



Programando em alto nível

Igor Sobreira

www.igorsobreira.com

Eu te conheço?

- Me chamo Igor Sobreira
- Faço TSI no IFPB (antigo CEFET-PB)
- Programo em Python a uns 3, 4 anos
- Trabalho com Python pra web (Django)
 - Smart Nuts Technology (www.smartnutstechnology.com)
- Velejo de windsurf (quando dá...)

Python começou ...

- ... em 1991, o primeiro release: 0.9.0
- Multi-paradigma
- Multi-plataforma
- Tipagem dinâmica e forte
- Biblioteca padrão extensa
- Fácil de aprender
- Preza pela beleza de código ... beleza?

Beleza Importa!



Código bonito é código legível

mais fácil de manter

VHLL

Linguagem de Altíssimo Nível



Para os humanos

```
>>> alunos = [ 'maria', 'pedro', 'fernanda', 'felipe' ]
>>> print 'SIM' if 'maria' in alunos else 'NÃO'
SIM
```

```
>>> for aluno in alunos:
...     print aluno.title()
...
Maria
Pedro
Fernanda
Felipe
```

Interpretador interativo

```
$ python
Python 2.6.1 (r261:67515, Dec 24 2008, 13:12:28)
[GCC 4.0.1 (Apple Inc. build 5465)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print "Olá ENSOL III"
Olá ENSOL III
>>> 1 + 2
3
>>>
```


Onde estamos?

Pra onde vamos?

- Python 3.0: quebra de compatibilidade
 - 2to3 ajuda na migração
 - ainda pouco usada (poucos módulos externos)
- Versão atual 2.6.2: compatível com a família 2.x
 - traz algumas novidades do Python 3.0
 - lança warnings de features depreciadas

Variáveis

A

C

T

G

Objetos.

Eles estão em todos os lugares

Eles estão em todos os lugares

Tipagem Dinâmica

“ ... executam em tempo de execução muitos comportamentos que outras linguagens realizam em tempo de compilação”

Wikipedia

- não é preciso declarar o tipo de uma variável
- mas é preciso inicializá-la

```
>>> a = 10          # `a` é um inteiro (int)
>>> a = "uma palavra" # agora, é uma string (str)
>>> b = a          # `b` é outro nome para `a`

>>> b == a        # `b` é igual a `a`
True
>>> b is a        # `b` é `a` (aponta para o mesmo objeto)
True

>>> print c        # `c` não foi inicializado
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'c' is not defined
```



... e Forte

```
>>> a = 12
```

```
>>> b = "30"
```

```
>>> a + b
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
>>> a + int(b)
```

```
42
```

○ que eu tenho disponível?



- listas [coleção, de, objetos]
- tuplas (listas, imutáveis)
- dicionários { chave: valor }
- “strings”
 - u’unicode’,
 - r’string cruas\n’

Baterias Inclusas

Quase tudo que você precisa vem embutido

<http://docs.python.org/library/>

re

StringIO

datetime

weakref

decimal

calendar

itertools

random

math

csv

os.path

os

pdb

robotparser

sqlite3

shelve

gzip

email

json

pickle

tarfile

xml.*

urllib/urllib2

hashlib

threading

multiprocessing



Multiplataforma



- Linux
- Unix
- Windows
- OS X
- Maemo - PyMaemo
- S60 - PyS60
- Android
- E também na JVM e .NET

Python

Implementação X Especificação

Implementação X Especificação

CPython

- Implementação de referência, oficial
- Escrita em C
- Versão atual: 3.0.1 / 2.6.2

Jython

- Python na JVM
- Implementação em Java
- Versão atual: 2.5.0

IronPython

- Python na .NET e Silverlight
- Versão atual: 2.0.1
 - Compatível com CPython 2.5

Stackless Python



- Um python “melhorado”
- Microthreads
 - Threads mais “leves”
 - Não são threads reais
- Um round robin scheduler embutido
executa as *tasklets*

PyPy



- “... *framework para produção de implementações de linguagens dinâmicas*”
- “*implementação da linguagem Python rápida, flexível e compatível, usando esse framework para permitir novas funcionalidades avançadas sem precisar codificar detalhes de baixo nível*”
- Python em (R)Python!
- Geração de um compilador JIT!!

Unladen Swallow

- Branch do CPython feito pelo Google
- Meta? ser 5x mais rápido que o CPython
- Como? usando o JIT da LLVM
- *Boa sorte google :-)*
- Desenvolvimento ativo

Programação Funcional

filter()

```
>>> numeros = range(1, 15)
>>> numeros
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
>>> def pares(n):
...     if n % 2 == 0:
...         return True
...     return False
...
>>> filter(pares, numeros)
[2, 4, 6, 8, 10, 12, 14]
```

zip()

```
>>> numeros = [1,2,3,4]
>>> letras = ['a','b','c','d']
>>> zip(numeros, letras)
[(1, 'a'), (2, 'b'), (3, 'c'), (4, 'd')]
```

E muito mais...

- Python tem um excelente suporte a programação funcional
- Muitas funções embutidas muito úteis
 - <http://docs.python.org/library/functions.html>
- functools, itertools
- O tratamento de parâmetros é muito flexível
- Funções são objetos de primeira classe
- Closures

Decorators

- Modificam, incrementam uma função ou método
- Fazem alguma coisa antes e/ou depois da função real
- Filtram parâmetros e/ou valores de retorno
- A partir do Python 2.6 podem decorar classes!

```
def Log(funcao):
    def wrapper(*args, **kwargs):
        print "Chamando: ", funcao.func_name
        print " - parametros: ", args, kwargs
        retorno = funcao(*args, **kwargs)
        print " - retorno: ", retorno
        return retorno
    return wrapper
```

`somar` é uma função decorada por `log`

```
@Log
```

```
def somar(a, b):
    return a + b
```

```
>>> somar(10, 20)
```

```
Chamando: somar
```

- parametros: (10, 20) {}
- retorno: 30

```
30
```

```
>>> somar(a=2, b=3)
```

```
Chamando: somar
```

