

Hagora para algo
completamente
diferente:

Python



E agora para algo completamente diferente. . .

Carlos Moraes dos Santos

Eduardo de Oliveira Padoan



Mainstream

Se você sabe alguma coisa de programação, você provavelmente conhece a maneira **Mainstream** de escrever.

- Declarar **variáveis**
- Realizar **operações**
- Usar **estruturas**

Variáveis

- `Var str1, str2 : String[30];`
 `int1, int2 : Integer;`
 `arr1 : Array of Byte[0..3];`
- `char[30] str1, str2;`
 `int int1 = 3, int2;`
 `int[3] arr1;`

Variáveis são usadas para armazenar **valores** e coisas assim.

Operações

- `foo := ((4 + i) / 3) * (a - (n MOD 2));`
- `foo = ((4 + i) / 3) * (a - (n % 2));`

Operações são realizadas entre variáveis e/ou constantes por meio de **operadores**.

Estrutura condicional simples

- ```
if i = n then
 // do something in Pascal
else
 // do another something in Pascal
```
- ```
if (i == n)
    // do something in C
else
    // do another something in C
```

Uma estrutura condicional simples permite executar um bloco de instruções caso uma expressão seja avaliada como verdadeira, ou, caso contrário, outro bloco.

Estrutura condicional composta

- `case n of`
 `3 : // do something in Pascal`
 `4 : // do another something in Pascal`
 `otherwise // do a default action in Pascal`
`end;`
- `switch(n) {`
 `case 3 : // do something in C`
 `case 4 : // do another something in C`
 `default : //do a default action in C`
`}`

Uma estrutura condicional composta permite especificar um bloco de instruções para cada condição em algum elemento.

Estrutura de repetição **for**

- `for i:=0 to 99 do`
 `// do something in Pascal`
- `for {i = 0; i < 99; i++}`
 `// do something in C`

Um loop for convencional itera por uma faixa de valores (não necessariamente) inteiros.

Estrutura de repetição **condicional**

- `while n < 99`
 `// do something in Pascal`
- `repeat`
 `// do something in Pascal`
 `until n > 99;`
- `while (n < 99)`
 `// do something in C`
- `do`
 `// do something in C`
 `while (n < 99)`

Um loop condicional convencional é executado enquanto uma determinada expressão for avaliada como verdadeira.



Ok, hora de começar com a palestra.

H agora
para algo
completamente
diferente...



Porque python é diferente?

A linguagem python também tem “variáveis”, operadores, estruturas condicionais e de repetição.

Mas, por que fazê-los da mesma maneira que os outros?



RTFM (Read The Fine Manual)

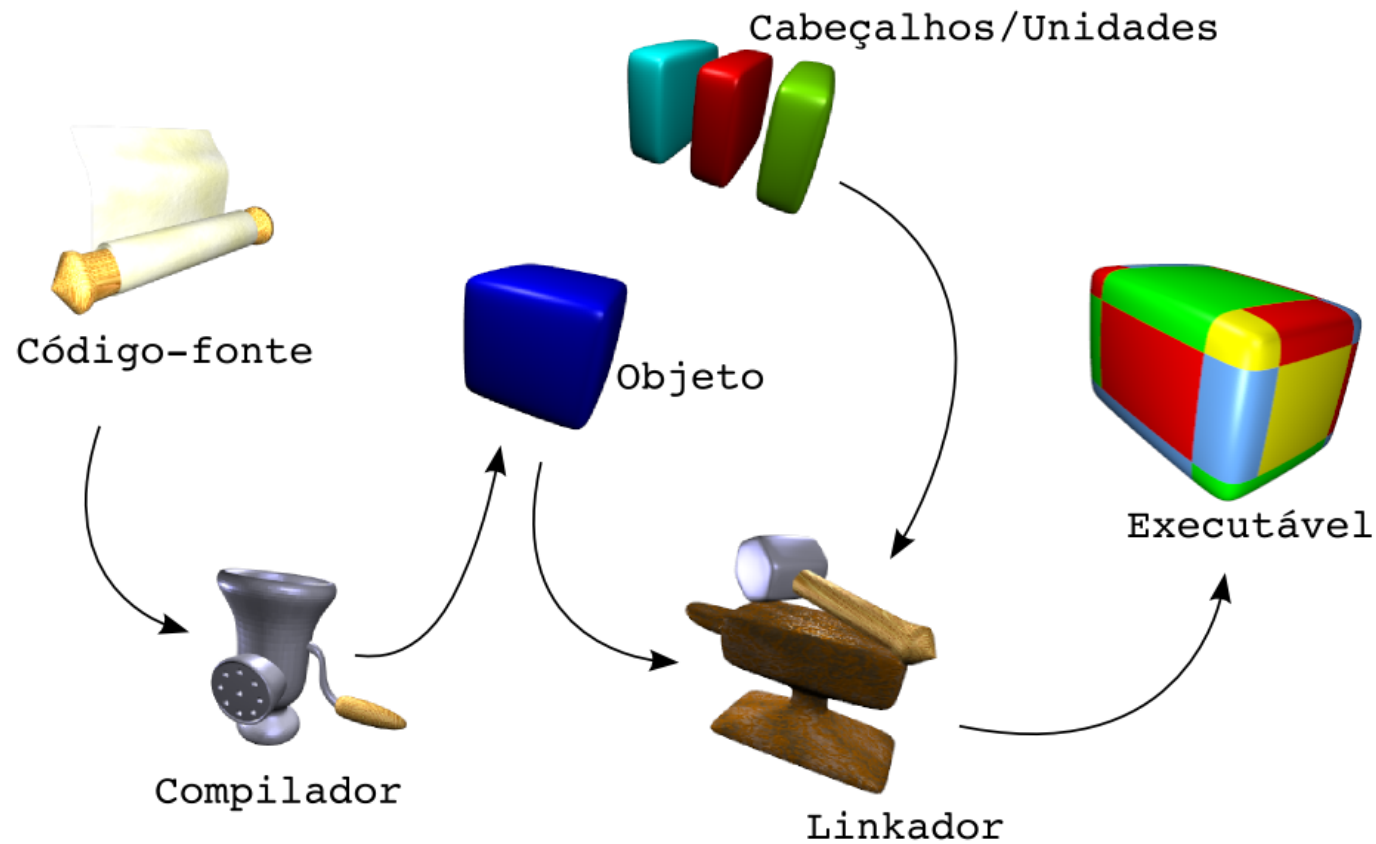
Na manpage do python. . .

```
knightstalker@farnheit[~]$ wtf python
```

