shows the following output:

```
Python 3.5.1 |Anaconda 2.5.0 (64-bit)| (default, Dec 7 2015, 11:16:01)  
Type "copyright", "credits" or "license" for more information.  
  
IPython 4.0.3 -- An enhanced Interactive Python.  
?          -> Introduction and overview of IPython's features.  
%quickref  -> Quick reference.  
help       -> Python's own help system.  
object?    -> Details about 'object', use 'object??' for extra details.  
%gui?      -> A brief reference about the graphical user interface.  
  
In [1]:
```



Partes Principais da Interface

IME-USP

File Edit Search Source Run Debug Consoles Tools View Help

Editor - /home/hitoshi/.spyder2-py3/temp.py

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8
```

Object inspector

Source Console Object

Usage

Here you can get help of any object by pressing Ctrl+H in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Object Inspector*.

New to Spyder? Read our [tutorial](#)

Object inspector Variable explorer File explorer

IPython console

Python 3.5.1 |Anaconda 2.5.0 (64-bit)| (default, Dec 7 2015, 11:16:01)
Type "copyright", "credits" or "license" for more information.

IPython 4.0.3 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
%gui? -> A brief reference about the graphical user interface.

In [1]:

Console History log IPython console

Permissions: RM End-of-lines: LF Encoding: UTF-8 Line: 1 Column: 1 Memory: 24 %

Clique na aba
- iPython console
para interagir com o
seu programa

Terminal de
entrada e saída



IME-USP

Exercício

- Para aprender a usar o spyder, vamos fazer o exercício da aula 6 sobre funções contida em:

<https://panda.ime.usp.br/aulasPython/static/aulasPython>

- Antes de continuar:

- use o seu navegador para abrir a página da aula:

<https://panda.ime.usp.br/aulasPython/static/aulasPython/aula06.html>

- abra o spyder e crie um novo arquivo, por exemplo, com o nome “funcoes.py”
- para criar o arquivo, você pode também clicar no menu