- parametros: {} {'a': 2, 'b': 3}
- retorno: 5

```
5
```

Programação Orientada a Objetos

```
class Evento(object):
    ''' Representa um Evento de um tema qualquer '''
    num_eventos = 0
    def __init__(self, titulo, tema):
        self.titulo = titulo
        self.tema = tema
        self.palestras = []
        Evento.num_eventos += 1
        self.numero = Evento.num_eventos

    def adicionar_palestra(self, palestra):
        self.palestras.append(palestra)

    def comecar(self):
        for palestra in self.palestras:
            print "Começando a palestra: " + str(palestra)
```

```
>>> ensol = Evento('III ENSOL', 'Software Livre')
>>> ensol.adicionar_palestra('Python')
>>> ensol.adicionar_palestra('Django')
>>> ensol.comecar()
Começando a palestra: Python
Começando a palestra: Django
```

```
>>> ensol.numero
```

```
1
```

```
>>> Evento.num_eventos
```

```
1
```

```
>>> ensol4 = Evento('IV ENSOL', 'Software Livre')
```

```
>>> Evento.num_eventos
```

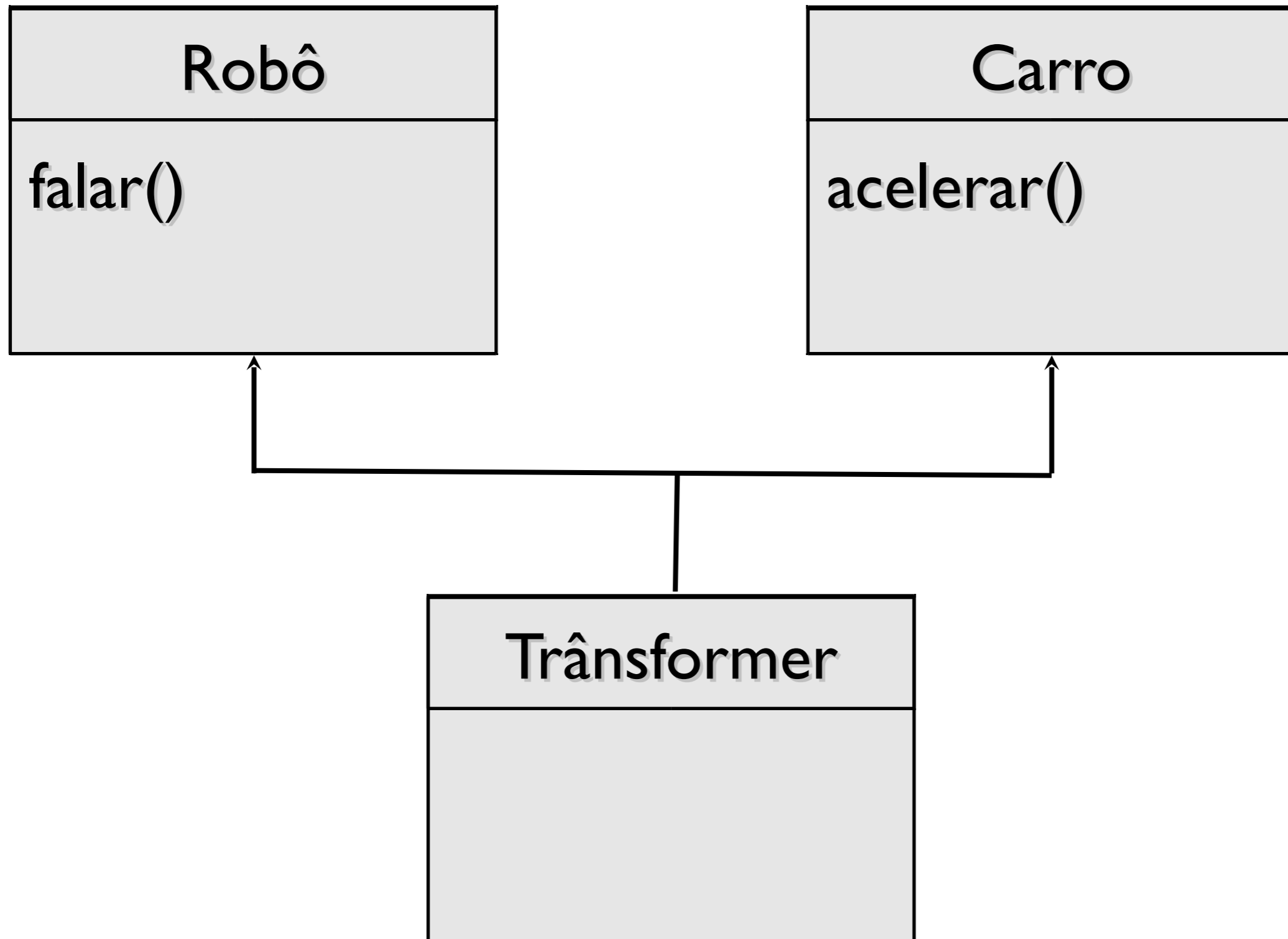
```
2
```

Métodos estáticos e de classe

```
@staticmethod  
def um_metodo():  
    ...
```

```
@classmethod  
def outro_metodo(cls):  
    ...
```

Herança Múltipla



Metaclasses

- Uma **classe** constrói **objetos**
- Uma **metaclasses** constrói **classes**

```
>>> a = [1,2]
>>> a.__class__
<type 'list'>

>>> list.__class__
<type 'type'>

>>> type.__class__
<type 'type'>
```



Classes Abstratas

- Novidade no Python 2.6
- Definem a interface de um objeto
- É uma forma de organizar as interfaces de suas classes, simplificando a reflexão
- Classes abstratas + herança múltipla torna desnecessário um tipo *interface*, como existe linguagens como Java e PHP

collections.Container

```
from abc import ABCMeta, abstractmethod

class Container:
    __metaclass__ = ABCMeta

    # permite que eu faça:
    # x in meu_container
    # mesmo que meu_container.__contains__(x)
    @abstractmethod
    def __contains__(self, x):
        return False
```

...

Properties

- Evite *getters* e *setters* desnecessários
- Use-os somente quando precisar
- E quando usar, seja transparente
- Não precisa *encapsular* algo que (ainda) não existe!

```
class Aluno(object):  
  
    def __init__(self, nome):  
        self.nome = nome
```

```
>>> p = Aluno("Pedro Silva")
```

```
>>> p.nome
```

```
'Pedro Silva'
```

```
>>> p.nome = 'Pedro Silveira'
```

```
class Aluno(object):

    def __init__(self, nome):
        self.primeiro_nome = ''
        self.ultimo_nome = ''
        self.nome = nome

    def _set_nome(self, nome):
        self.primeiro_nome = nome.split()[0]
        self.ultimo_nome = nome.split()[-1]

    def _get_nome(self):
        return self.primeiro_nome + " " + self.ultimo_nome

nome = property(_get_nome, _set_nome)
```

Isso é só começo ...

