

# Python vavit ! vupit !

## Introdução à Linguagem, História e Propaganda

Fernando Henrique

[liquuid.net](http://liquuid.net)

29 de setembro de 2009

# Onde e Quando

- Criada em 1989 pelo holandês **Guido van Rossum** no *Centrum voor Wiskunde en Informatica* (CWI), em Amsterdã, Holanda.
- Influenciada pela linguagem ABC, desenvolvida no CWI por Guido e outros nas décadas de 70 e 80. ABC tinha um foco bem definido: ser uma linguagem de programação para usuários inteligentes de computadores que não eram programadores: Físicos, Cientistas Sociais e até Linguistas.
- O projeto de sistema operacional distribuído Amoeba precisava de uma linguagem de *script*. Nasce o Python.

# Bases do Python

- Elementos que eram bem sucedidos no ABC.
- Estruturas de dados poderosas inclusas: Listas, Dicionários, *Strings*.
- Usar indentação para delimitar blocos, eliminando chaves.
- Fácil extensão (lição aprendida com os erros do ABC).
- Fácil de portar: além do Amoeba, gostaria de executar em Unix, Macintosh e Windows.
- Influências de Modula-2 e Modula-3: módulos e *namespaces*.

# Ambiente de Concepção

- **Universidade:** pessoas altamente especializadas para desenvolver e opinar sobre os elementos do projeto
- **Descontraído:** o nome Python vem da série de humor *Monty Python's Flying Circus*
- **Sem prazos, Sem pressão:** o desenvolvimento não foi pressionado por estratégias de marketing, prazos, clientes ou qualquer outro fator que pudesse influenciar nas decisões de projeto, **resultando em maior qualidade.**
- **Software Livre:** garante a vida da tecnologia.

# Características Básicas

- **Interpretada:** usa máquina virtual, facilita portabilidade.
- **Interativa:** pode-se programar interativamente, os comandos são executados enquanto são digitados. Por exemplo usando o ipython que facilita testes, desenvolvimento rápido e outros. Facilitadores estão presentes `help(obj)`.
- **Orientada a Objetos:** tudo<sup>1</sup> é objeto: números, *strings*, funções, classes, instâncias, métodos, ...
- **Fortemente Tipada:** Não existe *casts* e nem conversão automática, não se mistura tipos “automagicamente”.
- **Tipagem Dinâmica:** A tipagem de um objeto é feita em tempo de execução. Um objeto tem tipo, uma variável não.

---

<sup>1</sup>Quase tudo é um objeto :-)

# Características Importantes

- Sintaxe clara, sem caracteres “inúteis”:
  - blocos são marcados por indentação
  - parênteses são opcionais, só precisam ser utilizados para eliminar ambiguidades.
  - palavras-chave (*keywords*) e formações que ajudam na leitura, como `for ... in ...`.
- Fácil extensão: codificar nos módulos é muito fácil, podendo utilizar bibliotecas nativas, aproveitando desempenho, características nativas das plataformas, etc.
  - API Python/C é bem simples
  - Diversos conversores automáticos (SWIG, SIP, ...)
  - **Jython**: usando Python em Java e vice-versa.
  - **PyRex**: pseudo linguagem para facilitar integração Python + C/C++.

## Características Importantes (2)

- **Tipos básicos poderosos:** listas, dicionários (*hash tables*), *strings*, ... otimizados e de fácil uso.
- **Baterias Inclusas:** biblioteca padrão contém diversos recursos úteis: Interface Gráfica (Tk), XML, Servidores (TCP, UDP, HTTP, ...), HTML, protocolos de internet (email, http, ...), xmlrpc, ...
- Grande base de código e bibliotecas de terceiros
- Grande comunidade de desenvolvedores
- **Software Livre:** liberdade de uso, distribuição. Licença própria, compatível com GPL, porém pode distribuir somente o binário.
- **Independente:** a entidade sem fins lucrativos Python Software Foundation cuida da propriedade intelectual do Python.

# No Brasil

- **Embratel:** monitoramento das interfaces de *backbone* e clientes de internet, também existem *scripts* de uso interno.
- **CPqD:** monitoramento de centrais telefônicas.
- **Async:** desenvolvimento de software de automação comercial
- **GPr Sistemas:** Desenvolvimento de aplicações sob encomenda, sistemas como monitoramento de transporte terrestre via satélite são as soluções já feitas
- **Outras** que utilizam o Python para sistemas Web, como Varig, Serpro, Câmara, Interlegis, ...

Os sistemas web de gestão de conteúdo usando o trio Python/Zope/Plone vem crescendo a cada dia, principalmente em empresas grandes e no governo.