file → new file (ou simplesmente Ctrl N)



IME-USP

Edite o seu programa

- Na janela de edição escreva o seu programa.

- Você pode baixar uma solução de

<https://panda.ime.usp.br/aulasPython/static/aulasPython/exercicios/ex64.html>

- Salve o programa
file → save (ou Ctrl+S)
- Execute o programa
– run → run (ou F5)

Salvar Executar

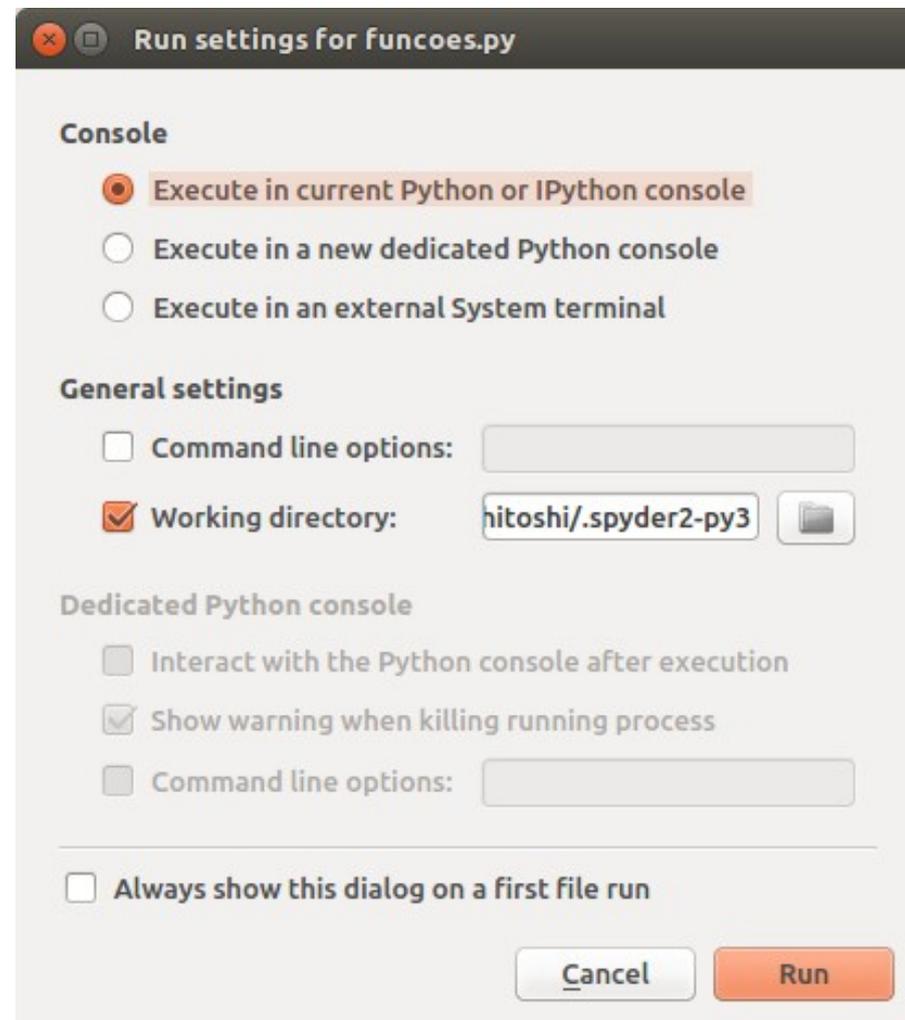
```
File Edit Search Source Run Debug Consoles Tools View Help
Editor - /home/hitoshi/.spyder2-py3/funcoes.py
funcoes.py
1 #-*- coding: utf-8 -*-
2 '''
3 Programa que le um inteiro n, n>=0, e imprime
4 os coeficientes da expansao de (x+y) elevado a n.
5 '''
6 def main():
7     # leitura do n
8     n = int(input("Digite n: "))
9
10    cont = 0
11    while cont <= n:
12        print("Coeficiente de x^%d y^%d: %d"%(n-cont, cont, combinacao(n, cont)))
13        cont += 1
14
15 def fatorial(k):
16     '''(int) -> int
17
18 Recebe um inteiro k e retorna o valor de k!
19
20 Pre-condicao: supoe que k eh um numero inteiro nao negativo.
21 '''
22
23    k_fat = 1
24    cont = 1
25    while cont < k:
26        cont += 1 # o mesmo que cont = cont + 1
27        k_fat *= cont # o mesmo que k_fat = k_fat * cont
28
29    return k_fat
30
31 def combinacao(m, n):
32     '''(int, int) -> int
33 Recebe dois inteiros m e n, e retorna o valor de m!/((m-n)! n!)
34 '''
35
36    return fatorial(m)/(fatorial(n)*fatorial(m-n))
37
38 # chamar a funcao main
39 main()
40
41
```



Execute o programa

IME-USP

- Ao executar o programa, você precisa escolher se ele deve ser executado:
 - no console (terminal) atual, ou
 - em um novo terminal dedicado.
- Durante o desenvolvimento do programa, execute no console atual, pois facilita o teste.



ao ser executado no console atual...



IME-USP

- na janela do console, digite o valor de n desejado (por exemplo, 5) e veja a saída do programa.
- No próprio console, você pode explorar as demais funções, como `fatorial(3)`.