- OO em python é simples e poderoso!
- é fácil deixar seu objeto parecido com um tipo nativo
- sobrecarga de operadores (+, -, /, *)
- descriptors
- alto nível de reflexão

Persistência de Dados



- **pickle** e **cPickle**: serializa objetos
- **shelve**: objeto *dictionary-like* persistente
 - o valor pode ser qualquer objeto *pickleble*
- Unix **dbm**: pares chave-valor
 - chaves e valores são strings

Bancos Relacionais

- Python DB API (PEP 0242)
- sqlite3 (builtin!)
- MySQL - mysqldb
- PostgreSQL - psycopg2
- Oracle - cx_Oracle
- Microsoft SQL Server - pymssql

ORMs

- SQLAlchemy
- Django ORM
- DeJaVu
- Storm
- SQLAlchemyObject
- Autumm

Banco de Dados OO

- ZODB
- Durus
- pod - Pickled Object Database



CouchDB
relax

couchdb-python

<http://code.google.com/p/couchdb-python/>

Programação Concorrente



subprocess

- iniciar processos e analisar *input/output/error*
- `subprocess.call(['ls', '-la'])`

threading

- Fácil manipulação de threads
- Semáforos
- Locks
- Eventos



GIL

Global Interpreter Lock

- Não permite que duas threads executem em paralelo na mesma VM!!
- threads são concorrentes, mas não paralelas
- facilita muito a implementação, e módulos externos em C/C++
- Paralelismo real? Use processos.

multiprocessing

- quase a mesma interface de threading
- mas com processos, e não threads
- abstrai criação, interação entre processos ...

Web 2.0

Aggregators Folksonomy Wikis User Centered Joy of Use
Blogs Participation Six Degrees Usability Widgets
Pagerank XFN Social Software FOAF Browser
Recommendation Sharing Collaboration Perpetual Beta Simplicity AJAX
Videocasting Podcasting Audio IM Video Design
Convergence CSS Pay Per Click
UMTS Mobility Atom XHTML SVG Ruby on Rails VC Trust Affiliation
OpenAPIs RSS Semantic Web Standards SEO Economy
OpenID Remixability REST Standardization The Long Tail
DataDriven Accessibility SOAP Microformats Syndication XML
Modularity

- Desenvolvimento web: frameworks
 - frameworks *full-stack* ou componentes separados
- Web Services
 - python conversa com: JSON, XML, YAML
 - SOAP, RPC, REST

Fazendo uma busca no Web Service do twitter.com

```
>>> from urllib2 import urlopen
>>> import json

>>> data = urlopen('http://search.twitter.com/search.json?q=igorsobreira').read()
>>> json.loads(data)['results'][1]['text']
u'Tendo uma aula sobre closures na lista #pythonbrasil'
```

Testes Automatizados

doctests

- as docstrings viram testes reais
- documentação e teste num só lugar
- basta simular uma sessão no interpretador interativo (>>>)


```
#-*- coding: utf-8 -*-
```

```
def fatorial(n):
```

```
    '''
```

```
    Retorna o fatorial de um número `n`
```

```
    >>> fatorial(0)
```

```
    1
```

```
    >>> fatorial(1)
```

```
    1
```

```
    >>> fatorial(3)
```

```
    6
```

```
    >>> fatorial(6)
```

```
    720
```

```
    '''
```

```
    return n * fatorial(n-1) if n > 1 else 1
```

```
if __name__ == '__main__':
```

```
    import doctest
```

```
    print doctest.testmod()
```

unittest

- Testes automatizados
- A versão em Python do JUnit
- Sua aplicação é formada por pequenas unidades testáveis

```
import random
import unittest

class TestSequenceFunctions(unittest.TestCase):

    def setUp(self):
        self.seq = range(10)

    def testshuffle(self):
        # make sure the shuffled sequence does not lose any elements
        random.shuffle(self.seq)
        self.seq.sort()
        self.assertEqual(self.seq, range(10))

    def testchoice(self):
        element = random.choice(self.seq)
        self.assert_(element in self.seq)

if __name__ == '__main__':
    unittest.main()
```

E por fora?



- Nose
- py.test

*Eles procuram e rodas os testes para você.
E ainda estendem o unittest padrão*



- python-mock
- minimock

*Não teste objetos
externos.*

*Falsifique-os, e espere o
melhor deles.*

Ou simplesmente.
Monkeypatch :-)

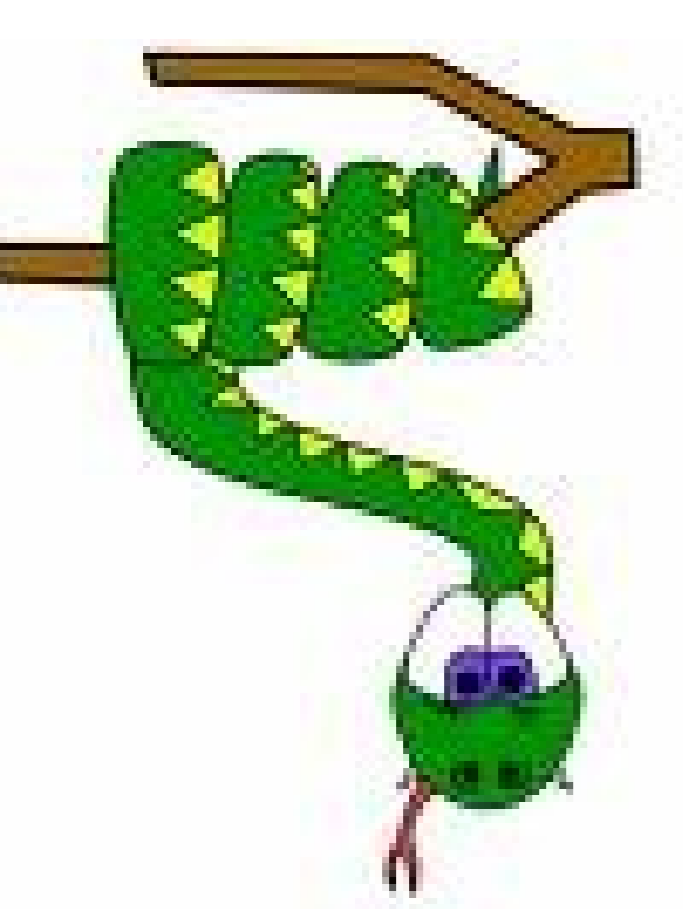


Conversa com outras linguagens


- C/C++
- Jython (Java)
- IronPython (.NET)

Onde usar?

