

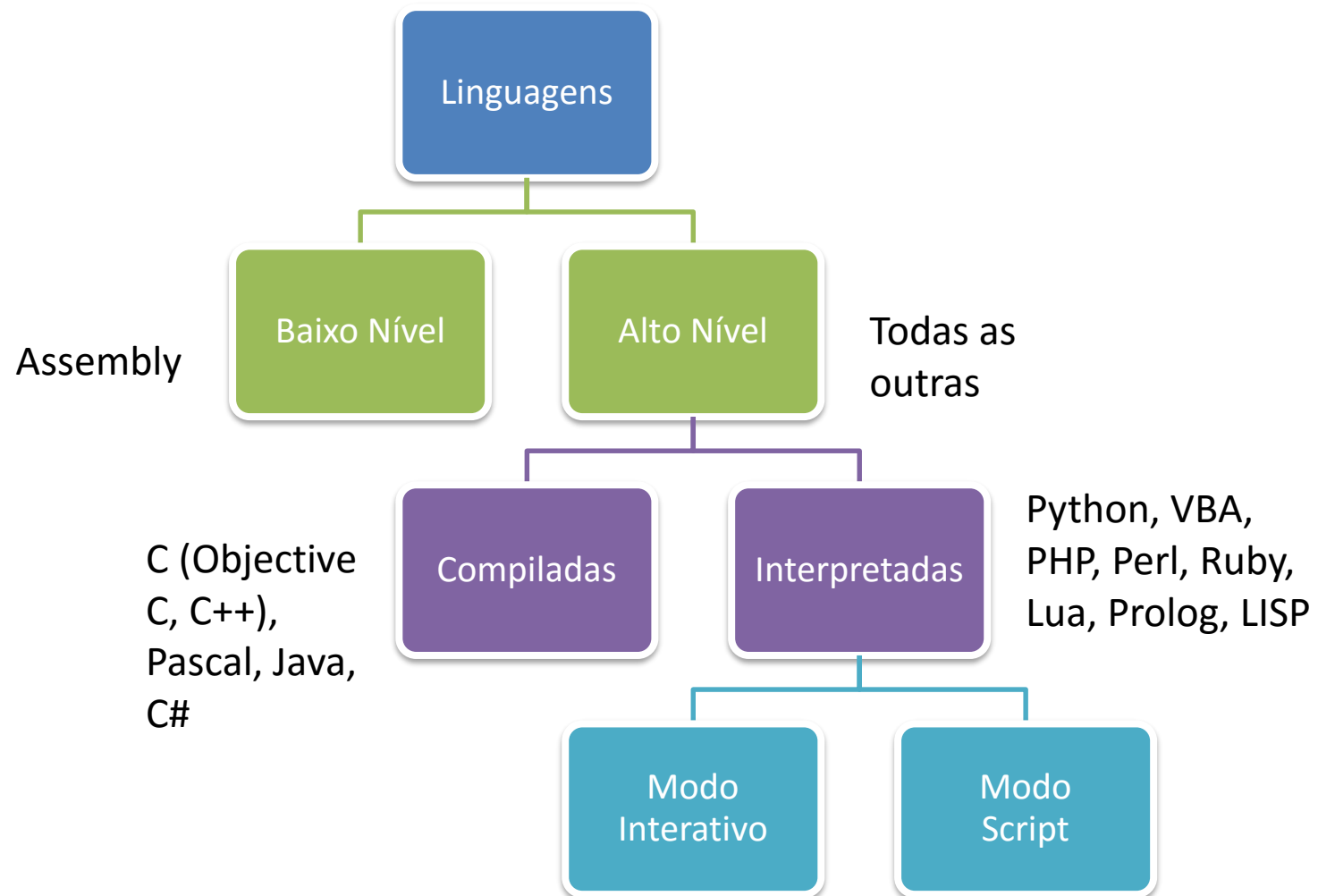
# Design de Software

Aula 2 – Cálculos, variáveis e I/O

2016 – Engenharia

Raul Ikeda [<rauligs@insper.edu.br>](mailto:<rauligs@insper.edu.br>)  
Fabio Ayres [<fabioja@insper.edu.br>](mailto:<fabioja@insper.edu.br>)

# Modos de Programação



# Python

O nome Python teve a sua origem no grupo humorístico britânico Monty Python



# Onde Python é usado?

usam largamente Python:



YAHOO!



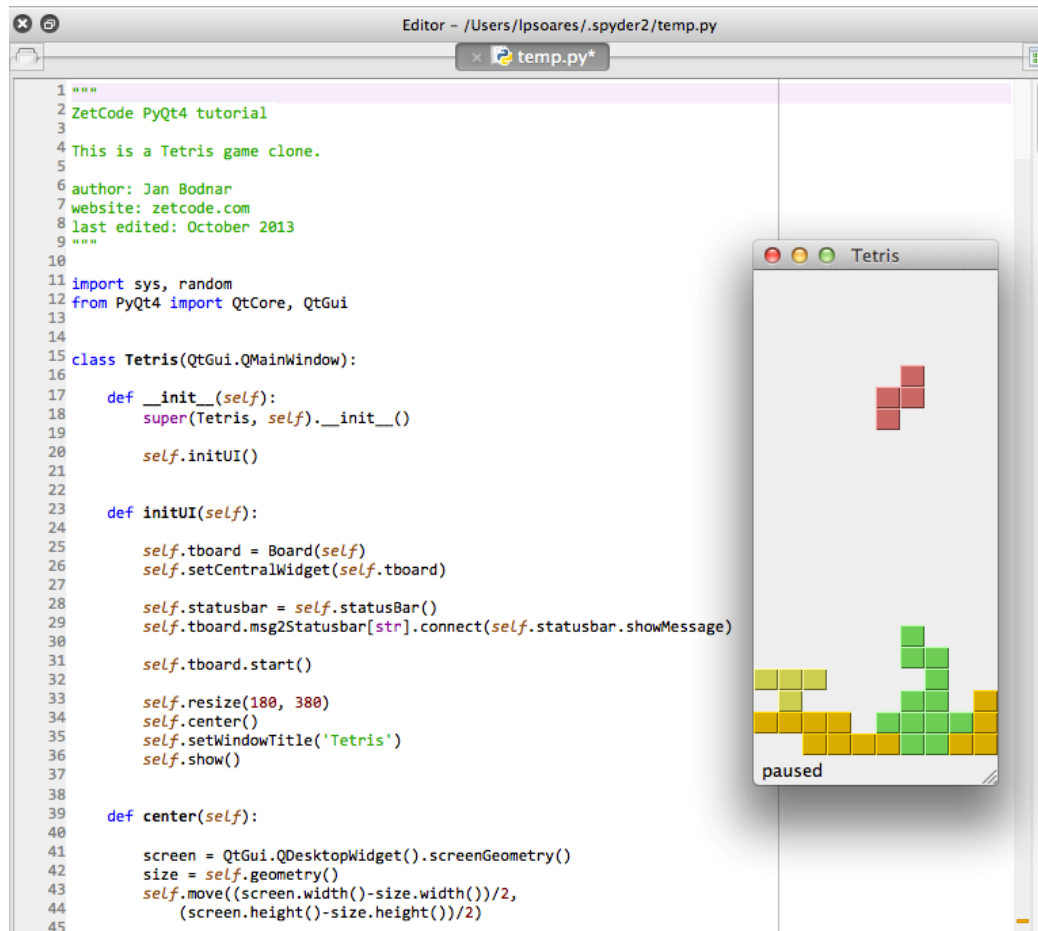
Exemplos de projetos:

- Django
- Dropbox
- Panda3D

Muito mais em [http://en.wikipedia.org/wiki/List\\_of\\_Python\\_software](http://en.wikipedia.org/wiki/List_of_Python_software)

# Como é a cara do Python?

Tetris: <http://zetcode.com/gui/pyqt4/thetetrisgame/>

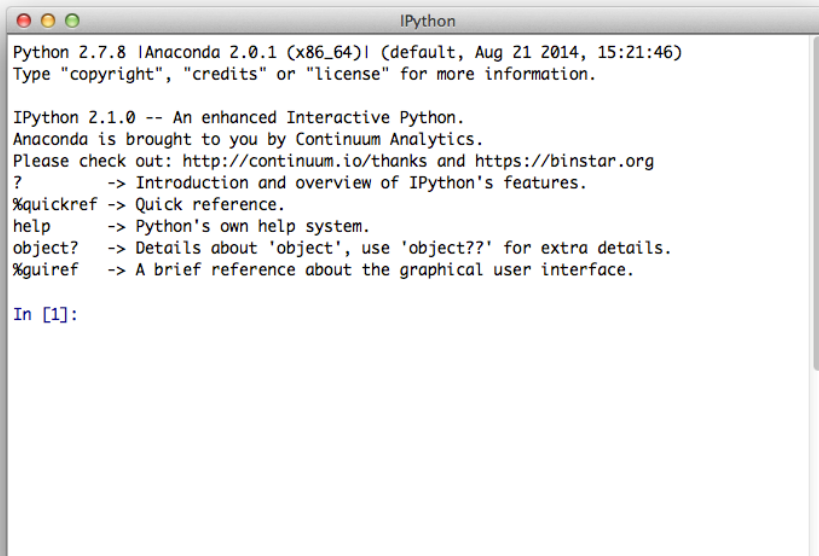


The image shows a screenshot of a Python IDE window titled "Editor - /Users/lpsoares/.spyder2/temp.py". The code is a Tetris game implementation using PyQt4. It includes a docstring, imports, and a class definition for Tetris. The game window, titled "Tetris", is visible in the foreground, showing a Tetris board with yellow and green blocks. The status bar at the bottom of the game window says "paused".

```
1 """
2 ZetCode PyQt4 tutorial
3
4 This is a Tetris game clone.
5
6 author: Jan Bodnar
7 website: zetcode.com
8 last edited: October 2013
9 """
10
11 import sys, random
12 from PyQt4 import QtCore, QtGui
13
14
15 class Tetris(QtGui.QMainWindow):
16
17     def __init__(self):
18         super(Tetris, self).__init__()
19
20         self.initUI()
21
22
23     def initUI(self):
24
25         self.tboard = Board(self)
26         self.setCentralWidget(self.tboard)
27
28         self.statusbar = self.statusBar()
29         self.tboard.msg2StatusBar[str].connect(self.statusbar.showMessage)
30
31         self.tboard.start()
32
33         self.resize(180, 380)
34         self.center()
35         self.setWindowTitle('Tetris')
36         self.show()
37
38
39     def center(self):
40
41         screen = QtGui.QDesktopWidget().screenGeometry()
42         size = self.geometry()
43         self.move((screen.width()-size.width())/2,
44                 (screen.height()-size.height())/2)
45
```

# Console Python

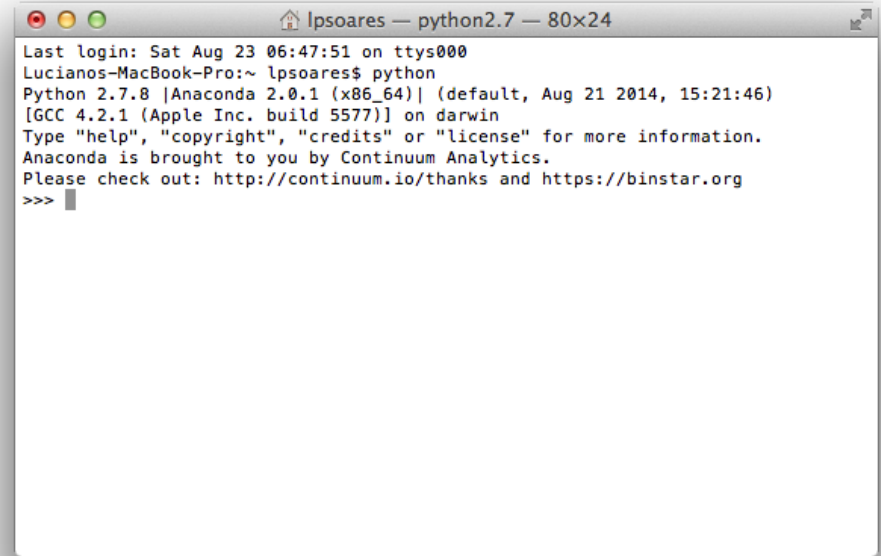
Permite executar comandos Python diretamente em uma linha de comando.

