Python

Thiago Franco de Moraes

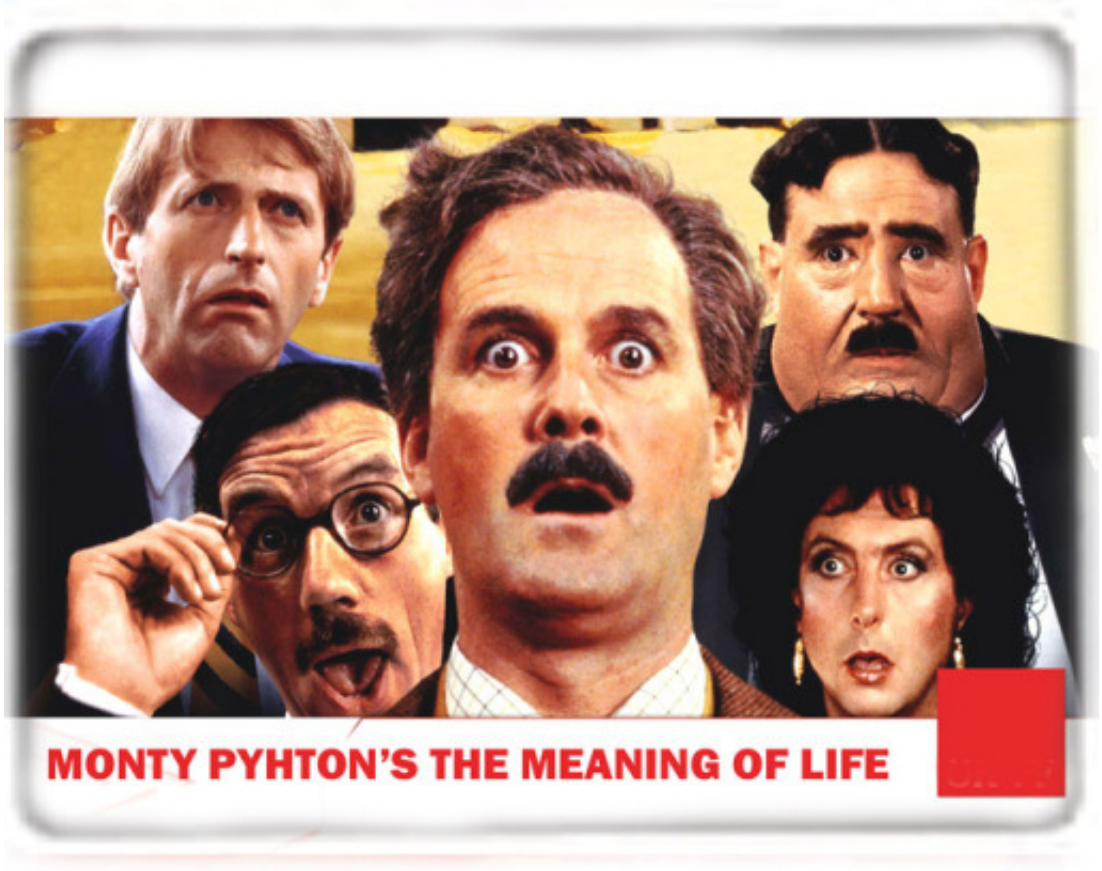
Paulo Henrique Junqueira Amorim

O que é Python

- Linguagem de programação de alto nível e interpretada;
- Multi-paradigma (Orientada à objetos, Funcional, Procedural);
- Multiplataforma (Linux, Mac OS X, Windows, *BSD, Haiku, ...);
- Open Source;
- Tipagem dinâmica;
- White spaces;
- Baterias incluidas.