- Web
- GUI
- Mobile
- Games e 3D
- Comp. Gráfica
- Ciência e Educação
- Rede
- Adm. de Sistemas



Quem usa?

 **Vídeos assistidos agora** (exibir todos)

editar   



President Obama kills a fly during an interview

16791 exibições
2frogmedia
★★★★★



Casate y Veras- Tarjeta

5205 exibições
vnevisionintl
★★★★★




CQC Oscar Filho com os BBB s cqc 15/06/2009

6586 exibições
videolim2
★★★★★



Beyoncé - Ego Remix (feat. Kanye West)

54161 exibições
xCurves
★★★★★

 **Vídeos em destaque**



Maria Rita com Quinteto em Branc...

1 mês atrás
474120 exibições
TramaRadiola
★★★★★



pobre millonaria EPS5 (Parte 3)

1 dia atrás
1175 exibições
vnevisionintl
★★★★★



Tehran Fired Up

3 dias atrás
132402 exibições
CBS
sem avaliação





The Mp3 Experiment Six

3 dias atrás
241680 exibições
ImprovEverywhere
★★★★★

[▶ CLIQUE e confira vídeos exclusivos do campeonato.](#)



0:00 / 0:31  



O futebol nos une: Copa Santander Libertadores

★★★★★

Publicidade

Deseja personalizar esta página inicial?
[Faça login](#) ou [inscreva-se](#) agora!

O que há de novo



YouTube em alta definição

Assista a seus vídeos preferidos em alta definição.



CENTRAL INTELLIGENCE AGENCY

THE WORK OF A NATION. THE CENTER OF INTELLIGENCE.



[About CIA](#)

[Careers](#)

[Offices of CIA](#)

[News & Information](#)

[Library](#)

[Kids' Page](#)

[Contact CIA](#)



[Play Movie](#) >

[Text Version](#) > | [Get Flash Player](#) >

[Iraqi Rewards Program](#)

[المكافآت](#)

Quick Links

- [World Factbook](#)
- [World Leaders](#)
- [Today's CIA](#)
- [Studies in Intelligence](#)
- [Diversity](#)
- [RSS](#) 

Mission

The Central Intelligence Agency (CIA) is an independent US Government agency responsible for providing national security intelligence to senior US policymakers.

For more on the Agency's mission, visit our [Strategic Intent](#).

Featured Story

[CIA World Factbook Launches New Web Site](#)

On Monday, June 8, the *World Factbook* team unveiled its thoroughly redesigned Web site. Presenting a cleaner look, improved navigation, and a host of added features, the new site provides visitors with a more user-friendly experience. More than 3 million visitors access the online *Factbook* monthly. The world-renowned *Factbook* provides visitors with wide-ranging and hard-to-locate information about the background, geography, people, government, economy, communications, transportation, military, and transnational issues for 266 countries and other entities. [more >](#)

What's New

June 11 - Posted new college student positions to CIA Careers.

June 11 - Updated *Chiefs of State and Cabinet Members of Foreign Governments*. [more >](#)

June 1 - CIA Pays Tribute to Fallen Officers.

May 29 - Message from the Director: Strengthening Our Foreign Language Capability.

[\[View all recent updates\]](#)

Showcase

Take a step inside the CIA through our [Headquarters Tour](#). Stroll down the [CIA Museum](#) corridor to see some of our artifacts. And learn about the CIA's history in our [Interactive Timeline](#).

[Privacy](#) [Copyright](#) [Site Policies](#) [USA.Gov](#) [FOIA](#) [DNI.Gov](#) [NoFEAR Act](#)

[Contact CIA](#) [Site Map](#)



Notícias

Mais Cultura seleciona oito projetos de TV sobre a juventude brasileira

Expansão da educação profissional abre 38 mil vagas

Acervo do Museu Nacional começa a ser digitalizado

CGU e UnB realizam evento para debater combate à corrupção no Brasil

[Veja a lista completa »](#)

Em Questão

Lei Seca reduz internações e mortes em mais de 20%

Serviços mais procurados

- » [Consulta Situação Cadastral Pessoa Física - CPF](#)
- » [Aposentadoria por idade](#)
- » [Andamento de processos de concessão inicial de benefícios](#)
- » [Aposentadoria especial](#)
- » [Andamento de processos de revisão de benefícios](#)
- » [Auxílio-doença](#)
- » [Concursos](#)
- » [Serviço de informações científicas em saúde pública](#)
- » [Aposentadoria por invalidez](#)
- » [Bulário Eletrônico](#)

[Veja mais serviços »](#)

Confiança no Brasil



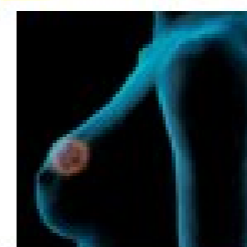
[Conheça a Campanha](#)

Campanha Bibliotecas



Sua cidade não tem biblioteca pública municipal? Informe-se aqui.

Ciência e Saúde



Nova tecnologia ajuda no diagnóstico do câncer de mama



Imóveis à venda

Imóveis para alugar

Buscar imóveis em

BUSCAR

ex.: "Itaim Bibi, São Paulo, SP", "Rio de Janeiro, RJ", "Porto Alegre, RS"...

Joao Pessoa, PB - 597 imóveis encontrados

Calcule seu
consórcio

CAIXA
CONSÓRCIOS

Calcule seu
seguro residencial

CAIXA
SEGUROS



VEN. A HORA
É AGORA.

CLIQUE.

Amplia mapa

Refine sua busca

[Começar de novo](#)

Outras Localidades

[Adicionar outra localidade](#)

Escolha a distância

Situação do Imóvel

- Todos
- Revendas
- Lançamentos

Preço

[Outras faixas de preço](#)

Tipo de Imóvel

[Selecione vários tipos](#)

[Quartos](#) [Banheiros](#)

Ordene por

Página 1 de 40

< ANT

PRÓX >

Bessa

fnobrega.com



R\$ 265.000,00

Apartamento

[mais detalhes](#) | [mostrar no mapa](#) | [imóveis próximos](#) | [escolas e hospitais próximos](#)

Manaira

fnobrega.com



R\$ 350.000,00

Casa

[mais detalhes](#) | [mostrar no mapa](#) | [imóveis próximos](#) | [escolas e hospitais próximos](#)

Bessa

fnobrega.com



R\$ 190.000,00

Apartamento

[mais detalhes](#) | [mostrar no mapa](#) | [imóveis próximos](#) | [escolas e hospitais próximos](#)

Agua Fria

fnobrega.com



R\$ 95.000,00

Map Satellite Hybrid



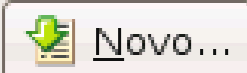
Clique em  para ver detalhes dos imóveis.
Use   para fazer zoom.