A screenshot of a macOS window titled "IPython". The window shows the IPython 2.1.0 startup banner, which includes the Python version (2.7.8), Anaconda version (2.0.1), and a list of help commands like %quickref, help, object?, and %gui. At the bottom, the prompt "In [1]:" is visible.

```
Python 2.7.8 |Anaconda 2.0.1 (x86_64)| (default, Aug 21 2014, 15:21:46)
Type "copyright", "credits" or "license" for more information.

IPython 2.1.0 -- An enhanced Interactive Python.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://binstar.org
?          -> 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.

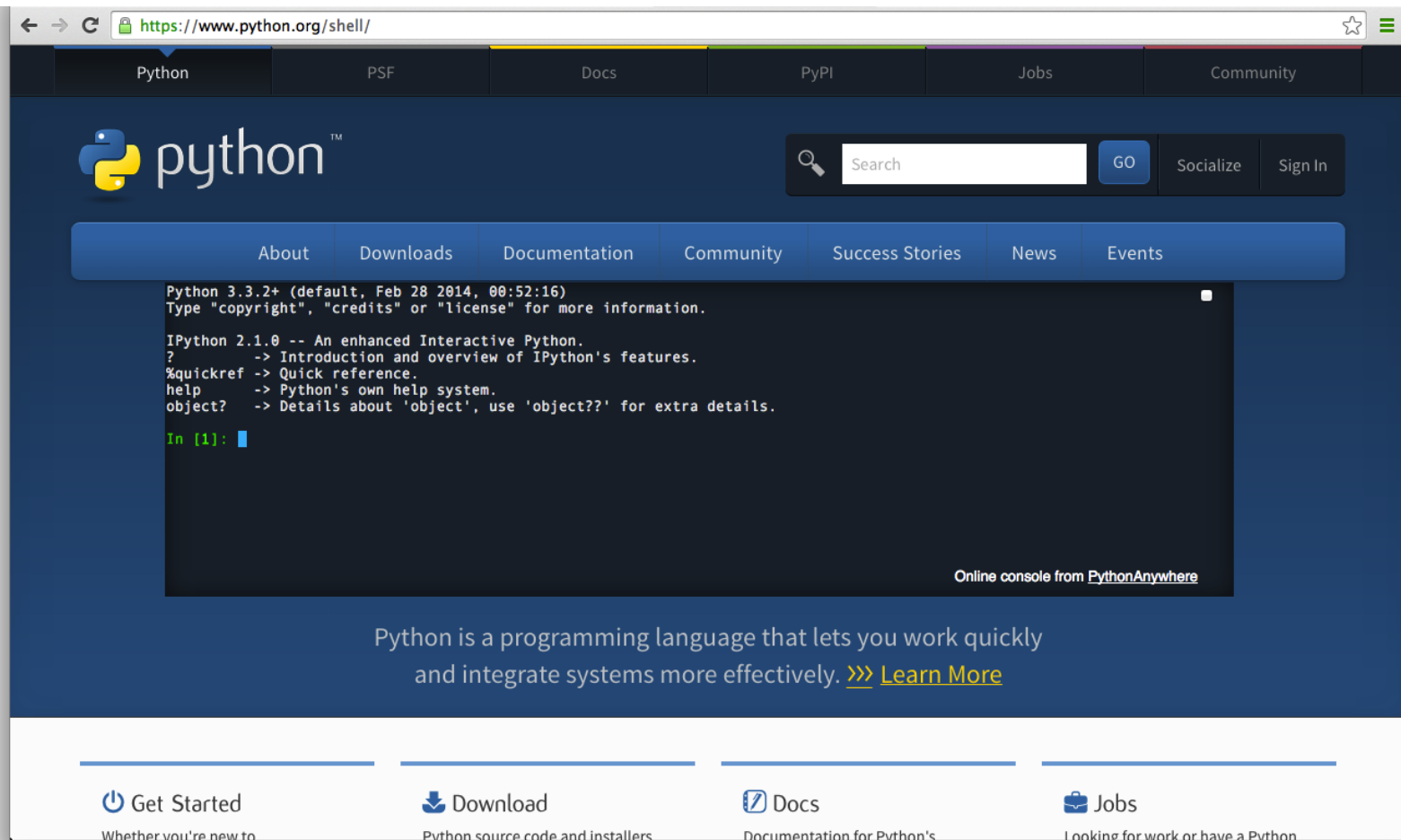
In [1]:
```

A screenshot of a macOS terminal window titled "lpsoares — python2.7 — 80x24". The terminal shows the output of the command "python", which is identical to the IPython startup banner shown in the previous screenshot. The prompt ">>>" is visible at the bottom.

```
Last login: Sat Aug 23 06:47:51 on ttys000
Lucianos-MacBook-Pro:~ lpsoares$ python
Python 2.7.8 |Anaconda 2.0.1 (x86_64)| (default, Aug 21 2014, 15:21:46)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://binstar.org
>>>
```

# Python 3 Online

<https://www.python.org/shell/>



The screenshot shows the Python 3 Online shell interface. At the top, there's a navigation bar with links: Python, PSF, Docs, PyPI, Jobs, and Community. Below this is a search bar with a magnifying glass icon and a 'GO' button, along with 'Socialize' and 'Sign In' links. A secondary navigation bar contains links: About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area features a terminal window with the following text:

```
Python 3.3.2+ (default, Feb 28 2014, 00:52:16)
Type "copyright", "credits" or "license" for more information.

IPython 2.1.0 -- 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.

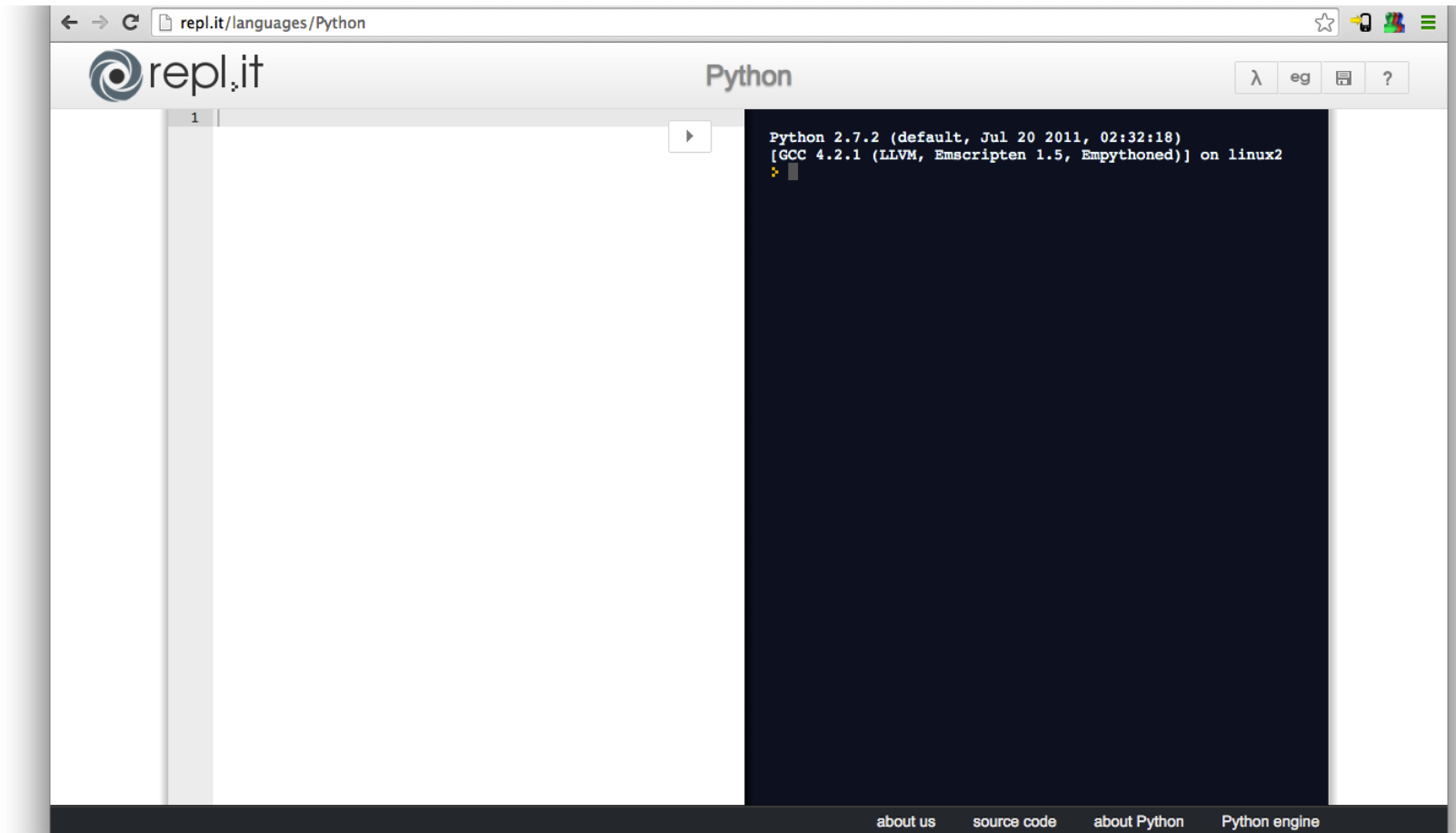
In [1]:
```