História

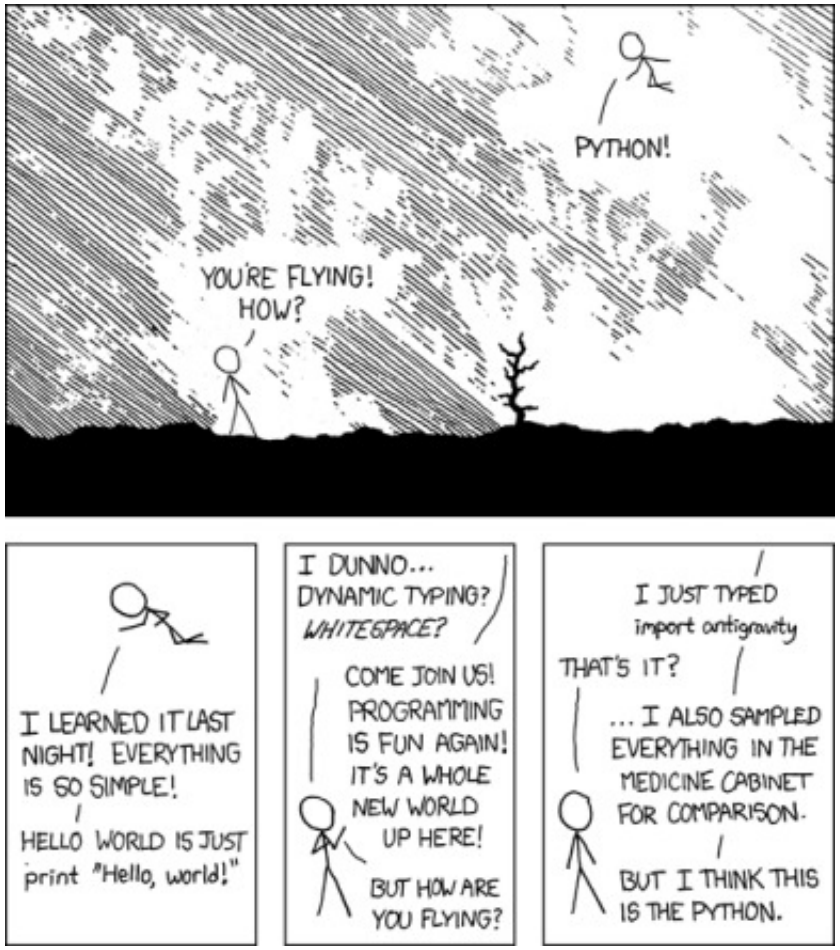
- Criada por Guido van Rossum em 1989;
- Baseada na linguagem ABC;
- Nome vem do grupo de Humor Monty Python.