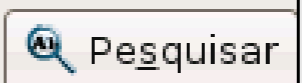

Call Center - Rio de Janeiro

Aync - Stoq. Gestão comercial





Stoq - Ponto de Vendas Admin

Vendas Pesquisar Caixa Ajuda

 Novo... Número do Pedido: 00009 Cliente: Alessandra Almeida It... Vendedor: Aync Open Source

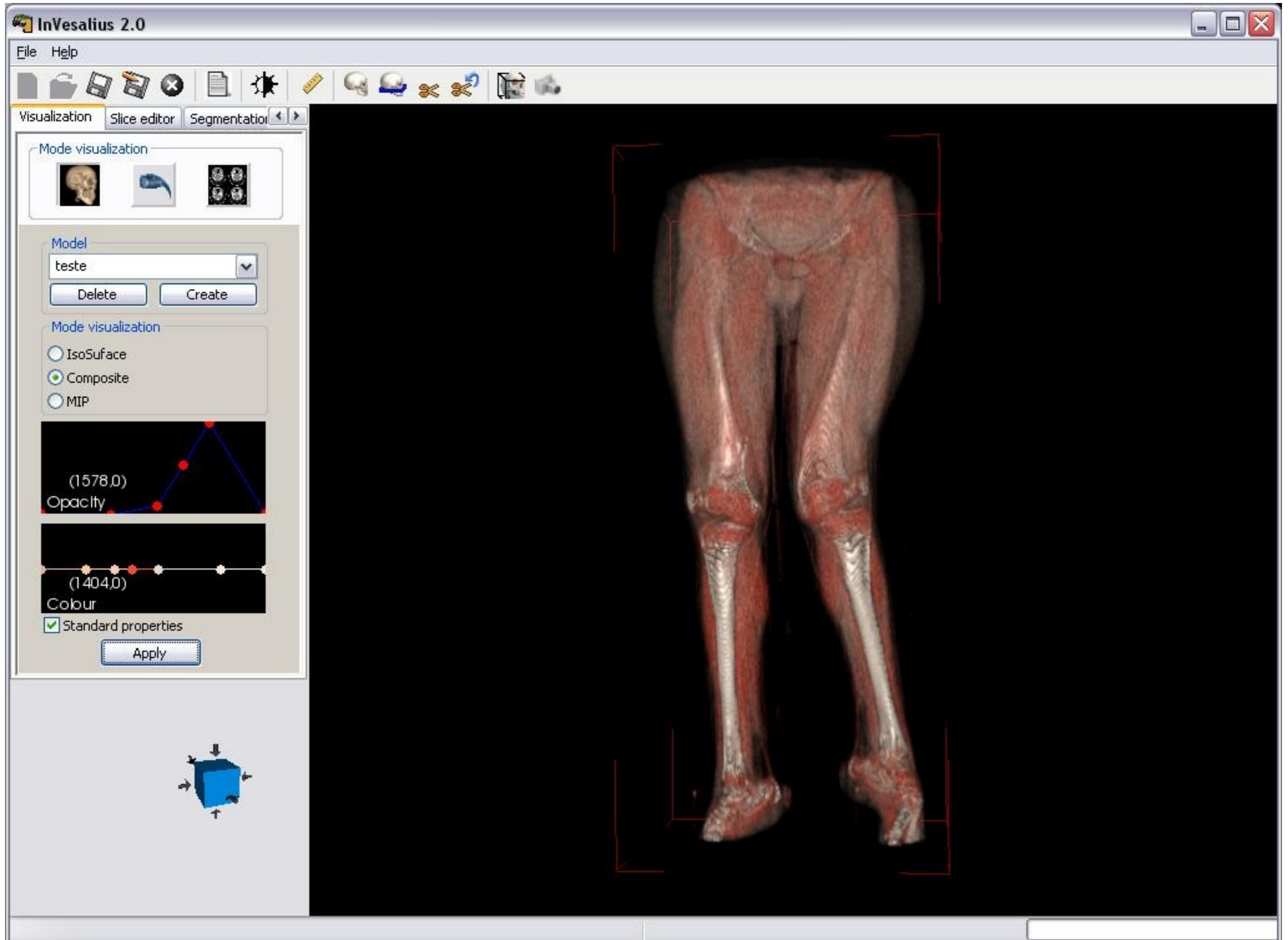
Código de Barras: Quantidade:  Adicionar  Pesquisar 

Referência	Descrição	Preço	Quantidade	Unidac	Total
00002	Bermuda Sarja	R\$ 149,00	2,0		R\$ 298,00
00004	Camiseta Tinturada Xcuba Comp	R\$ 89,00	1,0		R\$ 89,00
00008	Casaco Vilan	R\$ 503,00	1,0		R\$ 503,00