## No Mundo

- **Industrial Light & Magic:** automação interna: *“Python plays a key role in our production pipeline. Without it a project the size of Star Wars: Episode II would have been very difficult to pull off. From crowd rendering to batch processing to compositing, Python binds all things together”*
- **NASA:** repositório de CAD/CAE/PDM, gerência de modelos, integração e sistema colaborativo: *“We chose Python because it provides maximum productivity, code that’s clear and easy to maintain, strong and extensive (and growing!) libraries, and excellent capabilities for integration with other applications on any platform.”*
- **University of Maryland:** ensino: *“I have the students learn Python in our undergraduate and graduate Semantic Web courses. Why? Because basically there’s nothing else with the flexibility and as many web libraries”*

## No Mundo (2)

- **Apple:** ferramenta padrão desde o MacOS X.
- **Microsoft:** investimento no IronPython para a plataforma .NET.
- **Disney:** jogos e Sistemas internos de automação e criação, patrocínio do PyQt.
- **Bank Boston:** sistema Web usando Python e Zope.
- **Nokia:** sistema de programação para celulares da série 60, permite mais recursos que o Java.
- **Atari:** jogos, como “Temple of Elemental Evil”.
- **Yahoo!:** Yahoo! Groups foi escrito inicialmente em puro python: 180.000 linhas de código cuidavam de tudo, tratando mais de 200 mensagens/segundo em um simples Pentium 400Mhz.

## No Mundo (3)

- **Nortel:** sistemas web “ChartWare”, “WebBook” e “WebTrack” são exemplos.
- **Philips:** automação da linha de semicondutores na fábrica de Fishkill.
- **Lawrence Livermore National Laboratories:** ambiente de engenharia numérica.
- **Red Hat:** diversas ferramentas para linux, o instalador das distribuições Red Hat e Fedora (“Anaconda”).
- **Gentoo Linux:** sistema de gerência de pacotes “Portage”.
- **Blender3D:** software pode ser estendido usando plugins Python.

# No Mundo: Google

Maior “case” Python da atualidade:

*“Python has been an important part of Google since the beginning, and remains so as the system grows and evolves. Today dozens of Google engineers use Python, and we’re looking for more people with skills in this language.”* — Peter Norvig, director of search quality at Google, Inc.

## No Mundo: Google (2)

- Sistema de ajuda do GMail
- Google Groups
- Sistema de compilação de aplicativos (*build system*).
- Sistema de empacotamento e entrega de dados (*packaging system*).
- Sistema de monitoramento e manutenção do cluster
- Sistema de testes
- Análise de registros (*logs*), ié análise de falsos clicks no *Sponsored Ads*.
- Prototipação
- Recentemente liberando código em <http://code.google.com>, como Goopy.
- Requisito para contratar profissionais Java: saber Python! :-)

## Bruce Eckel

Bruce Eckel é o autor de livros de renome, como “Thinking in Java” e “Thinking in C++”, mas hoje prefere pensar em Python. Ele apresentou 10 razões por que ele ama esta linguagem, algumas delas:

### ■ Python is about me

- Outras linguagens: *“Sim, nós estamos tentando fazer sua vida mais fácil com esta linguagem, mas estas coisas são mais importantes”.*
- Python: *“Nós tentamos fazer sua vida mais fácil, e é isto. Fazer sua vida mais fácil é o que não vamos comprometer.”.*
- **Exemplo:** C++ tentou fazer a vida do desenvolvedor mais fácil, porém comprometeram-se com performance e compatibilidade com C++, o que não ajudou em muito a vida do desenvolvedor. Qualquer problema do C++ pode ser explicado devido a estas duas características.
- **Exemplo:** Java achou que o marketing era mais importante.

## Bruce Eckel (2)

- **Menos Porcaria:** Dizem que uma pessoa armazena  $7 \pm 2$  informações ao mesmo tempo, quanto menos porcaria, melhor.
  - **Java:** 3 passos para abrir um arquivo. Muito mais para iterar sobre suas linhas.
  - **Python:** 1 chamada para abrir um arquivo. Construções fáceis para iterar sobre o conteúdo:  

```
for line in file('fname').readlines(): print line.
```
  - A falta de caracteres inúteis e a obrigatoriedade de uma indentação consistente também ajuda a se concentrar no problema.
- **Produtividade é mais importante que performance:** otimizar prematuramente é um grande erro.
  - A maioria dos programas não precisa de alta-performance
  - A maioria dos programas precisa de alta-produtividade
  - Dos programas que precisam de performance, a maioria se resolve com um *profiler* e otimizações algorítmicas.