Um pouco de filosofia

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

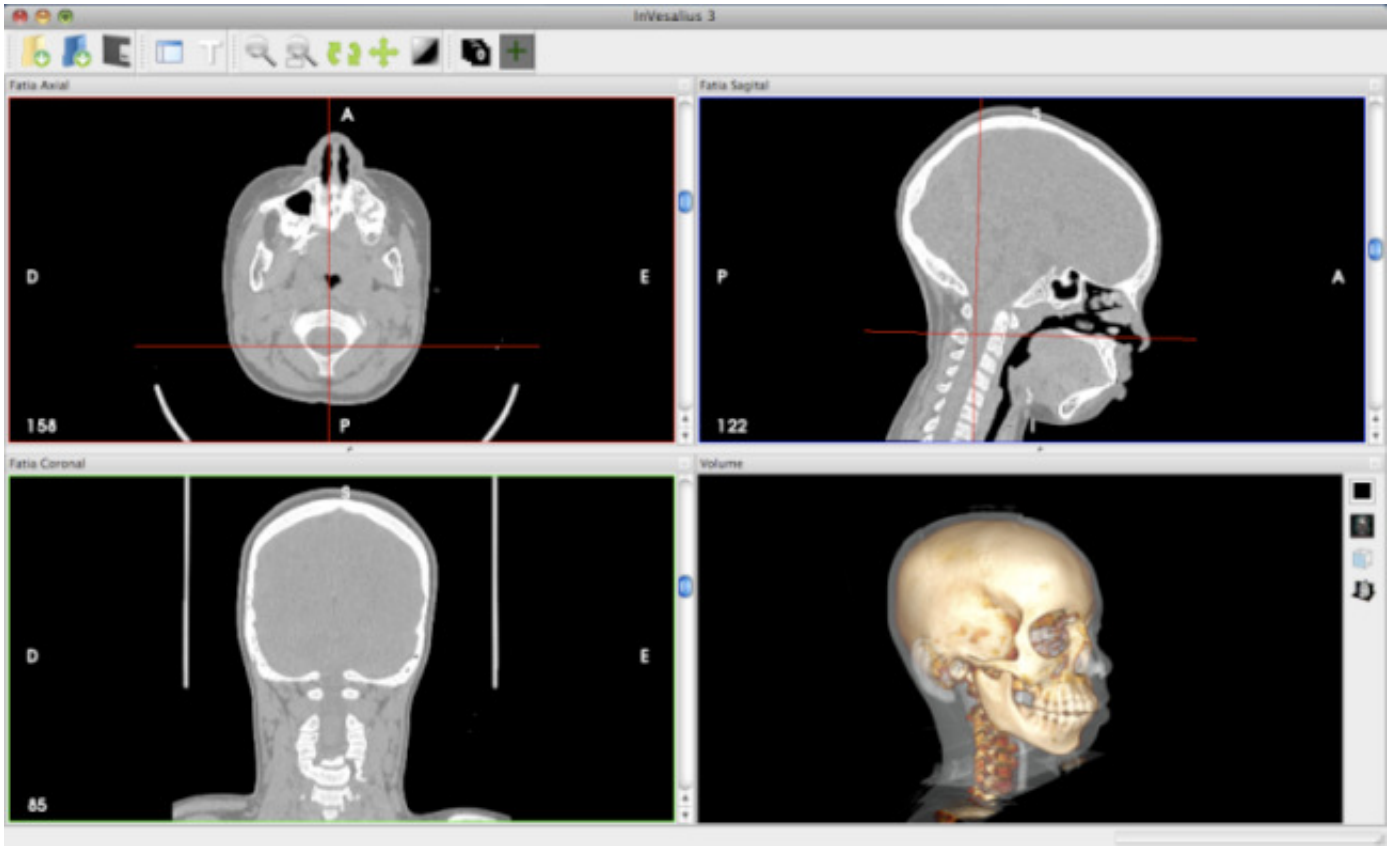
Programar é divertido!



Quem usa?

- **Google**: coode.google.com, google groups, ferramentas internas;
- **Globo**: globo.com;
- **Facebook**: FriendFeed (Tornado);
- **Gimp**: Extensões;
- **Blender**: Extensões;
- **Mozilla**: Firefox Sync Server;
- **NASA**: Internamente;
- **Industrial Light & Magic (ILM)**: Pipeline de renderização;
- ...
- Mais em http://www.python.org/about/success/.