 Editar  Remover  Entrega  Confirmar

Total: R\$ 890,00

InVesalius



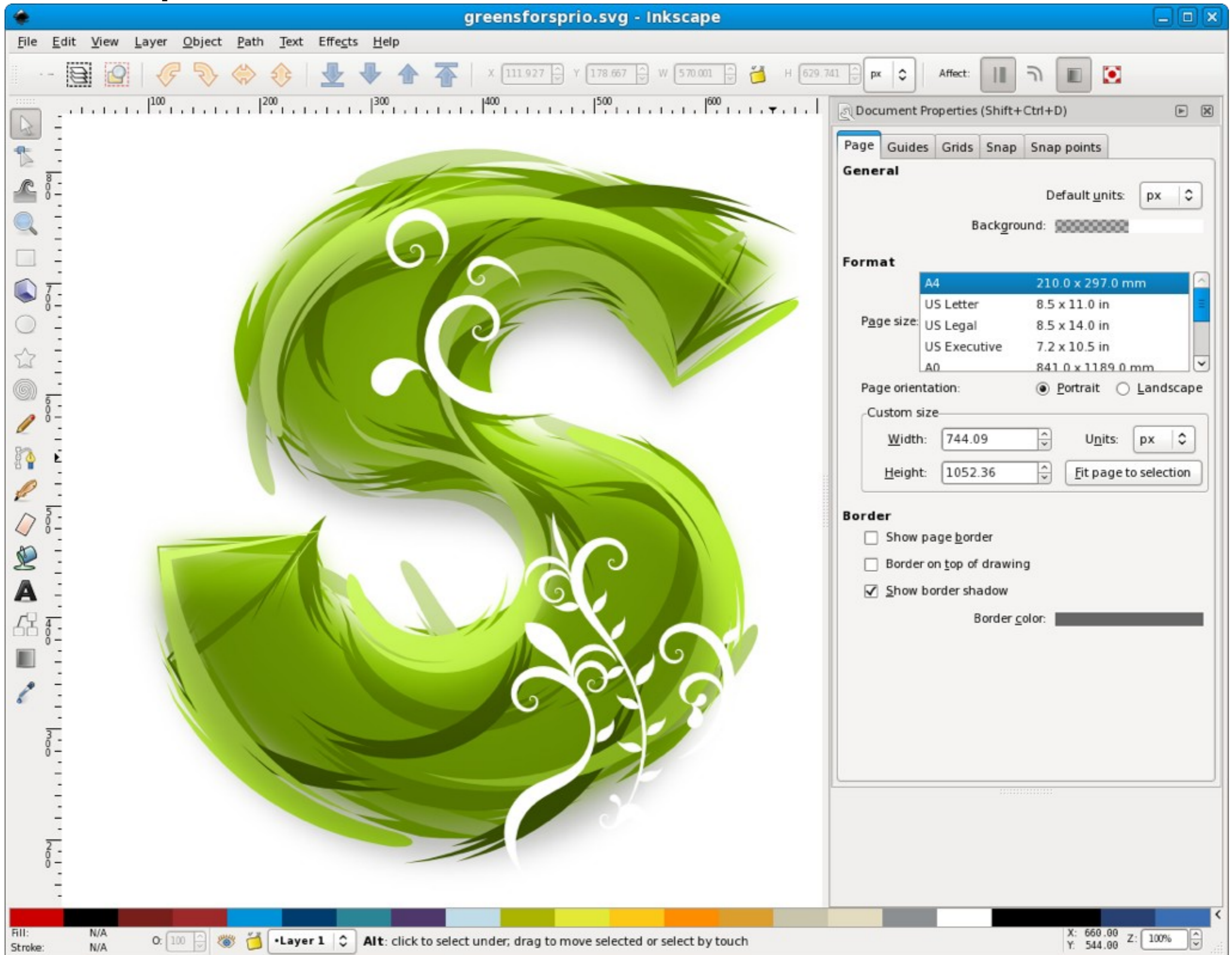
Blender

The screenshot displays the Blender 2.36 interface with a 3D scene of a character named Margarito sitting at a table. The interface is divided into several panels:

- Top Panel:** Contains the 3D Viewport, Transform Properties, and View Properties. The 3D Viewport shows a character in a green shirt and blue pants sitting at a wooden table. The Transform Properties panel shows the selected object's position (X: 0.88, Y: 0.00, Z: -4.37) and rotation (X: 0.00, Y: 0.00, Z: -4.37).
- Right Panel:** Contains the Outliner and Properties panels. The Outliner shows a list of objects including Margarito, Botella, Camera.001, Circle.001, Cube.001, Empty, Obipo.005, Armature, and Margarito. The Properties panel shows the selected object's properties, including location (LocX, LocY, LocZ), rotation (QuatW, QuatX, QuatY, QuatZ), and size (SizeX, SizeY, SizeZ).
- Bottom Panel:** Contains the Timeline and Action Editor. The Timeline shows a sequence of frames from 0 to 600. The Action Editor shows a list of actions including Action, Armature, and Margarito.
- Left Panel:** Contains the Properties panel for the selected object, showing material and ramp properties. The material properties include color (R: 1.000, G: 0.336, B: 0.155) and texture (Ramp).

The interface also includes a top menu bar with File, Add, Timeline, Game, Render, and Help. The status bar at the bottom shows the current frame (607) and various system information.

Inkscape



INdT - Nokia





Documentação

Download

Download these documents

Docs for other versions

[Python 3.0 \(stable\)](#)
[Python 2.7 \(in development\)](#)
[Python 3.1 \(in development\)](#)
[Old versions](#)

Other resources

[FAQs](#)
[Guido's Essays](#)
[New-style Classes](#)
[PEP Index](#)
[Beginner's Guide](#)
[Book List](#)
[Audio/Visual Talks](#)
[Other Doc Collections](#)

Quick search

Enter search terms or a module, class or function name.

Python v2.6.2 documentation

Welcome! This is the documentation for Python 2.6.2, last updated Jun 19, 2009.

Parts of the documentation:

What's new in Python 2.6?

or all "What's new" documents since 2.0

Tutorial

start here

Using Python

how to use Python on different platforms

Library Reference

keep this under your pillow

Language Reference

describes syntax and language elements

Python HOWTOs

in-depth documents on specific topics

Extending and Embedding

tutorial for C/C++ programmers

Python/C API

reference for C/C++ programmers

Installing Python Modules

information for installers & sys-admins

Distributing Python Modules

sharing modules with others

Documenting Python

guide for documentation authors

Indices and tables:

Global Module Index

quick access to all modules

General Index

all functions, classes, terms

Glossary

the most important terms explained

Search page

search this documentation

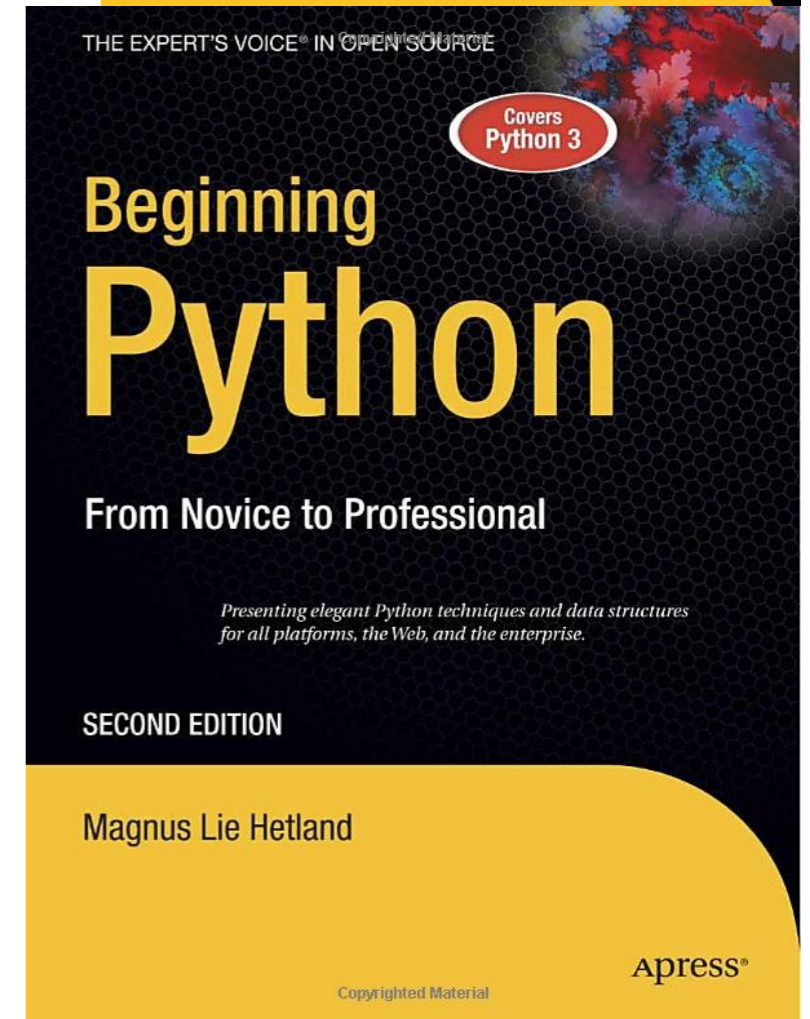
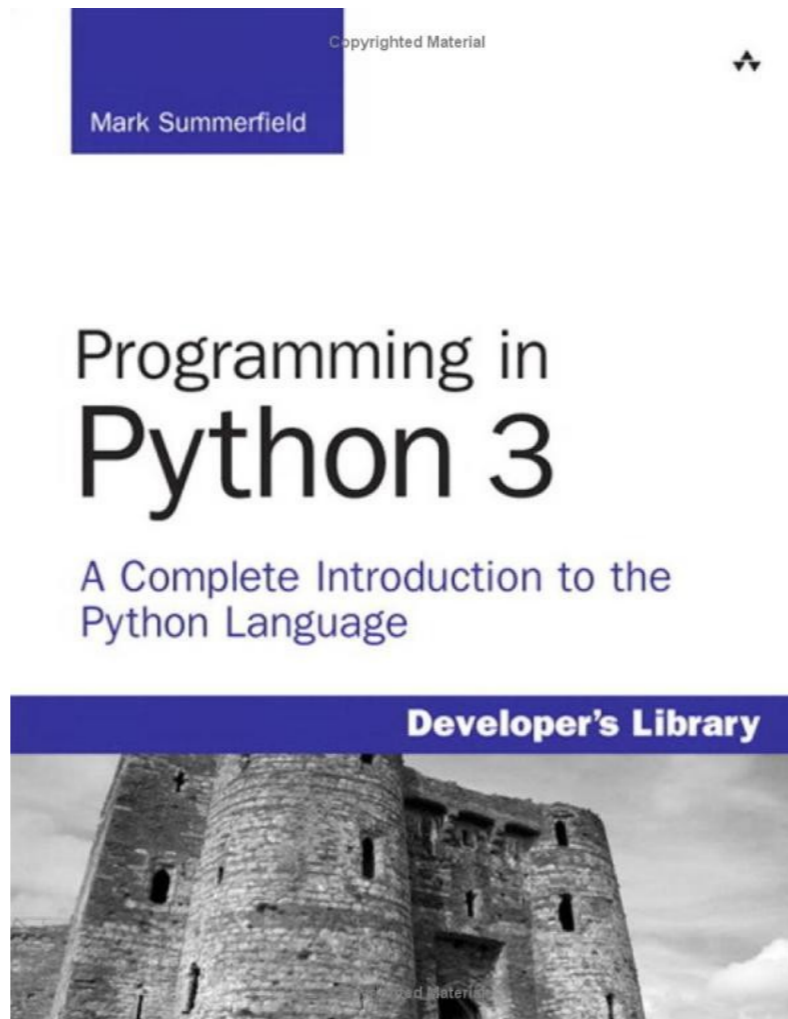
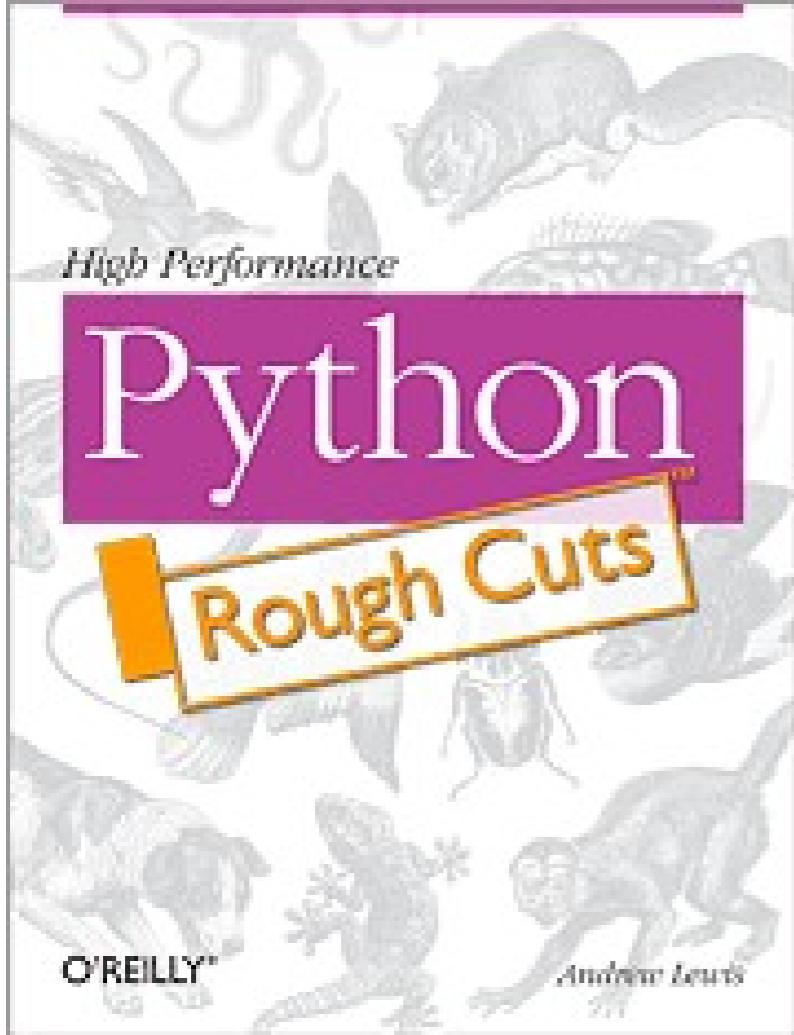
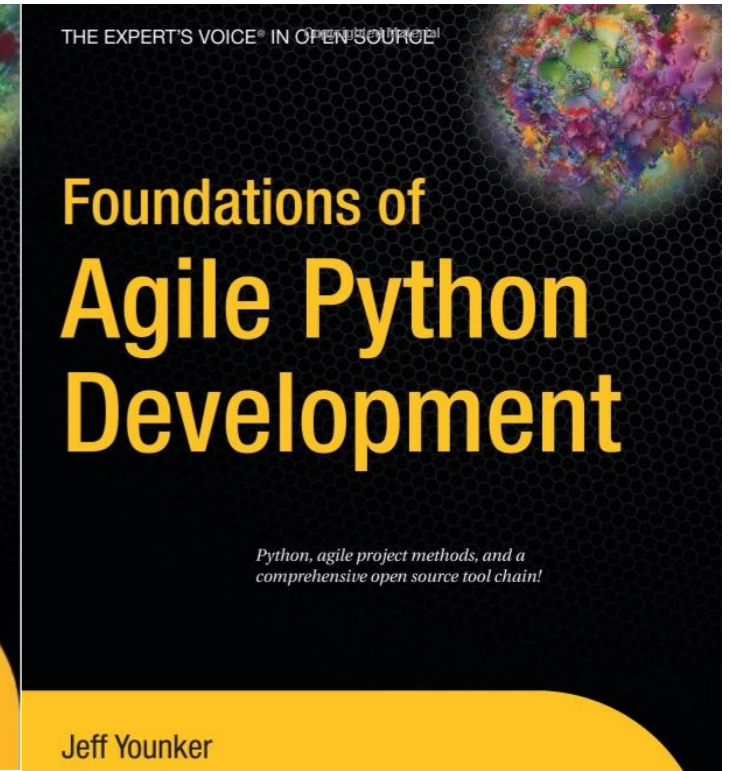
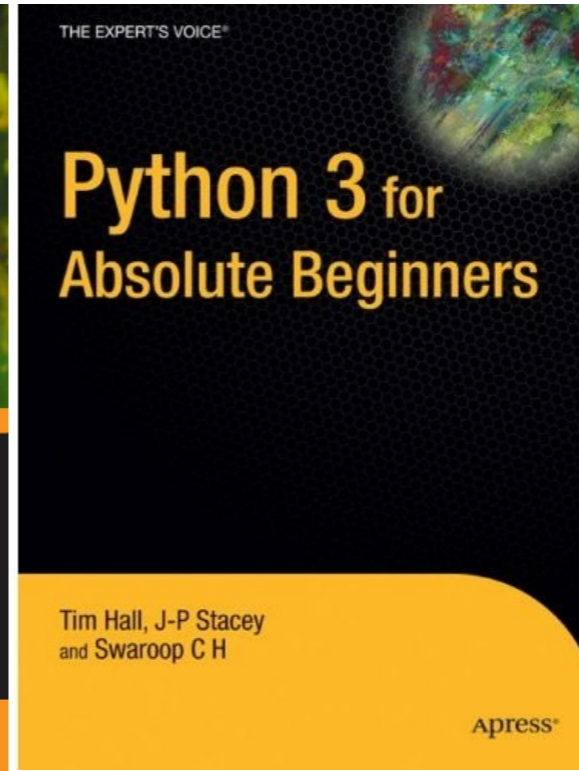
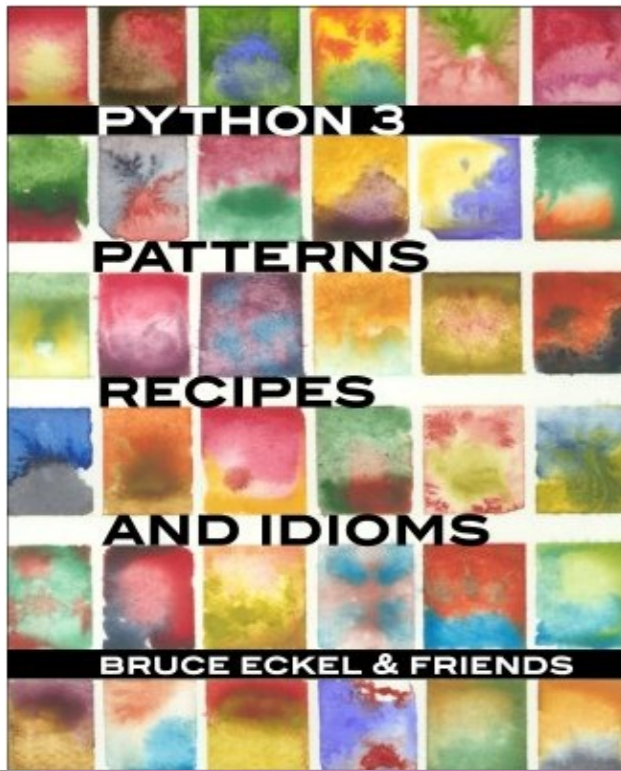
Complete Table of Contents

lists all sections and subsections

Meta information:

[python.org.br](#)

[docs.python.org](#)





python™

Para desenvolvedores



Python para desenvolvedores

<http://ark4n.wordpress.com/python/>



A Comunidade



Encontro Brasileiro

- I PyconBrasil (2005) Unicamp, São Paulo/SP
- II PyconBrasil (2006) Interlegis, Brasília/DF
- III PyconBrasil (2007) SOCIESC, Joinville/SC
- Pyconbrasil [4] (2008) UVA, Rio de Janeiro/RJ
- **Pythonbrasil [5] UCS, Caxias do Sul/RS**
 - **10 a 12 de Setembro de 2009**

- python.org.br
- Grupos
 - python-brasil - 2960
 - django-brasil - 1034
 - zope-pt - 1047
- IRC
 - #python-br
 - #django-br
 - #qualquer-ferramenta-ou-framework



**Bem vindo ao mundo
Python**

Obrigado. Dúvidas?

Igor Sobreira
www.igorsobreira.com