Below the terminal window, it says "Online console from [PythonAnywhere](#)". At the bottom of the main content area, there's a message: "Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)". The footer contains four columns: "Get Started" (with a power icon and the text "Whether you're new to"), "Download" (with a download icon and the text "Python source code and installers"), "Docs" (with a book icon and the text "Documentation for Python's"), and "Jobs" (with a briefcase icon and the text "Looking for work or have a Python").



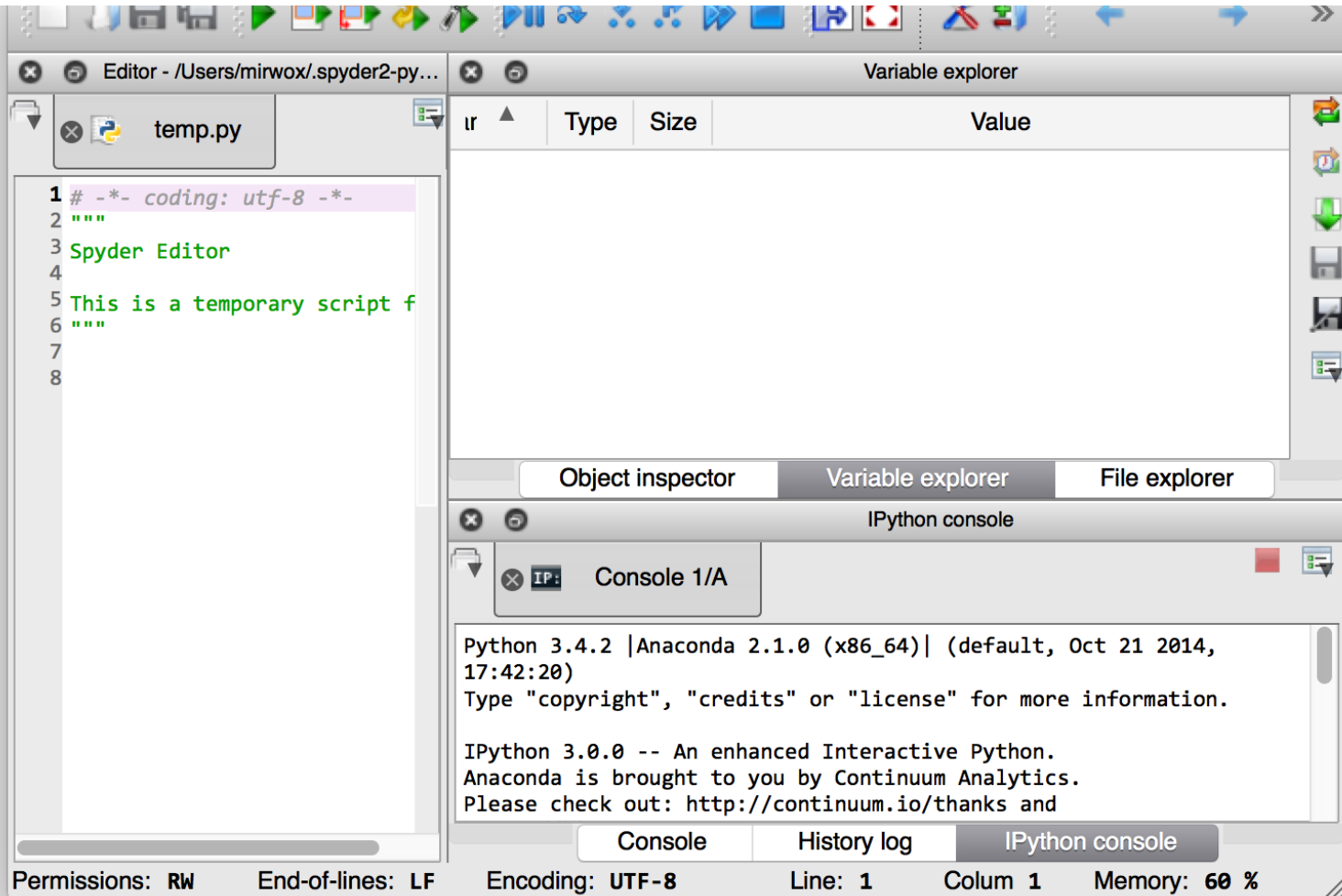
# Python 3 Online

<http://repl.it/languages/Python3>



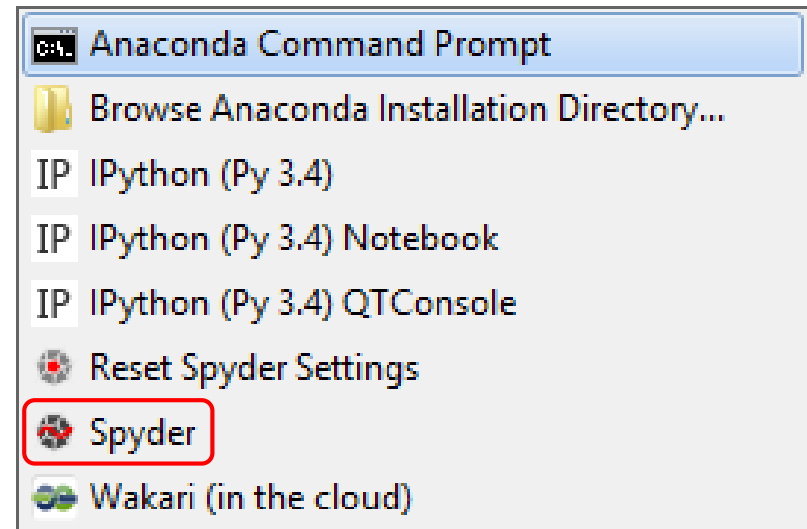
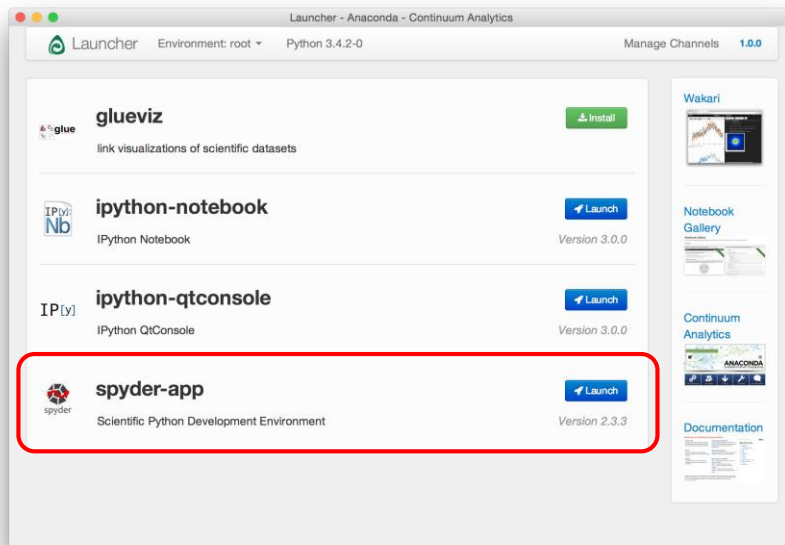


# Anaconda



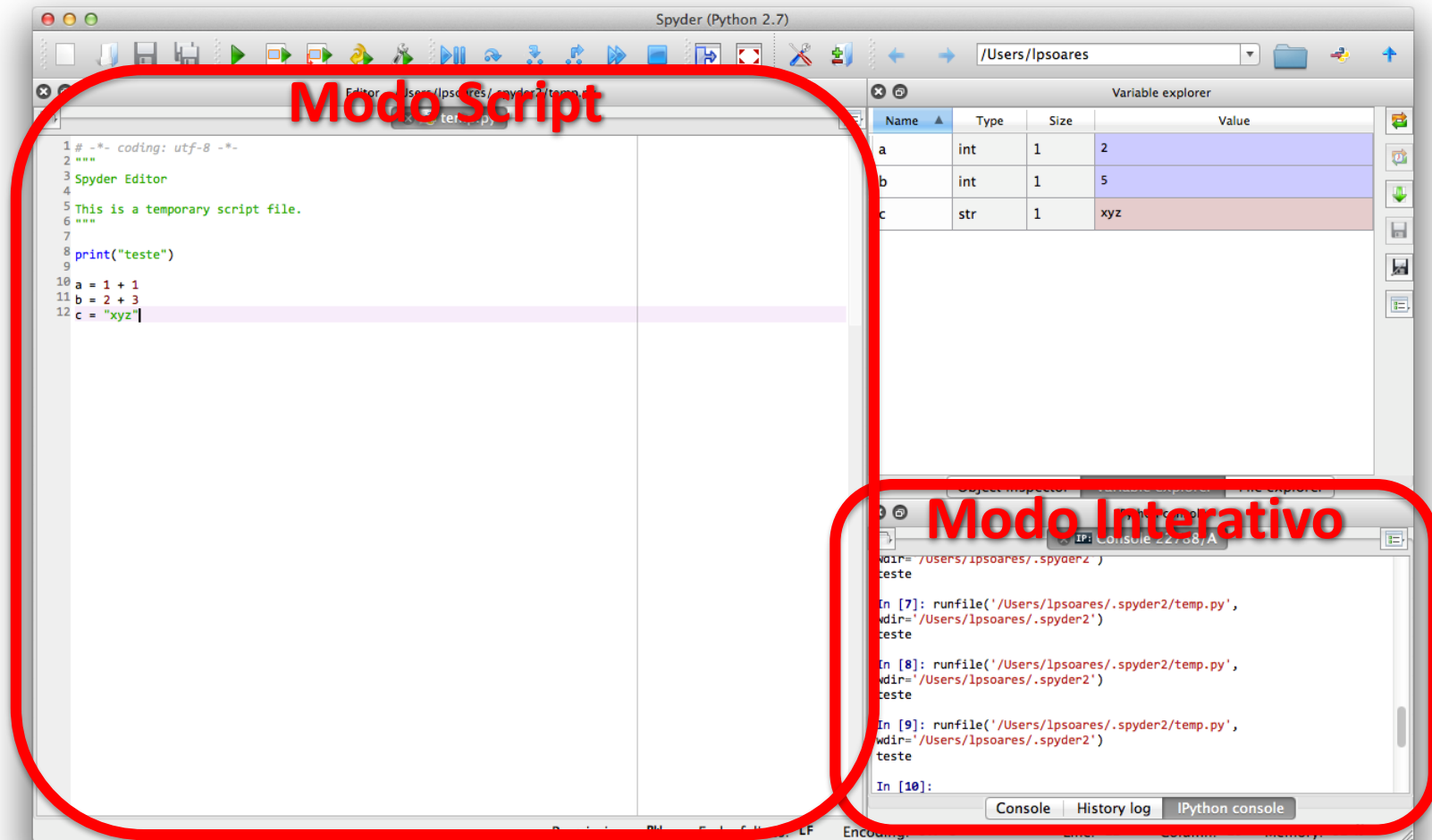
# Usando o Anaconda

Vantagem: inclui mais de 195 dos mais populares pacotes de Python para ciências, matemática, engenharia, análise de dados, etc. Além de ser fácil instalar mais pacotes.