Invesalius
Python + wxPython + VTK + GDCM + Numpy + Nibabel

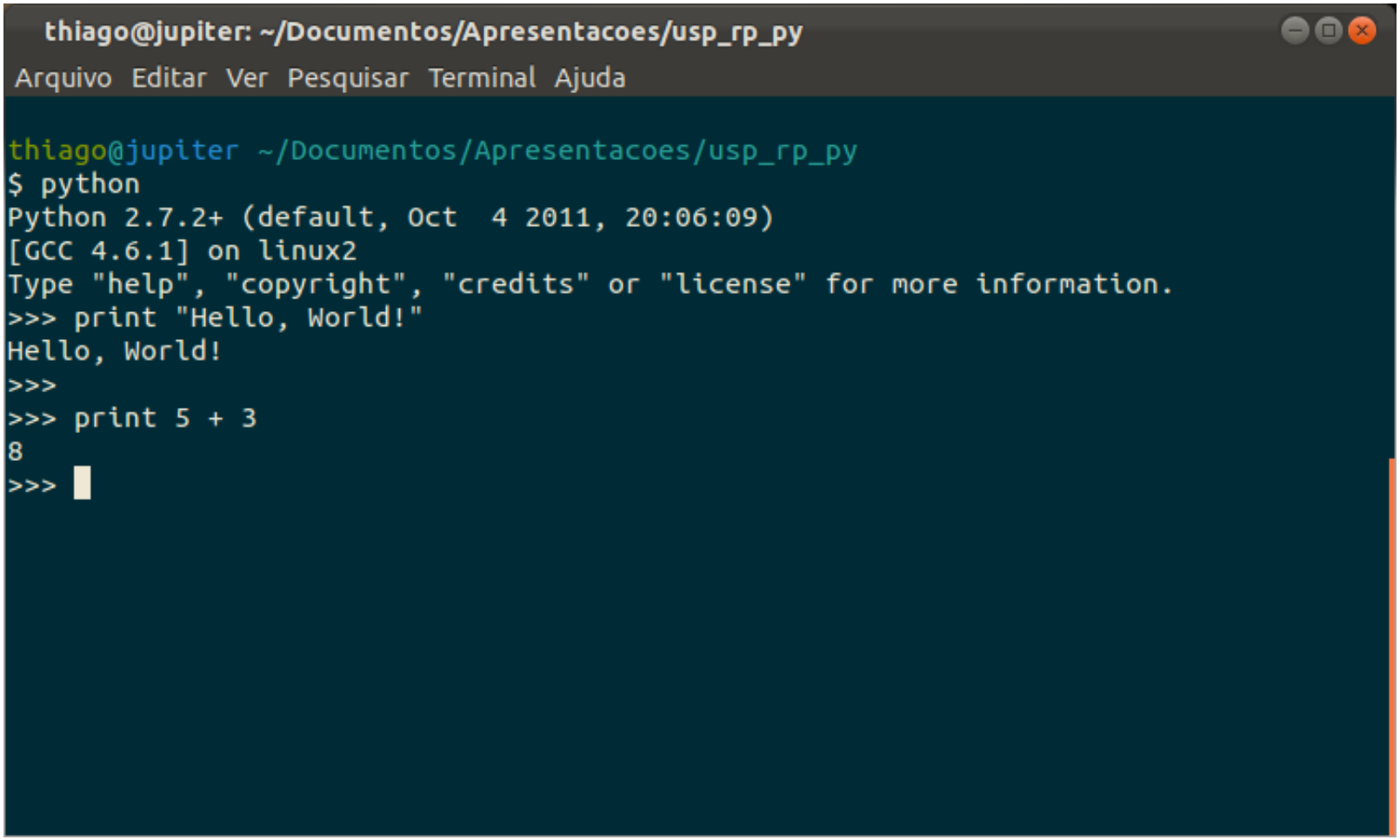


http://svn.softwarepublico.gov.br/trac/invesalius

Baterias incluídas

- XML(**ElementTree**), HTML
- Interface gráfica (**TKinter**)
- Protocolos de internet: HTTP, FTP (**ftplib**), SMTP (**smtplib**), POP3 (**poplib**), ...
- Internacionalização: gettext, locale
- Persistência de dados: SQL (**sqlite3**), dbm, pickle
- ZIP (**zipfile**), BZ2, Gzip, Tar (**tarfile**)
- Thread (**threading**), Multiprocesso (**multiprocessing**)
- Comunicação via rede e interprocessos: sockets, ssl, signal
- ...
- Mais em Pypi - http://pypi.python.org/pypi

Interpretador Interativo



Tipos de dados

```
 1 #Números
 2 123, 3.1415, 1+3j
 3
 4 #Strings
 5 'Python', u'파이썬'
 6
 7 #Listas
 8 [1, 3.145, 'lista', u'listão']
 9
10 #Dicionários
11 {'nome': 'James Bond', 'cod': '007'}
12
13 #Tuplas
14 (1, 2, 3, 'quatro')
```