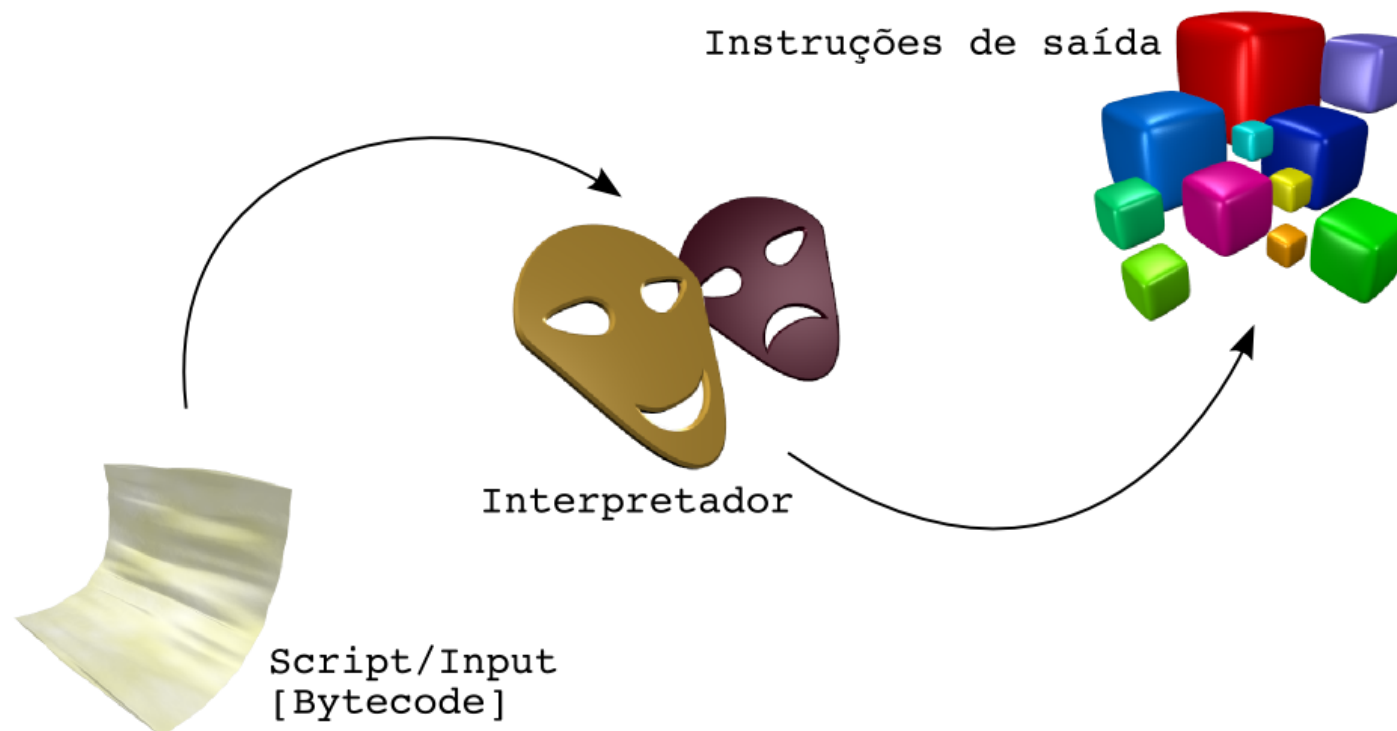
DESCRIPTION

Python is an interpreted, interactive, object-oriented programming language that combines remarkable power with very clear syntax. For an introduction to programming in Python you are referred to the Python Tutorial. The Python Library Reference documents built-in and standard types, constants, functions and modules. Finally, the Python Reference Manual describes the syntax and semantics

#Compiled

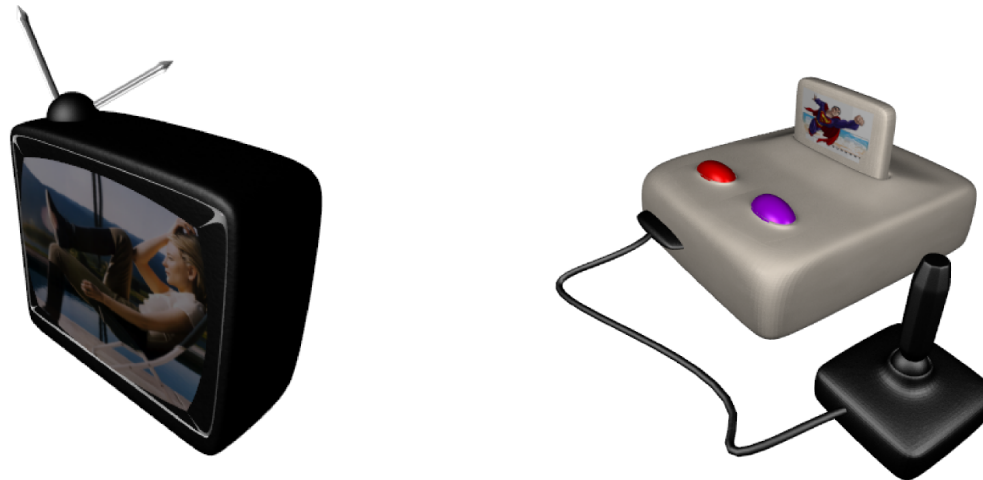


Interpreted



Interactive

- Shell
- Programação passo-a-passo
- Não necessariamente interativo(scripts e bytecodes)



Object-oriented

- Objeto

Agrega atributos(características) e métodos(ações).

- Classe

“templates” para instanciação dos objetos(“novos” objetos).

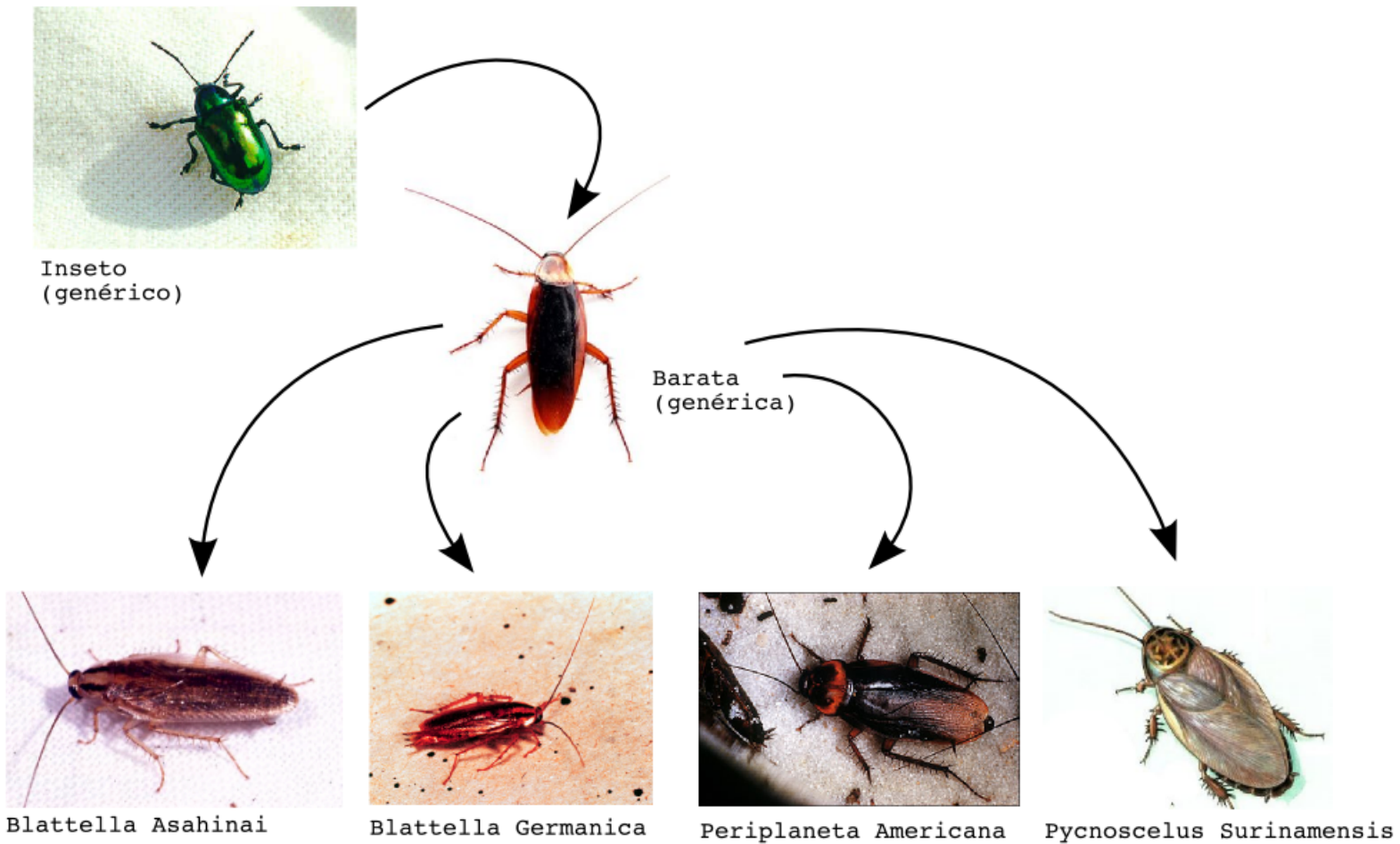


Object-oriented

O que é uma

Periplaneta americana?