```
IPython console
IP: Console 1/A x
quitkref -> quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
%guieref  -> A brief reference about the graphical user interface.

In [1]: runfile('/home/hitoshi/.spyder2-py3/funcoes.py', wdir='/home/hitoshi/.spyder2-py3')

Digite n: 5
Coeficiente de x^5 y^0: 1
Coeficiente de x^4 y^1: 5
Coeficiente de x^3 y^2: 10
Coeficiente de x^2 y^3: 10
Coeficiente de x^1 y^4: 5
Coeficiente de x^0 y^5: 1

In [2]: fatorial(3)
Out[2]: 6

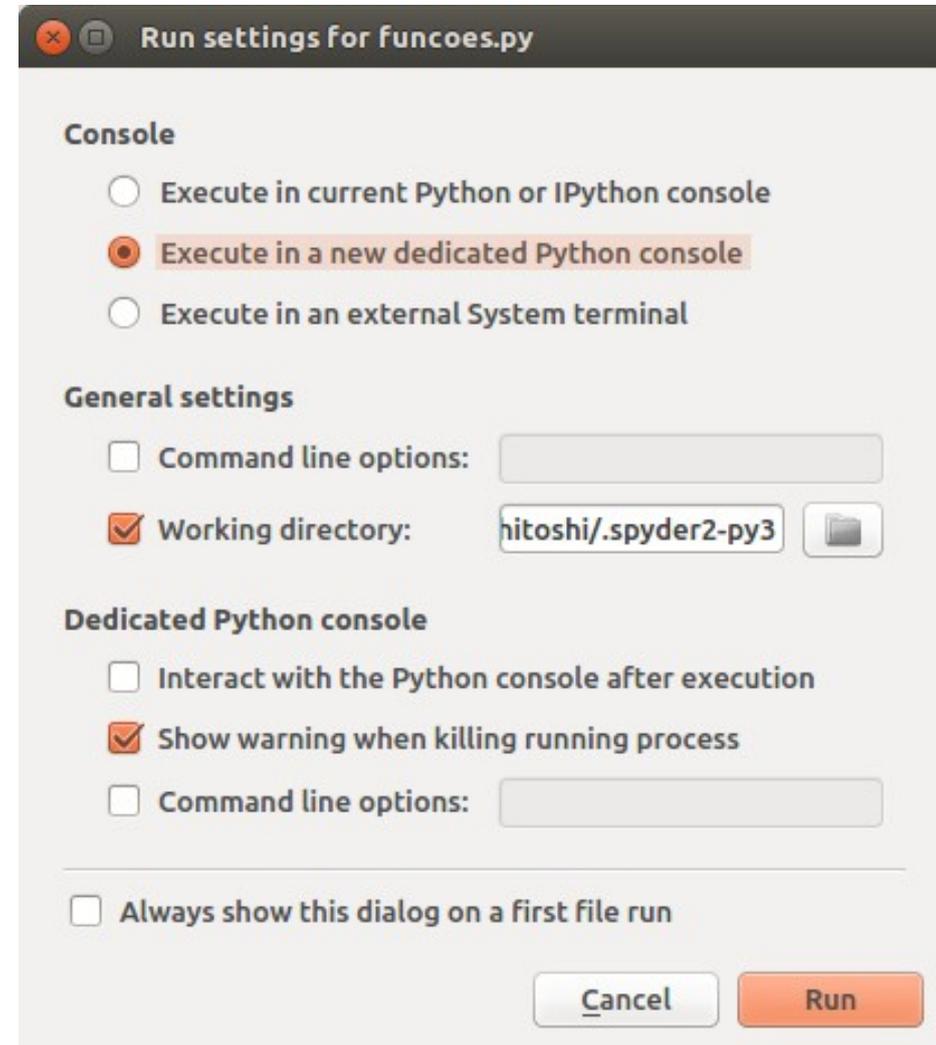
In [3]:
```



IME-USP

Mas antes de entregar um exercício...

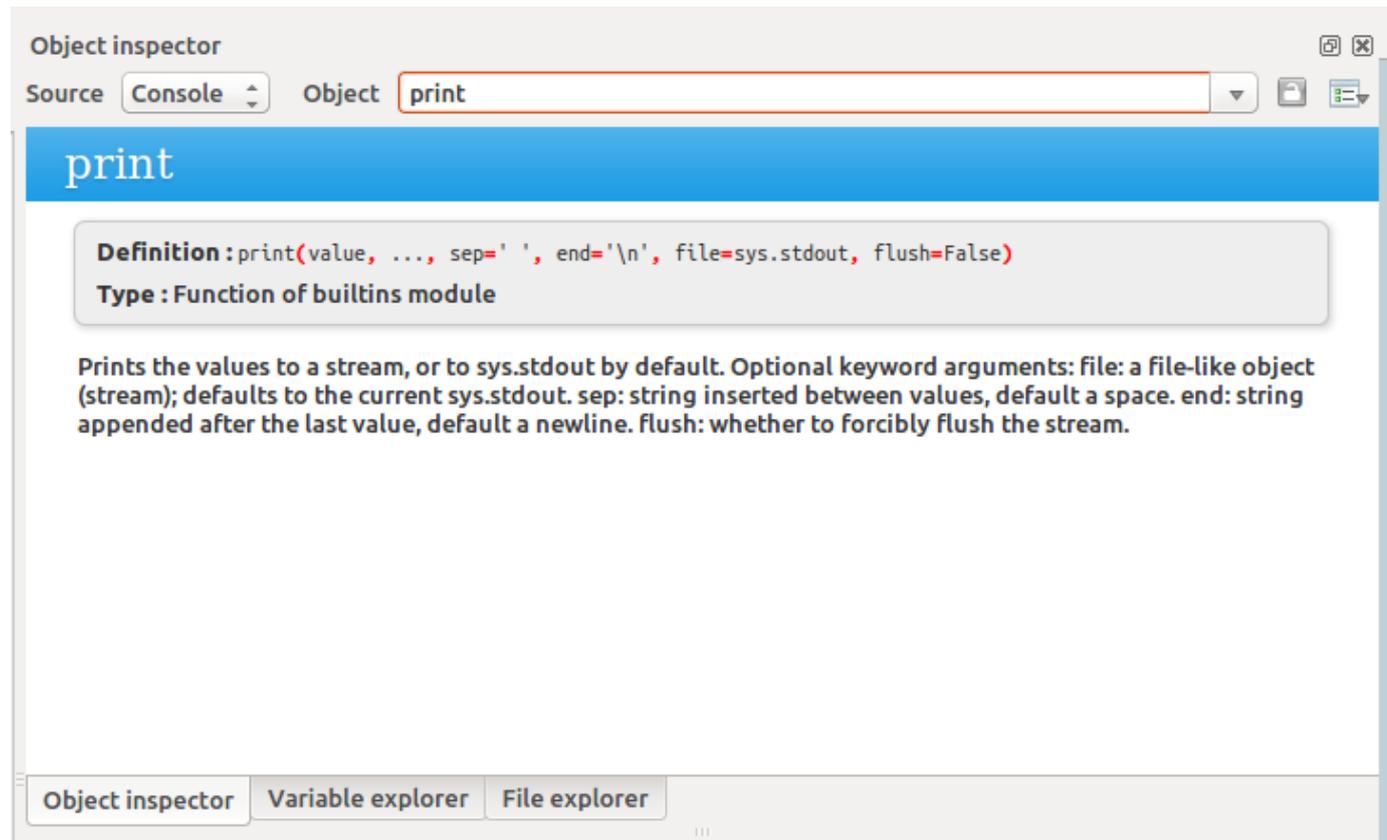
- Certifique-se que o seu exercício rode independentemente do estado do seu console. Para isso clique em:
run → configure (ou F6)
- e execute o programa em um novo console, como mostrado na figura ao lado.





Próximos passos

- Explore as áreas de inspeção. Por exemplo, clique na aba “object inspector” e na linha “object” clique o nome de uma função do Python:



Experimente mudar os parâmetros **sep** e **end** para ver o que acontece. Por exemplo:
`print("ola", "mundo", sep=" ", end=" ")`



- Experimente também os recursos de depuração (Debug)
 - coloque “breakpoints” (pontos de parada) em seu programa. Por exemplo dentro de uma função. para isso:
 - clique em uma linha e aperte F12. A linha deve receber um ponto vermelho. Apertando F12 novamente, o breakpoint é desligado.
 - clique na aba de “variable explorer” para ver o conteúdo da variáveis
 - clique no botão “Debug file” na barra de ferramentas (Ctrl+F5)
 - você pode rodar linha a linha (Ctrl+F10)
 - observe que, a cada linha executada, o valor das variáveis e atualizado automaticamente
 - ou rodar até o próximo breakpoint (Ctrl+F12)



IME-USP

Para saber mais

- Consulte a documentação do
 - Python
<https://docs.python.org/3/>
 - spyder
<https://pythonhosted.org/spyder/>