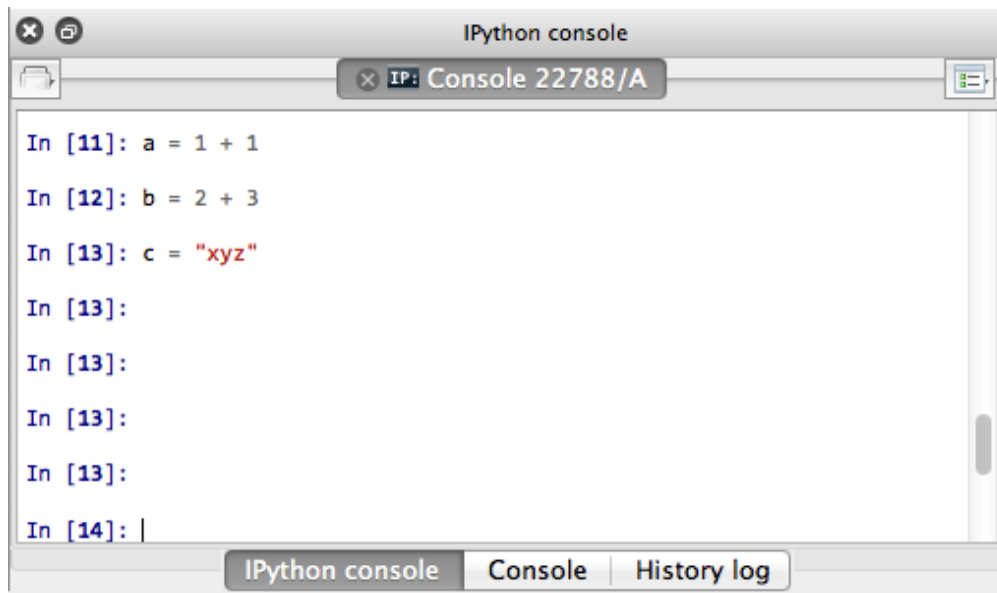
# Entendendo a IDE Spyder

(*Interactive Development Environment*)



# Vantagens de se trabalhar em uma IDE

Por exemplo no Variable Explorer é possível ver o estado das variáveis do seu programa:

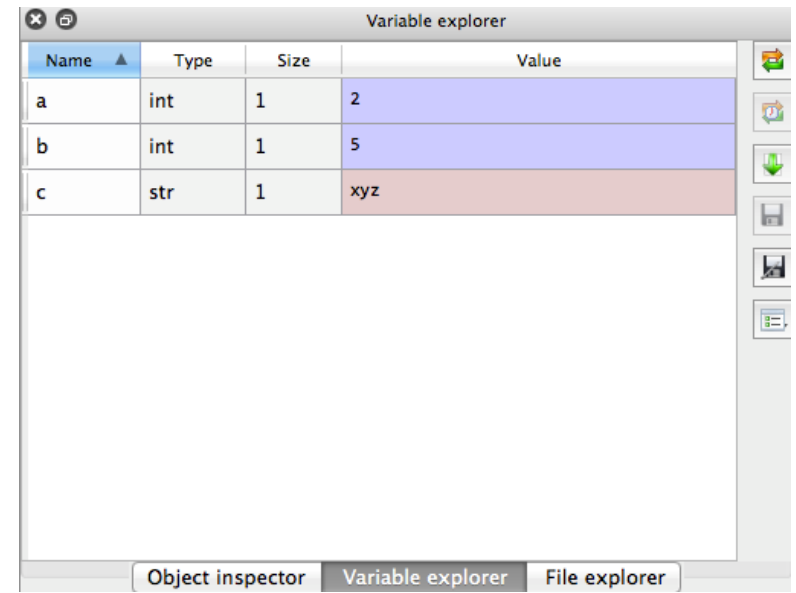


The screenshot shows the IPython console window with the following content:

```
IPython console
IP: Console 22788/A

In [11]: a = 1 + 1
In [12]: b = 2 + 3
In [13]: c = "xyz"
In [13]:
In [13]:
In [13]:
In [13]:
In [14]: |
```

The window has tabs for 'IPython console', 'Console', and 'History log'.

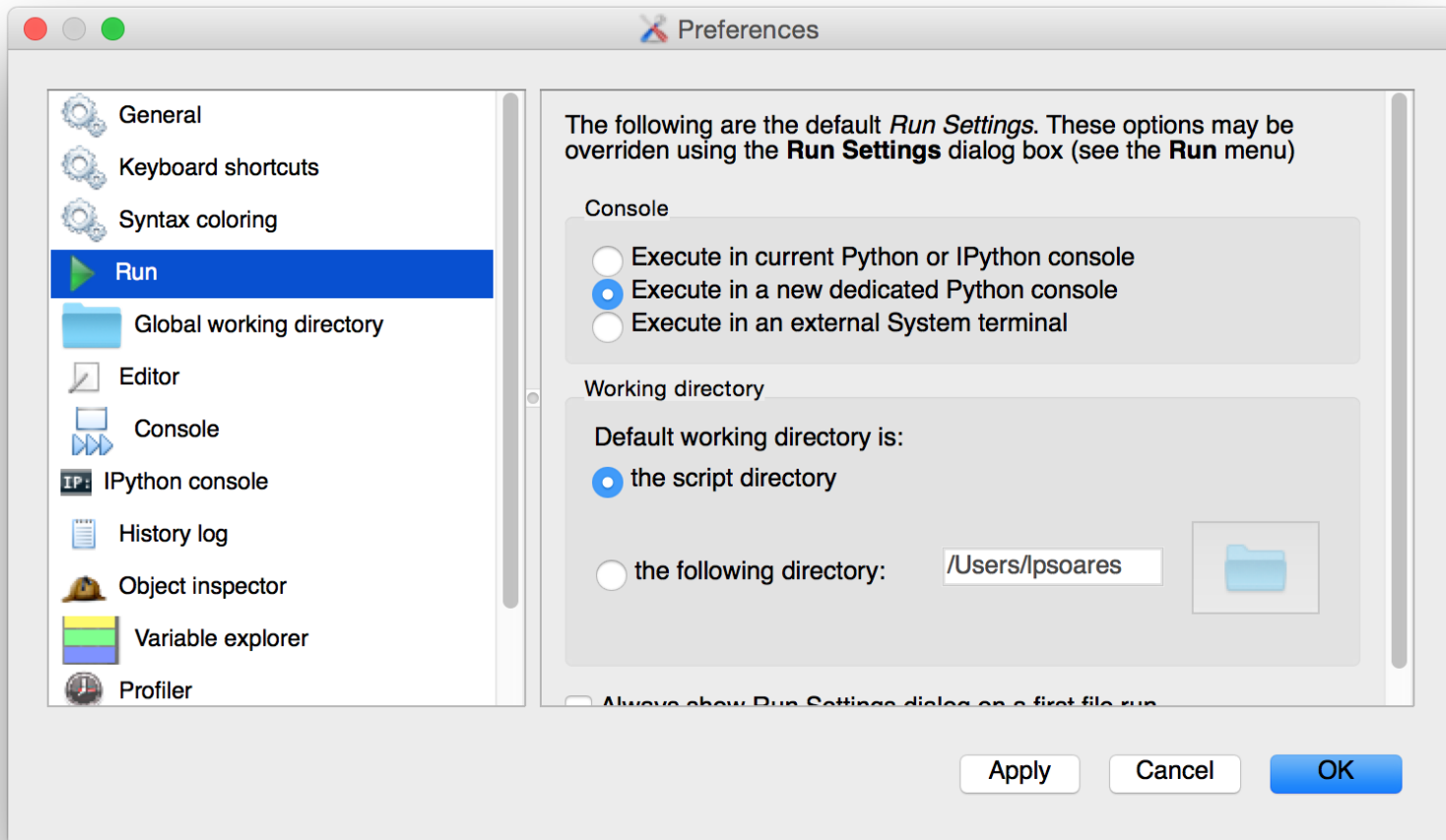


The screenshot shows the Variable explorer window with a table of variables:

Name	Type	Size	Value
a	int	1	2
b	int	1	5
c	str	1	xyz

The window has tabs for 'Object inspector', 'Variable explorer', and 'File explorer'.