Object-oriented



Object-oriented

**Tem um cara na platéia
querendo pisar na gente...**



Object-oriented

De fato, python é uma linguagem fortemente orientada a objetos;

- Inteiros são objetos
- Strings são objetos
- Floats são objetos
- Funções são objetos
- Classes são objetos



Remarkable Power

- Um design compacto que reúne praticamente todo o necessário
- Standard Library
- Modularidade e extensibilidade
- Funções built-in
- Estruturas de dados de alto nível



Very Clear Syntax

99 garrafas de cerveja no muro em Perl

```
$nBottles = $ARGV[0];  
$nBottles = 100 if $nBottles eq '' || $nBottles < 0;  
  
foreach (reverse(1 .. $nBottles)) {  
    $s = ($_ == 1) ? "" : "s";  
    $oneLessS = ($_ == 2) ? "" : "s";  
    print "\n$_ bottle$s of beer on the wall,\n";  
    print "$_ bottle$s of beer,\n";  
    print "Take one down, pass it around,\n";  
    print $_ - 1, " bottle$oneLessS of beer on the wall\n";  
}
```



Very Clear Syntax

99 garrafas de cerveja no muro em C++ (orientado a objetos)

```
#include <fstream.h>

enum Bottle { BeerBottle };

class Shelf {
    unsigned BottlesLeft;
public:
    Shelf( unsigned bottlesbought )
        : BottlesLeft( bottlesbought )
    {}
    void TakeOneDown()
    {
        if (!BottlesLeft)
            throw BeerBottle;
        BottlesLeft--;
    }
    operator int () { return BottlesLeft; }
};
```



```
int main( int, char ** )
{
    Shelf Beer(99);
    try {
        for (;;) {
            char *plural = (int)Beer !=1 ? "s" : "";
            cout << (int)Beer << " bottle" << plural
                << " of beer on the wall," << endl;
            cout << (int)Beer << " bottle" << plural
                << " of beer," << endl;
            Beer.TakeOneDown();
            cout << "Take one down, pass it around," << endl;
            plural = (int)Beer !=1 ? "s":"";
            cout << (int)Beer << " bottle" << plural
                << " of beer on the wall." << endl;
        }
    }
    catch ( Bottle ) {
        cout << "Go to the store and buy some more," << endl;
        cout << "99 bottles of beer on the wall." << endl;
    }
    return 0;
}
```




Very Clear Syntax

99 garrafas de cerveja no muro em python

```
def bottle(n):
    try:
        return { 0: "no more bottles",
                  1: "1 bottle"} [n] + " of beer"
    except KeyError: return "%d bottles of beer" % n

for i in range(99, 0, -1):
    b1, b0 = bottle(i), bottle(i-1)
    print "%(b1)s on the wall, %(b1)s,\n"\
          "take one down, pass it around,\n"\
          "%(b0)s on the wall." % locals()
```



De novo

I agora
para algo
completamente
diferente...

Usando python. . .



O shell

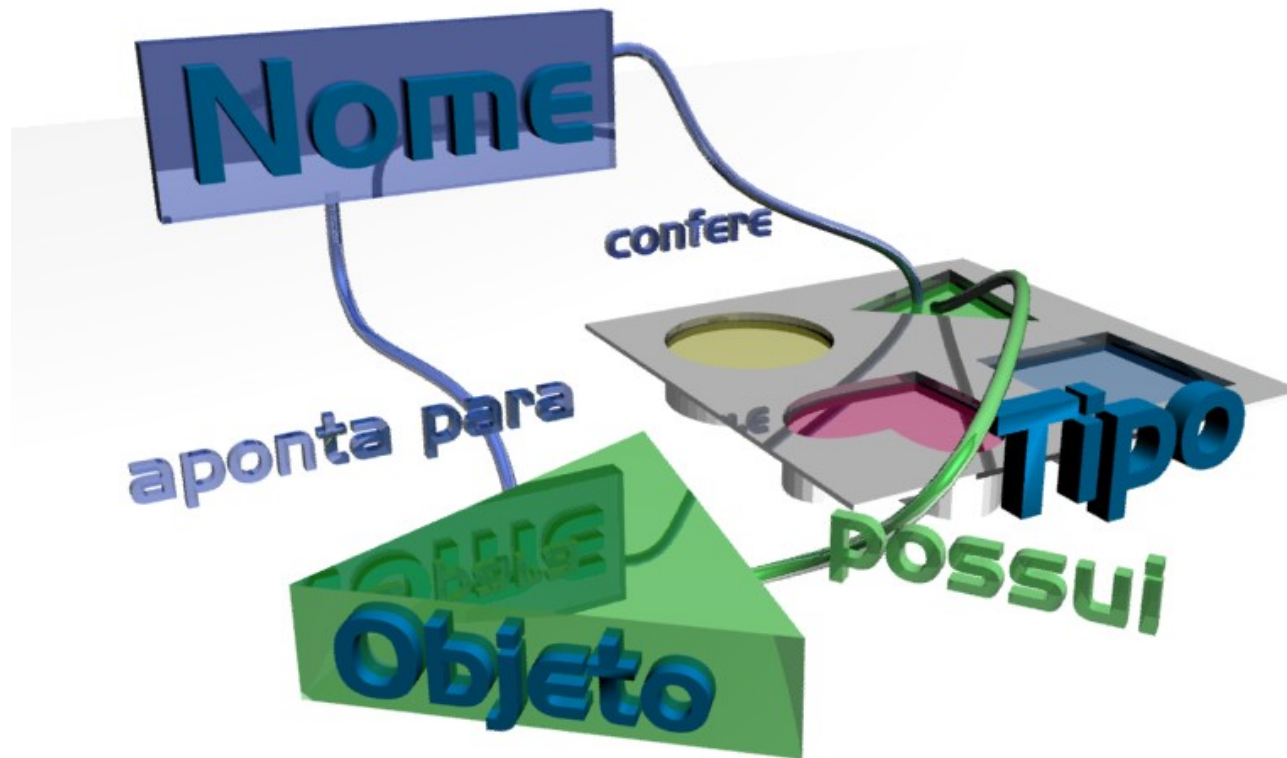
IDLE(Integrated DeveLopment Environment) é uma IDE que acompanha a instalação padrão do python.



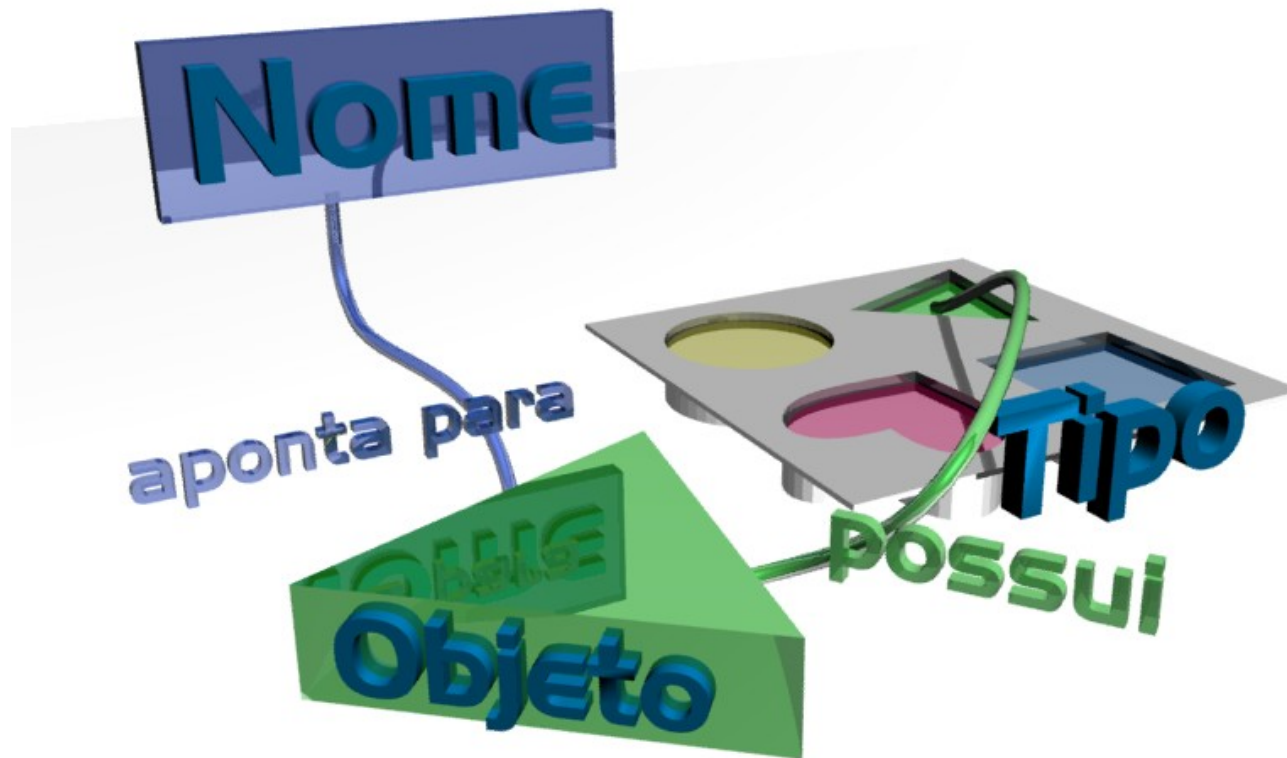
Aí vamos nós

Say hello!