Funções

```
 1 def hello():
 2     print "Hello World!"
 3 
 4 >>> hello()
 5 Hello World!
 6 
 7 def primo(n):
 8     for i in xrange(2, n):
 9         if n % i == 0:
10             return False
11     return True
12 
13 >>> primo(27)
14 False
15 
16 add = lambda a,b: a + b
17 
18 >>> add("Foo", "Bar")
19 FooBar
```

Classes

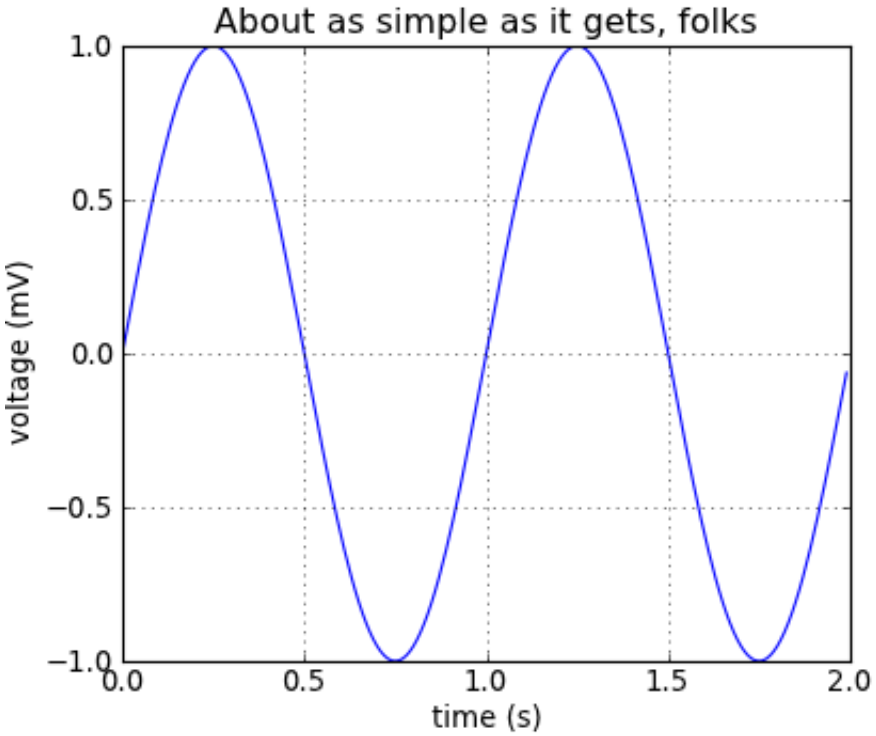
```
 1 import random
 2 class Manolo():
 3     def __init__(self, nome):
 4     self.nome = nome
 5
 6 def faz_trabalho(self):
 7     if random.randint(1, 2) == 1:
 8             return "%s, está cansado!" % self.nome
 9     return "Pô, mandaí, que %s faz!" % self.nome
10
11 class MatadorDeAluguel(Manolo):
12     def faz_trabalho():
13     return "Quem %s deve matar?" % self.nome
14
15 >>> manolo = Manolo("Tiburcio")
16 >>> print manolo.faz_trabalho()
17 "Tiburcio, está cansado"
18
19 >>> print manolo.faz_trabalho()
20 "Pô, mandaí, que Tiburcio faz!"
```

Python cientifico

- **Numpy -** Vetores e matrizes N-dimensionais.
- **Scipy -** Algoritmos numéricos e simbólicos.
- **Matplotlib -** Gerar gráficos.
- **VTK -** Visualização.
- **RPy -** Interface entre a linguagem R e Python.
- **Biopython -** Ciências biológicas.
- **Sympy -** Matemática simbólica.
- ...