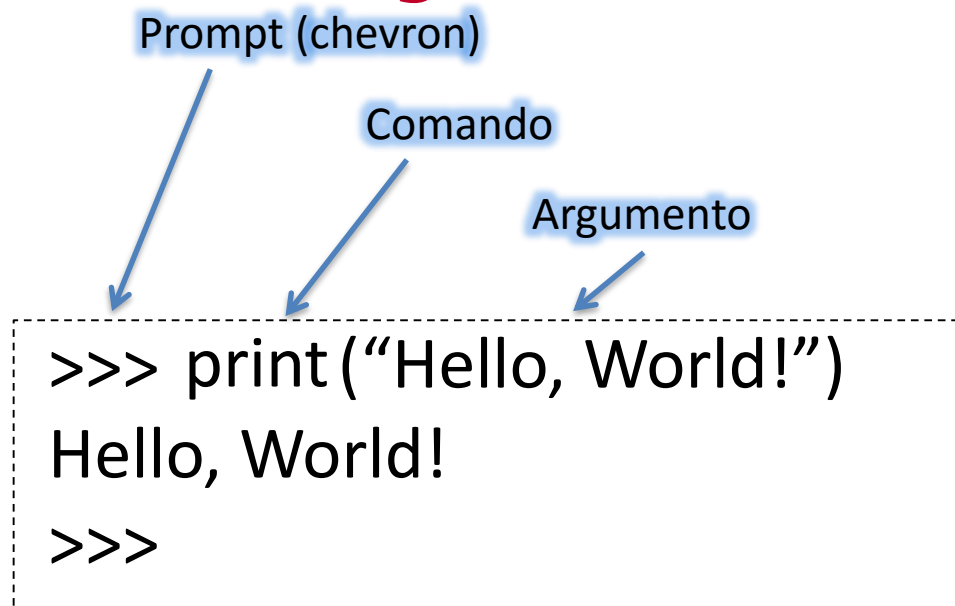
# Dedicated Console



# Primeiro Programa

Prompt (chevron)      Comando      Argumento

```
>>> print("Hello, World!")  
Hello, World!  
>>>
```



Teste o seu “Hello, World!” no prompt (Anaconda ou online)

# Buscando Ajuda

Visite <https://www.python.org/>

Digite `help()` na linha de comando





# Atividade

Use a linha de comando como uma calculadora:

Exemplo:

```
>>> 2 + 3.141
5.141
>>> 7 **2
49
>>> 5 ** (34/(7 - 1/3))
3670.684197150057
```

# Operadores e Operandos

Operador	Ação
+	Adição
-	Subtração
*	Multiplicação
**	Potência
/	Divisão
//	Divisão de inteiros
%	Resto da divisão

# Prompt como calculadora

Continue no Anaconda ou online

Teste todas as opções: parênteses, exponenciação, multiplicação, divisão, divisão inteira, resto, adição e subtração

# Precedência

Qual o valor de y?

$$y = 3 / 2 * 4 / 2 * * 6 / 3 + 49 * * (1 / 2)$$

# Ordem dos Operadores

Python segue a convenção matemática do PEMDAS:

**P**arênteses têm a maior prioridade e pode ser usado para forçar uma expressão a ser avaliada na ordem que você quiser.

A **E**xponenciação tem a próxima precedência mais alta.

**M**ultiplicação e **D**ivisão têm a mesma precedência, que é maior do que **A**dição e **S**ubtração.

Além disso as operações de mesma prioridade são avaliados da esquerda para a direita.

# Variáveis

Armazenam valores para o programador.  
= é atribuição

```
x = 10.0
```

```
y = "Ola"
```

```
z = 1 + 2
```

```
q = True
```

# Prática de variáveis

Use o prompt para responder ao problema usando variáveis:

Volume de uma esfera com raio  $r$  é  $\frac{4}{3}\pi r^3$

Qual é o volume de uma esfera com raio igual a 5?

Dica: Descubra como usar o  $\pi$  em Python



# import

```
>>> import math
```

```
>>> dir(math)
```

```
['__doc__',  
'__loader__',  
'__name__',  
'__package__',  
'acos',  
'acosh',  
'asin',  
'asinh',  
'atan',  
'atan2',  
'atanh',  
'ceil',  
'copysign',  
'cos',
```

```
'cosh',  
'degrees',  
'e',  
'erf',  
'erfc',  
'exp',  
'expm1',  
'fabs',  
'factorial',  
'floor',  
'fmod',  
'frexp',  
'fsum',  
'gamma',  
'hypot',  
'isfinite',  
'isinf',
```

```
'isnan',  
'ldexp',  
'lgamma',  
'log',  
'log10',  
'log1p',  
'log2',  
'modf',  
'pi',  
'pow',  
'radians',  
'sin',  
'sinh',  
'sqrt',  
'tan',  
'tanh',  
'trunc']
```

# Modulos

Módulos (module) são como caixas de ferramentas à nossa disposição

- Também chamadas de 'bibliotecas' (*library*)
  - Você tem que 'importar' o módulo quando quiser usá-lo

```
>>> import math
>>> print(math.pi)
```
  - Ou então importa só o que precisa

```
>>> from math import pi
>>> print(pi)
```