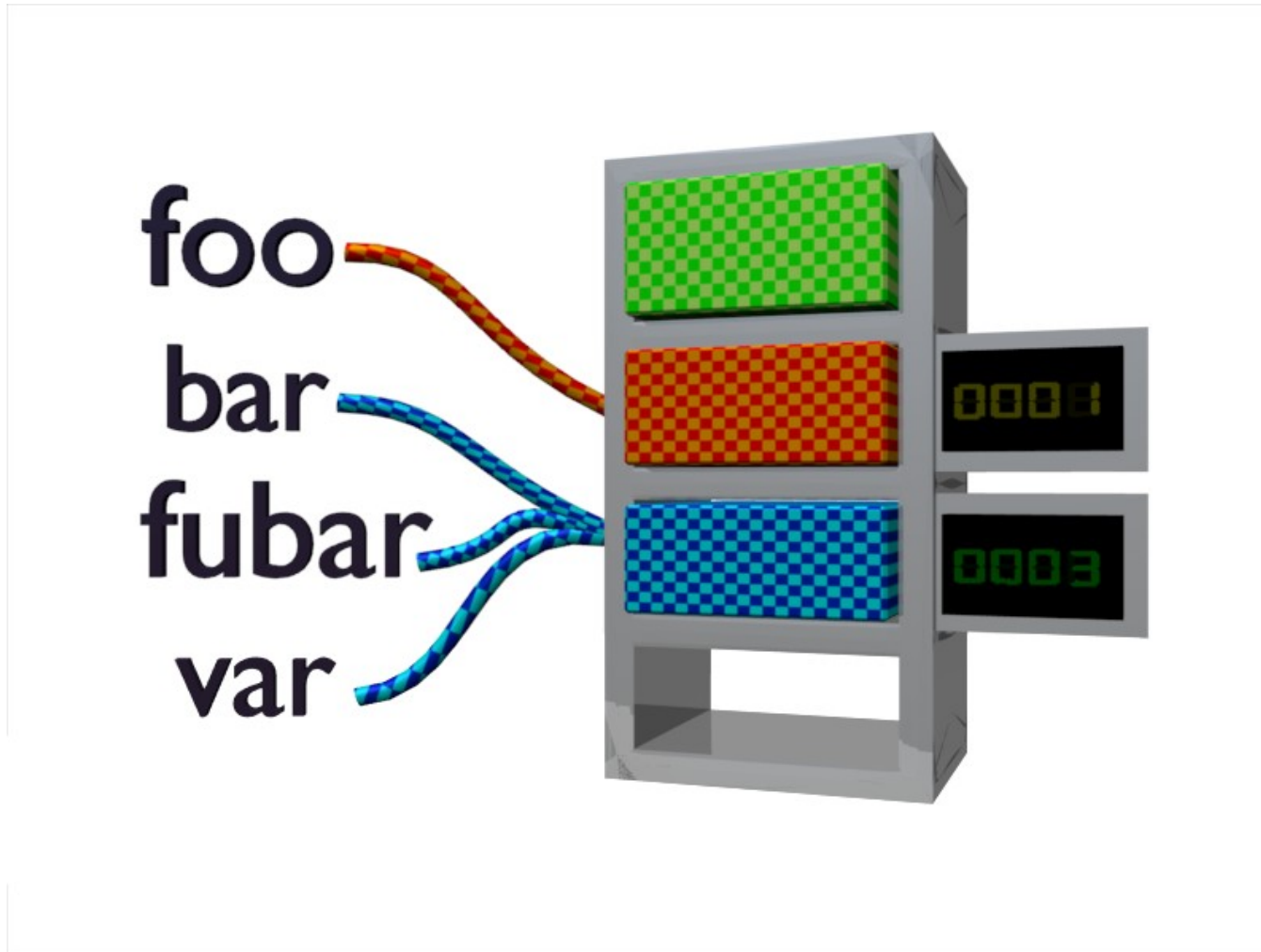
Tipagem estática



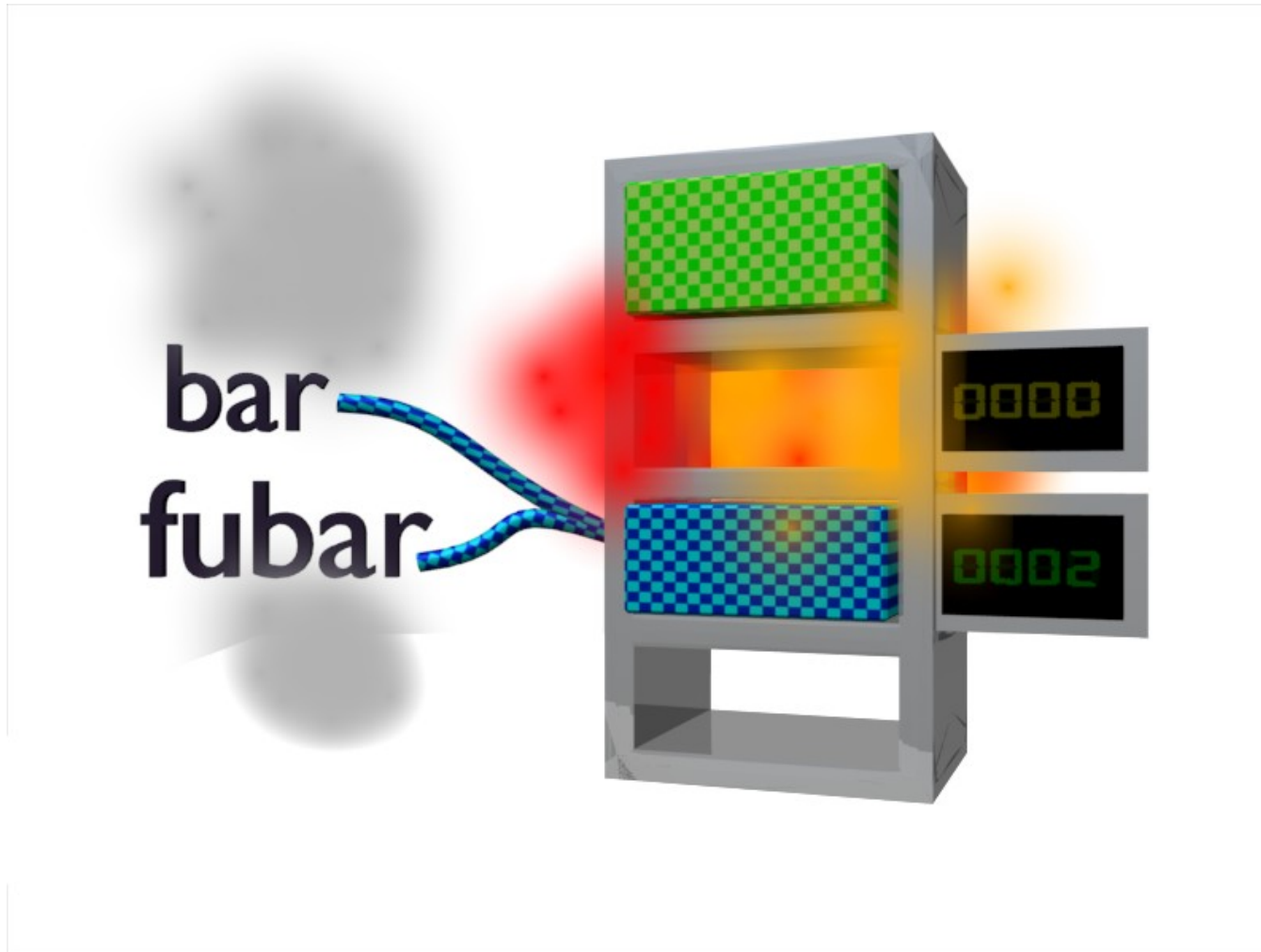
Tipagem dinâmica



Garbage collector



Garbage collector





There and back again

I agora
para algo
completamente
diferente...

Um pouco de história. . .