Plotando um gráfico

```
 1 from pylab import *
 2
 3 T = arange(0.0, 2.0, 0.01)
 4 S = sin(2*pi*t)
 5 Plot(t, s, linewidth=1.0)
 6
 7 Xlabel('time (s)')
 8 Ylabel('voltage (mV)')
 9 Title('About as simple as it gets, folks')
10 Grid(True)
11 Show()
```



Quais ferramentas utilizar?

- Interpretador interativo + editor de textos (Vim, Emacs, notepad, ...) - Simples assim!

Não tem IDE, não?

- SPE
- Pida
- Eric4
- Eclipse
- Wing
- Komodo
- PyCharm


django

O que é Django?

- Framework web desenvolvido em Python;
- Padrão MVC (Model, View, Controler);
- Reusabilidade e plugabilidade;
- DRY (Don't Repeat Yourself);
- Open Source;

ORM

- Escreva uma vez só, execute em vários banco de dados;
- Suporta PostgreSQL, MySQL, sqlite3, Oracle.

Código Python

```
 1 from django.db import models
 2
 3 class Poll(models.Model):
 4     question = models.CharField(max_length=200)
 5     pub_date = models.DateTimeField('date published')
 6
 7 class Choice(models.Model):
 8     poll = models.ForeignKey(Poll)
 9     choice = models.CharField(max_length=200)
10     votes = models.IntegerField()
```

ORM
Código SQL

```
 1 BEGIN;
 2 CREATE TABLE "polls_poll" (
 3     "id" serial NOT NULL PRIMARY KEY,
 4     "question" varchar(200) NOT NULL,
 5     "pub_date" timestamp with time zone NOT NULL
 6 );
 7 CREATE TABLE "polls_choice" (
 8     "id" serial NOT NULL PRIMARY KEY,
 9     "poll_id" integer NOT NULL REFERENCES "polls_poll" ("id"),
10     "choice" varchar(200) NOT NULL,
11     "votes" integer NOT NULL
12 );
13 COMMIT;
```

Brincando com os dados

```
 1 >>> from polls.models import Poll, Choice # Import the model classes we just wrote.
 2
 3 >>> Poll.objects.all()
 4 []
 5
 6 >>> import datetime
 7 >>> p = Poll(question="What's up?", pub_date=datetime.datetime.now())
 8
 9 >>> p.save()
10
11 >>> Poll.objects.filter(id=1)
12 [<Poll: What's up?>]
```

Views

- Onde fica a lógica da aplicação;
- Trabalha com a idéia de requisição e resposta;

```
 1 from django.template import Context, loader
 2 from polls.models import Poll
 3 from django.http import HttpResponse
 4
 5 def index(request):
 6     latest_poll_list = Poll.objects.all().order_by('-pub_date')[:5]
 7     t = loader.get_template('polls/index.html')
 8     c = Context({
 9         'latest_poll_list': latest_poll_list,
10     })
11     return HttpResponse(t.render(c))
```

Templates

```
1 {% if latest_poll_list %}
2     <ul>
3     {% for poll in latest_poll_list %}
4         <li><a href="/polls/{{ poll.id }}/">{{ poll.question }}</a></li>
5     {% endfor %}
6     </ul>
7 {% else %}
8     <p>No polls are available.</p>
9 {% endif %}
```

Roteamento de URLs

- Mapeia um url com sua correspondente função da view;
- Utiliza-se de Regex (expressões regulares).

```
1 from django.conf.urls import patterns, include, url
2
3 from django.contrib import admin
4 admin.autodiscover()
5
6 urlpatterns = patterns('',
7     url(r'^polls/$', 'polls.views.index'),
8     )
```

Contato
CTI - http://www.cti.gov.br/

Thiago Franco de Moraes (tfmoraes@cti.gov.br)

Paulo Henrique Junqueira Amorim (phamorim@cti.gov.br)