# Pacotes

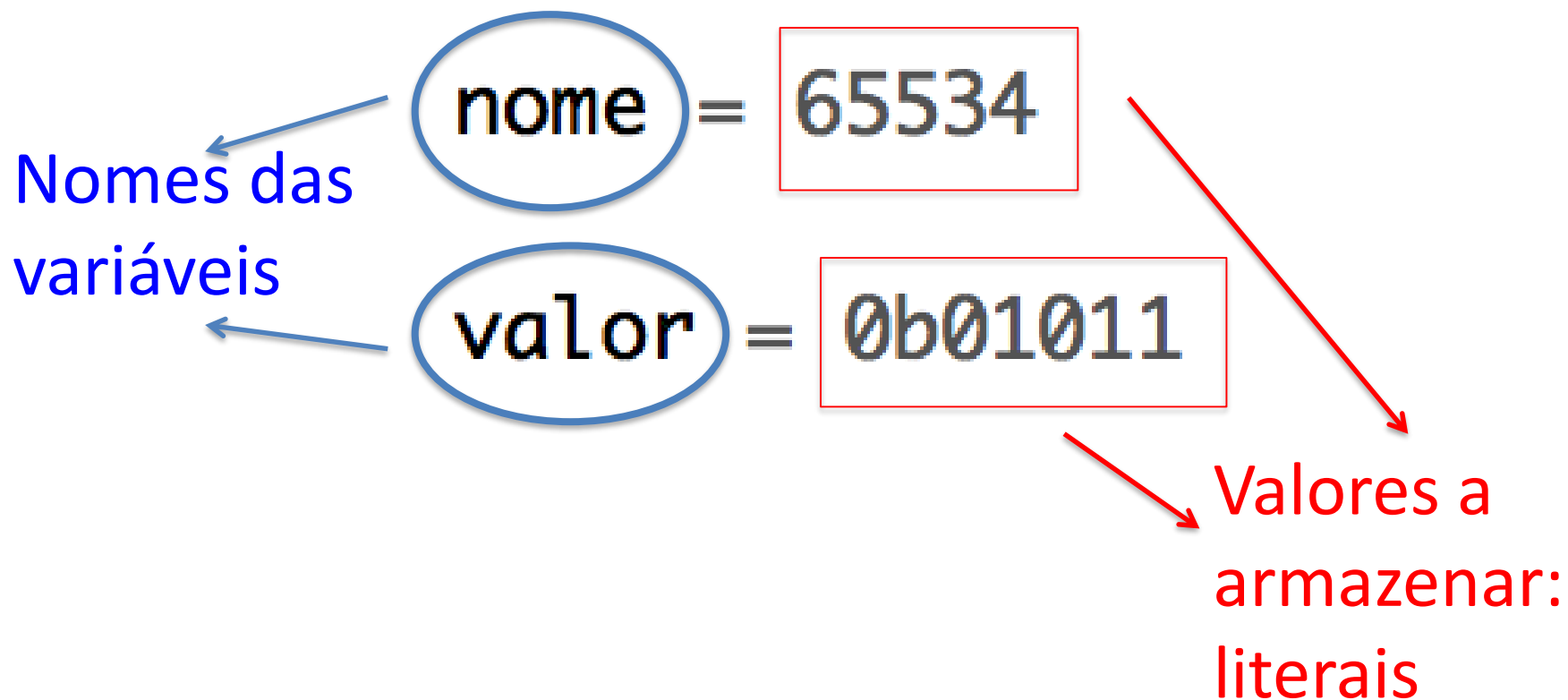
Pacote (*package*): Um “combo” de módulos

- Python tem pacotes para tudo!
  - Programação de jogos
  - Manipulação de documentos web
  - Criptografia
  - Banco de dados
  - Estatística
  - Etc

conda install

pip install

# Literais



# Literais Numéricos

## Decimal

```
valor = 99877
```

## Octal

```
valor = 0o125252
```

## Hexadecimal

```
valor = 0xaaaa
```

## Binário

```
valor = 0b1010101010101010
```

## Real (ponto flutuante ou float)

```
valor = 1.0
```

# Pedindo dados para o usuário

- A função `input()` pede um texto para o usuário, e retorna este texto ao programa
  - Você pode guardar esse texto em uma variável para usar em seguida

```
>>> cor_favorita = input("Qual a sua cor favorita? ")
Qual a sua cor favorita? azul
>>> print(cor_favorita)
azul
>>> type(cor_favorita)
<class 'str'>
>>>
```

- Se você quiser um número ao invés de um texto, tem que converter de texto ('string') para número

# Praticando leitura de dados

Atividade: Digite e rode o seguinte programa

```
# Saude o cliente.  
nome = input('Qual é o seu nome? ')  
print('Seja bem vindo', nome)  
  
# Calcule a gorjeta.  
gasto_string = input('Quanto você gastou? ')  
gasto = float(gasto_string)  
gorjeta = 0.1 * gasto  
print('A gorjeta sugerida é', gorjeta)
```



# Tipos de Valores e Variáveis

Como saber o tipo? Como converter tipos?

```
>>> type('Hello, World!')
<type 'str'>
>>> type(17)
<type 'int'>
>>> n = 17
>>> pi = 3.1415926535897932
>>> type(n)
<type 'int'>
>>> type(pi)
<type 'float'>
```

# Tipos de Valores e Variáveis (2)

Funções para conversão de tipos

`int()` converte para inteiro.

```
In [12]: int(3.14)
Out[12]: 3
```

`float()` converte para ponto flutuante

```
In [13]: float(4)
Out[13]: 4.0
```

`str()` converte para string

```
In [14]: str(4)
Out[14]: '4'
```

# Nomes de Variáveis

- Devem começar com uma letra
- Podem ter nomes bem longos
- Podem usar o `_` (*underscore*), mas não sozinho
- Não podem ter caracteres especiais
- Não podem ser palavras reservadas

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

# Exercícios

1. Escreva um programa que pergunte a quantidade de km percorridos por um carro alugado, assim como a quantidade de dias pelos quais o carro foi alugado. Calcule o preço a pagar, sabendo que o carro custa R\$60,00 por dia e R\$0,15 por km rodado
2. Escreva um programa que pergunte a temperatura em Celsius e a converta para Fahrenheit.

## Exercícios

3. Suponha que o preço de um livro é de R\$24,95, mas as livrarias recebem um desconto de 40%. O frete custa R\$3,00 para a primeira cópia e 75 centavos para cada cópia adicional. Qual é o custo de atacado total para 60 cópias?
4. Se eu deixar minha casa às 06h52 e correr um quilômetro em um ritmo leve (8:15 por quilômetro), em seguida, 3 quilômetros num ritmo puxado (7:12 por quilômetro) e 1 quilômetro em ritmo leve novamente. A que horas eu chego em casa para o café da manhã?

# Exercícios

5. Escreva um programa que pergunte, em sequência, uma quantidade de dias, horas, minutos e segundos para o usuário. Depois calcule o total em segundos e imprima.
6. Escreva um programa que calcule a redução de tempo de vida de um fumante a partir do número de cigarros. Pergunte quantos cigarros ele fuma por dia e há quantos anos fuma. Imprima o tempo de vida perdido em dias. Considere que um cigarro rouba 10 minutos de expectativa de vida.

# Exercícios

7. Escreva um programa que peça os dois catetos de um triângulo retângulo e calcule o valor da hipotenusa.

Dica: `math.sqrt()` calcula o valor da raiz quadrada



# Recursos online

Enthought Academic

<https://www.enthought.com/>

Codecademy

<http://www.codecademy.com/en/tracks/python>

Learn Python

<https://learnpython.trinket.io/>

# Para a próxima aula

- Condições (if, else)

Capítulos 4 do livro

**Introdução à Programação com Python,**  
de Nilo Ney Menezes.

(disponível na biblioteca)



# Insper

[www.insper.edu.br](http://www.insper.edu.br)

# Fun: import antigravity