1990 - A invenção



Python foi inventado por **Guido van Rossum**, matemático holandês.

Guido Van Rossum

1990 - O propósito inicial



Andrew Tanenbaum

Guido estava trabalhando no desenvolvimento do sistema operacional distribuído *Amoeba* de *Andrew Tanenbaum*.

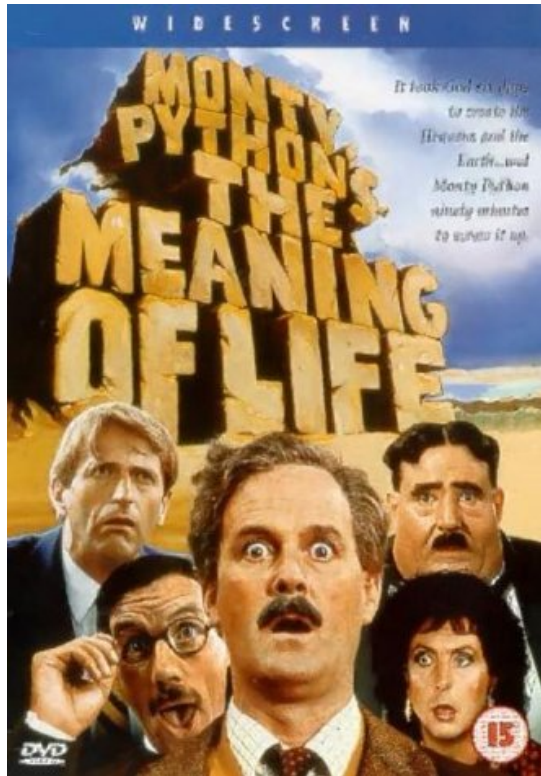
Python seria a linguagem de script desse OS.

1990 - As influências

- {C && C++}
- Pascal;
- Modula;
- (lisp)
- \$Perl

Python foi influenciado pela linguagem de ensino **ABC**, a qual Guido tinha participado do desenvolvimento; assim como muitas outras linguagens.

1990 - As más influências



Python foi assim nomeado porque Guido era fã incondicional de [Monty Python's Flying Circus](#), um seriado da TV britânica dos anos 60-70.

Monty Python e o
Sentido da Vida

1991 - O primeiro lançamento



Lata de SPAM

O primeiro lançamento ao público foi realizado em 1991.

1994 - UseNet



UseNet newsgroups

Em 1994 foi aprovada a criação do grupo [comp.lang.python](#) na UseNet(atual [GoogleGroups](#)).

Licença



OpenSource Initiative

Python foi **livre** como a liberdade de expressão desde o início. Pode ser usado tanto em software proprietário como software livre.



Porque usar python

- Sintaxe limpa
- Fácil de aprender
- Material de aprendizado na web
- Módulos e extensões
- Extensível
- Flexível
- Estruturas de dados únicas
- Multiparadigma



- Tempo de desenvolvimento(agile programming)
- Independência de plataforma
- Software livre
- Comunidade
-



Porque NÃO usar python

- Interpretação e velocidade de processamento
- Não é "Mainstream"
- Escasso material em português do Brasil



E lá vamos nós de novo. . .

Agora
para algo
completamente
diferente...

Python hoje em dia

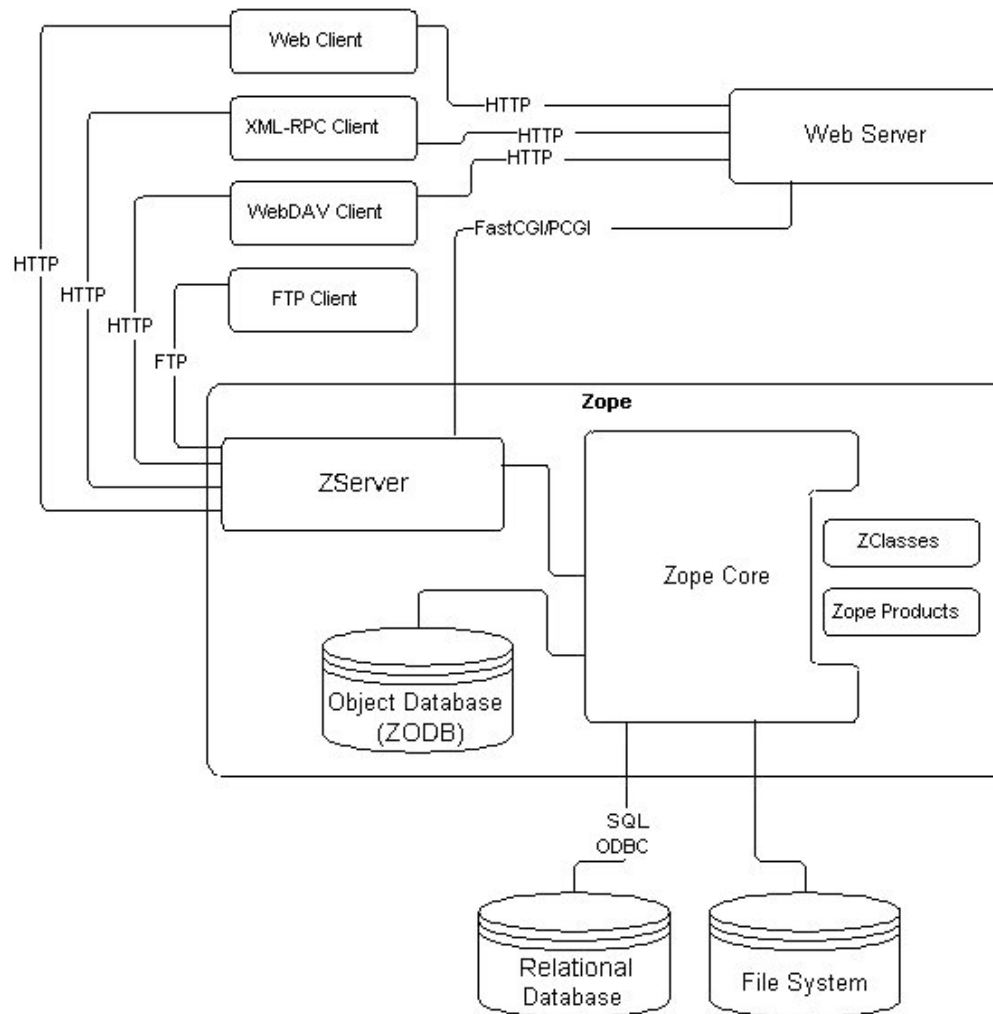
Zope

Zope é um servidor de aplicação usado para:

- Sistemas de gerenciamento de conteúdo
- Portais
- Intranets
- Aplicações

Com os produtos(módulos) corretos instalados, o zope pode rodar em um cluster(ZEO), gerenciar conteúdo dinamicamente(CMF), ou mesmo publicar o conteúdo em outras mídias(Silva), dentre muitos outros.

Zope - Arquitetura



ERP5

ERP5 é um sistema ERP/CRM/MRP/SCM/PDM de missão crítica usado tanto por organizações industriais como agências governamentais.

- Usa Zope em seu backend
- É composto de uma aplicação Zope e uma série de extensões
- Suporte à UNIX, Windows e Mac.

ERP5 screenshots

ERP5 / Transformation / 124B401 / Créé par : Martine_Cirot - 2003/04/02 13:00:41.743 GMT+2

Composants | Opérations | Metadata

Transformation: 1245401

Description: transformation complete du 124B401

Produit principal: modele/1245401

Axe d'état: transformation_state

Stade de production: fini

Transformations utilisées: 12410

Etat: valide

Axes de variantage: coloris, taille

Variantes possibles: coloris, taille/duite/38, taille/duite/40, taille/duite/42, coloris/modele/124B401/1

Affichage en lignes: coloris

Affichage en colonnes: taille

Affichage en tableaux: coloris, taille

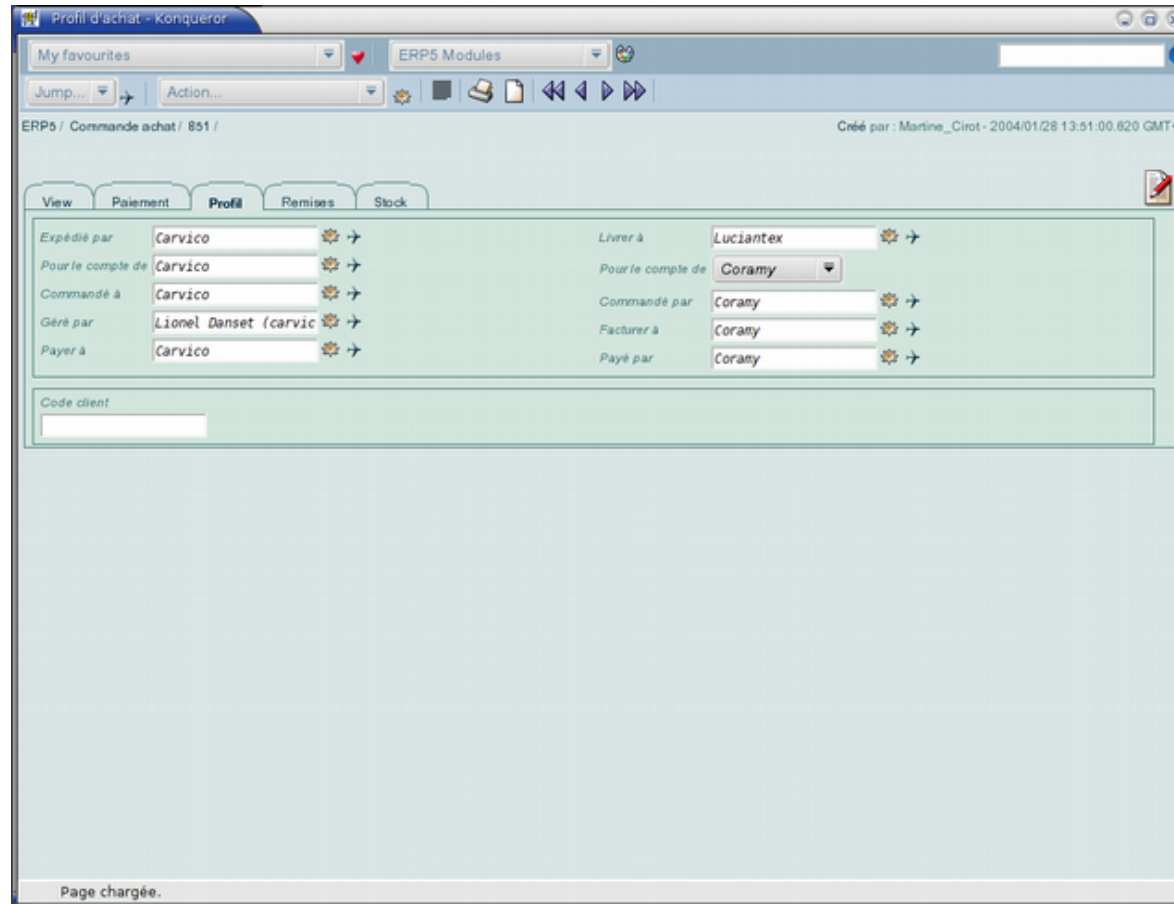
Composants: 9 Records - 0 item(s) selected

Ligne	Description	Composant	Quantité	Unité
3	laminette 6g	composant/LA001	0.009	Poids/Kilogramme
8	fil polyester	composant/FI002	1.0	Longueur/Metre
4	patte de proprete	composant/PA004	1.0	Unité/Centaine
2	doublure	tissu/DB001	1574.41	Surface/Centimetre_carre
7	fil mousse	composant/FI001	1.0	Longueur/Metre
0	tissu 1	tissu/TI382	2288.85	Surface/Centimetre_carre
1	tissu 2- bretelle - colletage 2,5x310 2aig +6g double rempli, petit écartement(P18)-brides	tissu/TI382	800.4	Surface/Centimetre_carre

http://erp5.coramy.fr/coramy/transformation/124B401/1/view?selection_index=6&selection_name=transformed_resource_selection

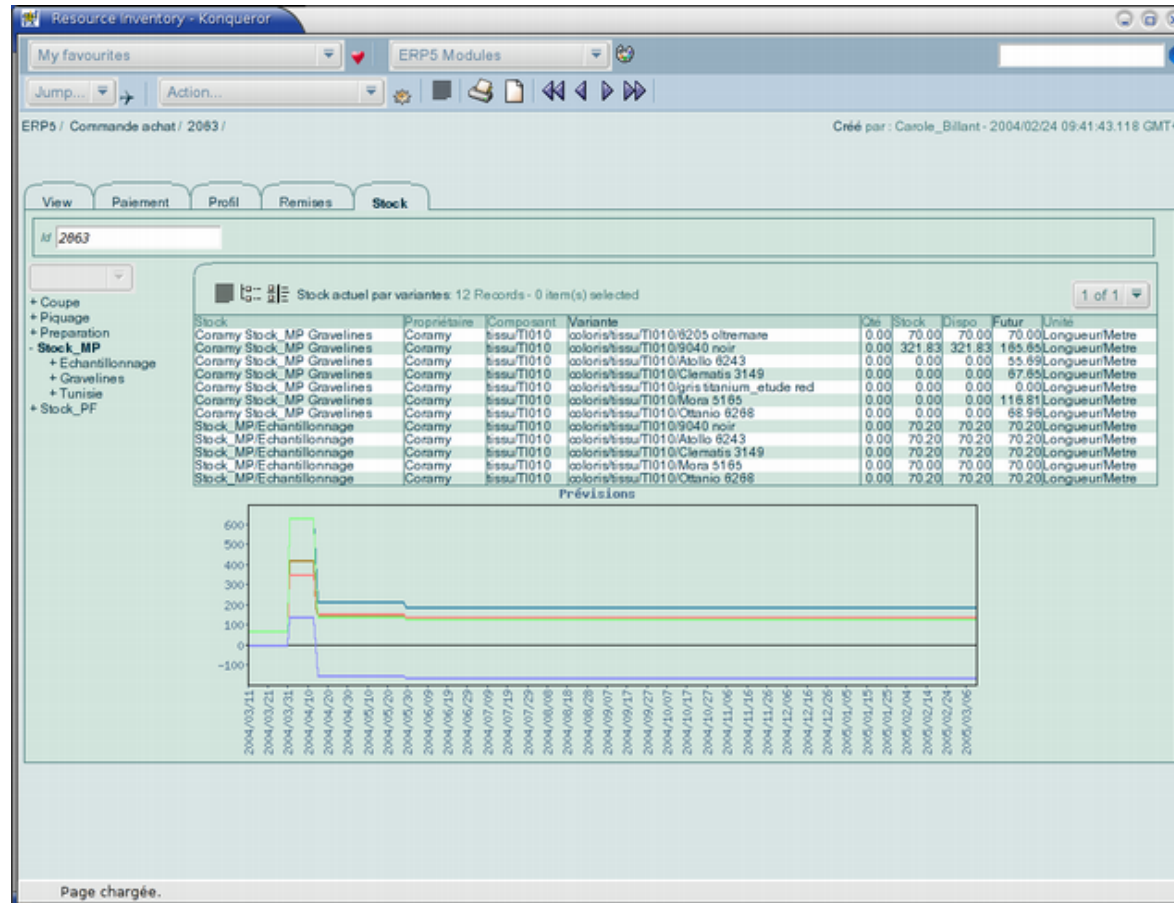
Registrando uma transação

ERP5 screenshots



Visualizando o perfil de um produto

ERP5 screenshots



Visualizando um estoque

ERP5 screenshots

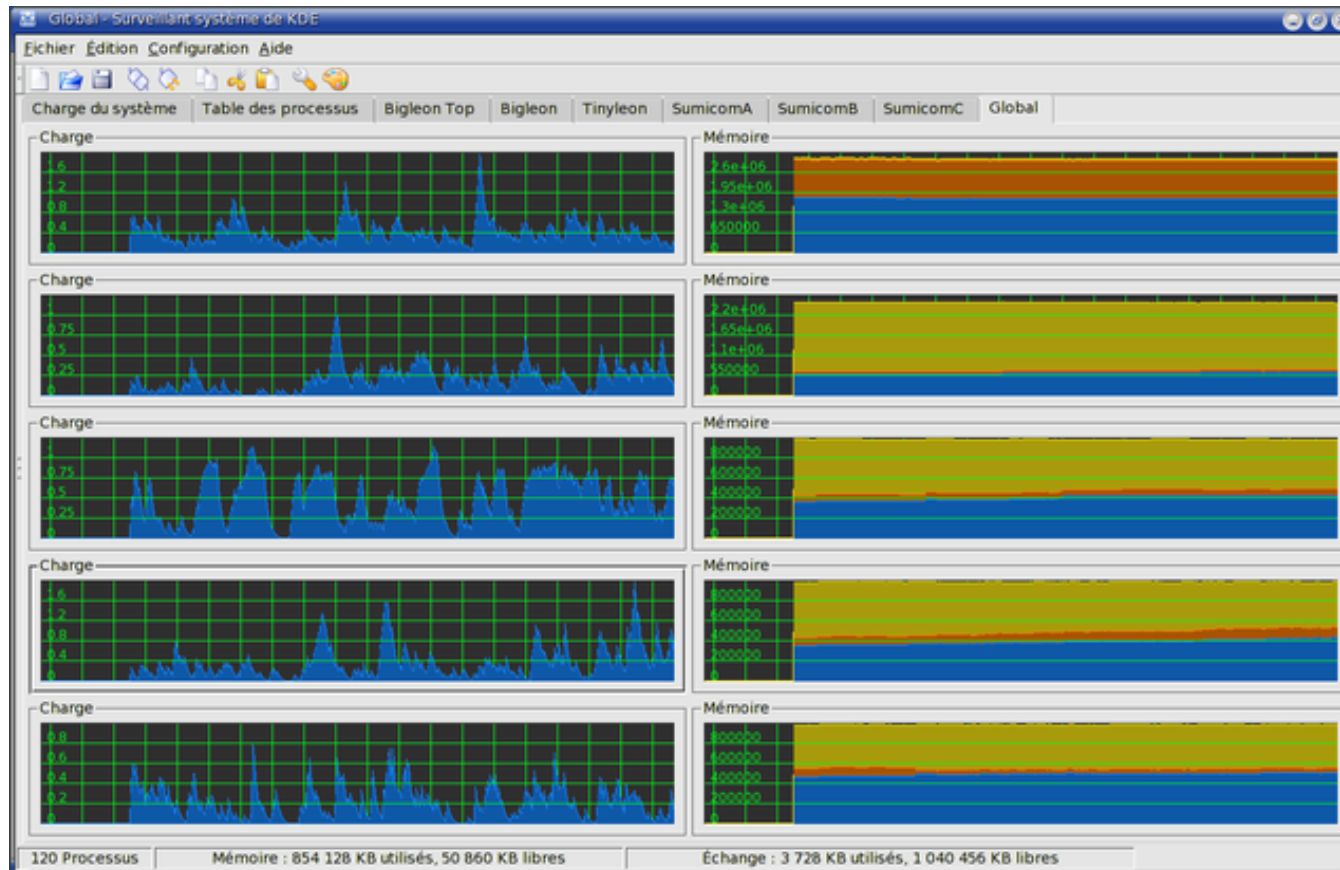


Gráfico de carga de um cluster ERP5

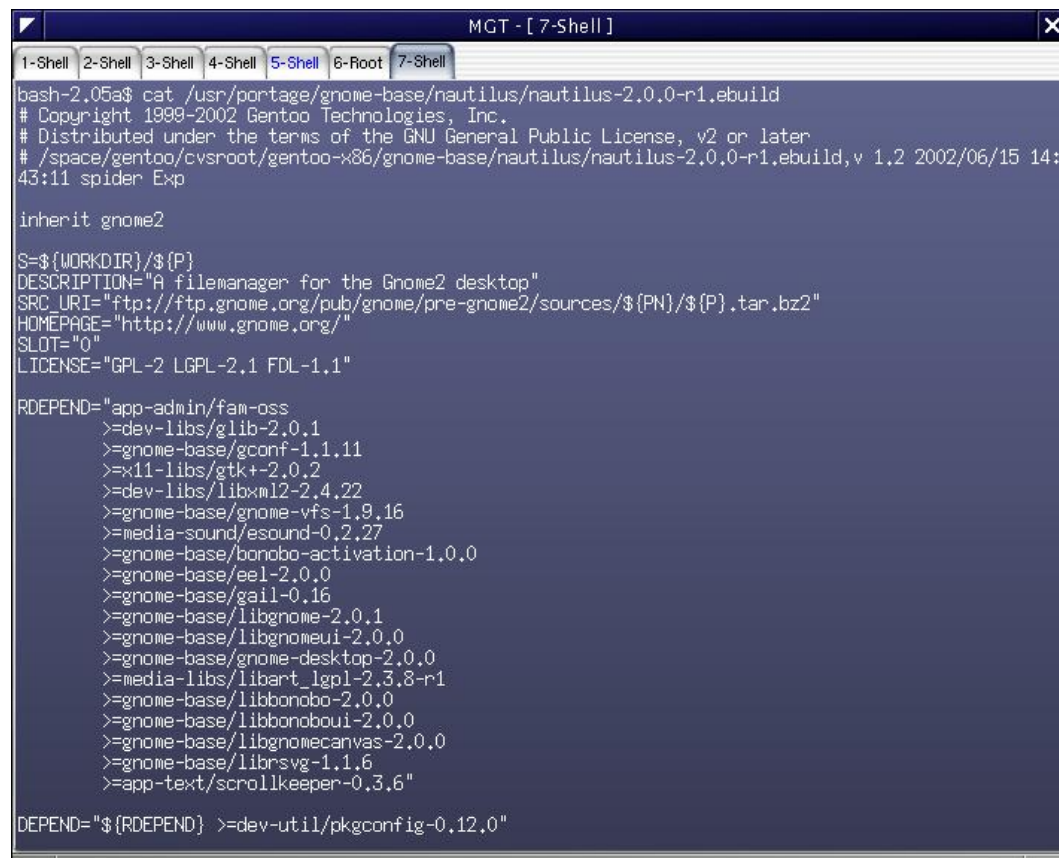


Gentoo Linux

Gentoo Linux é uma metadistribuição GNU/Linux projetada para ser modular; portátil e otimizada na máquina do usuário.

- Sistema de gerenciamento de pacotes(Portage) implementado em python.
- Instalação(praticamente) from scratch ou pré-compilada.

Portage - Screenshots



```
MGT - [ 7-Shell ]
1-Shell 2-Shell 3-Shell 4-Shell 5-Shell 6-Root 7-Shell
bash-2.05a$ cat /usr/portage/gnome-base/nautilus/nautilus-2.0.0-r1.ebuild
# Copyright 1999-2002 Gentoo Technologies, Inc.
# Distributed under the terms of the GNU General Public License, v2 or later
# /space/gentoo/cvsroot/gentoo-x86/gnome-base/nautilus/nautilus-2.0.0-r1.ebuild,v 1.2 2002/06/15 14:
43:11 spider Exp

inherit gnome2

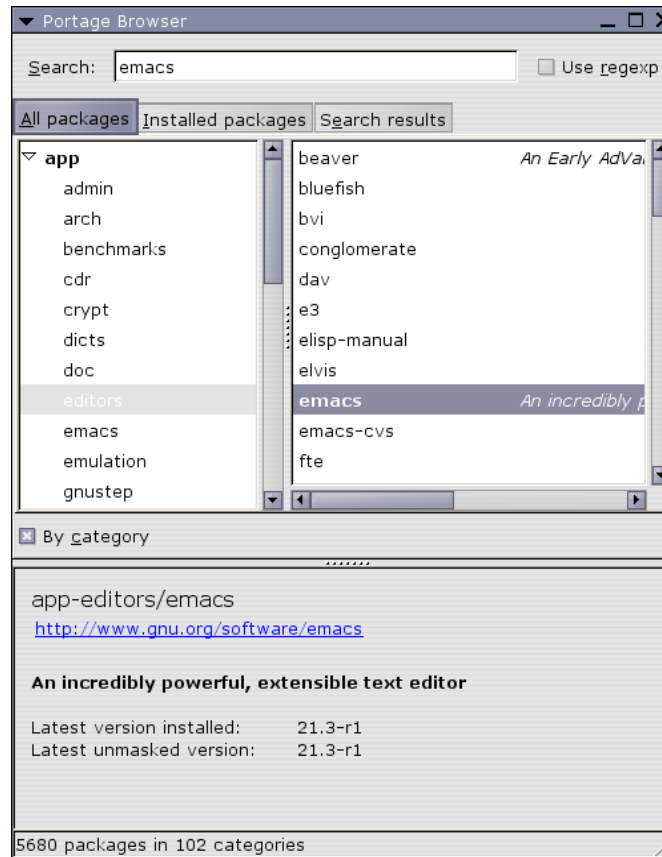
S=${WORKDIR}/${P}
DESCRIPTION="A filemanager for the Gnome2 desktop"
SRC_URI="ftp://ftp.gnome.org/pub/gnome/pre-gnome2/sources/${PN}/${P}.tar.bz2"
HOMEPAGE="http://www.gnome.org/"
SLOT="0"
LICENSE="GPL-2 LGPL-2.1 FDL-1.1"

RDEPEND="app-admin/fam-oss
        >=dev-libs/glib-2.0.1
        >=gnome-base/gconf-1.1.11
        >=x11-libs/gtk+-2.0.2
        >=dev-libs/libxml2-2.4.22
        >=gnome-base/gnome-vfs-1.9.16
        >=media-sound/esound-0.2.27
        >=gnome-base/bonobo-activation-1.0.0
        >=gnome-base/eel-2.0.0
        >=gnome-base/gail-0.16
        >=gnome-base/libgnome-2.0.1
        >=gnome-base/libgnomeui-2.0.0
        >=gnome-base/gnome-desktop-2.0.0
        >=media-libs/libart_lgpl-2.3.8-r1
        >=gnome-base/libbonobo-2.0.0
        >=gnome-base/libbonoboui-2.0.0
        >=gnome-base/libgnomecanvas-2.0.0
        >=gnome-base/librsvg-1.1.6
        >=app-text/scrollkeeper-0.3.6"

DEPEND="${RDEPEND} >=dev-util/pkgconfig-0.12.0"
```

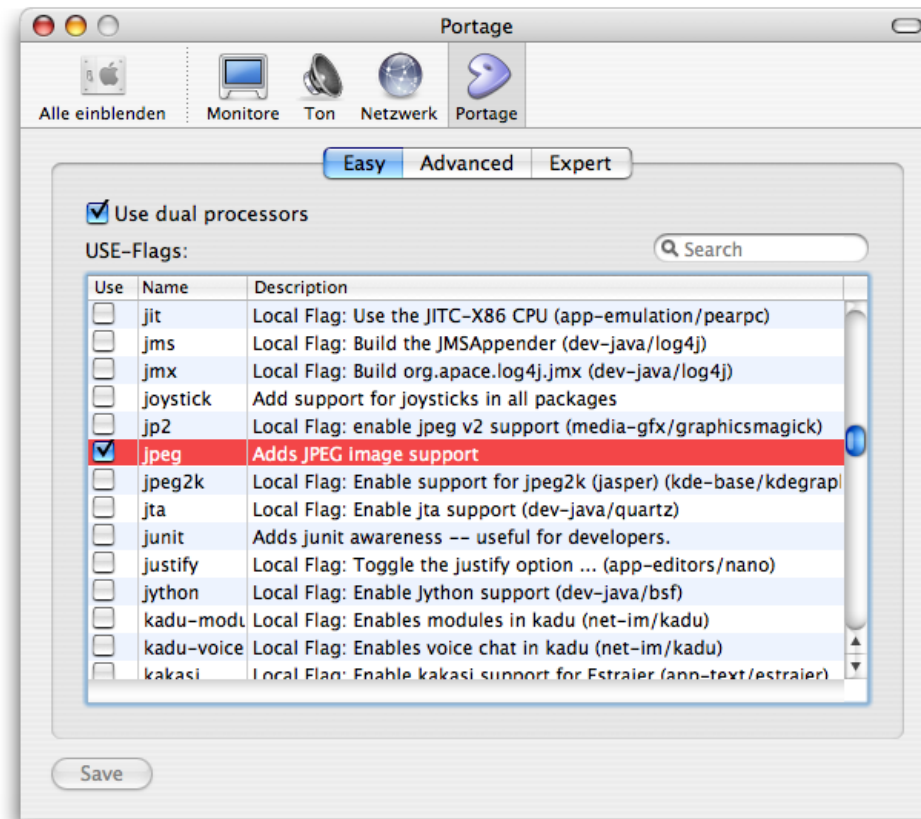
Portage em um sistema Gentoo

Portage - Screenshots



GPortage - Interface gráfica com PyGTK

Portage - Screenshots



Portage em um Mac OS X



Blender

Uma aplicação 3d que oferece recursos disponíveis somente em softwares comerciais.

- Interface em OpenGL
- Interpretador python built-in
- API Python

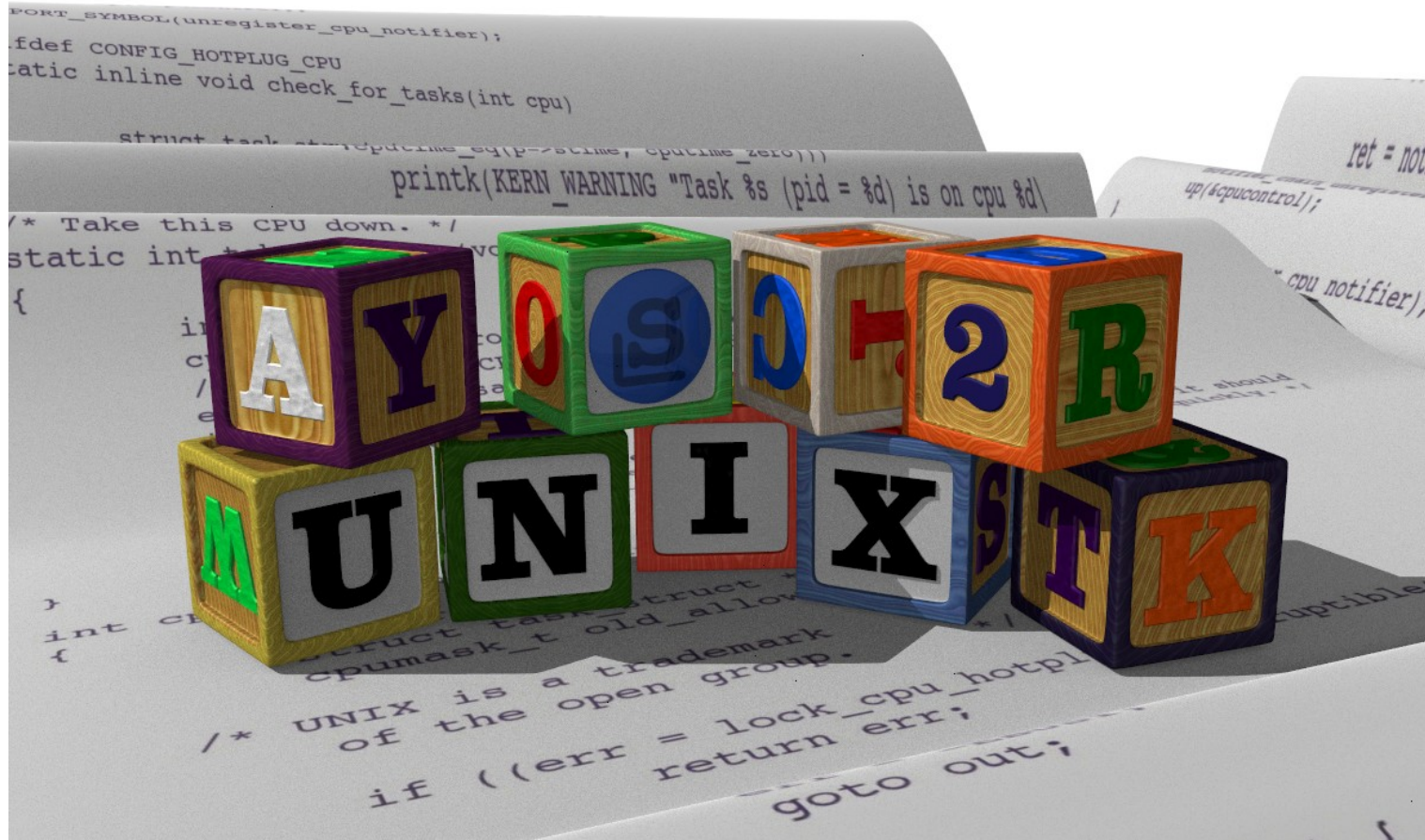
Multiplataforma, usado por profissionais de todo o mundo e ganhador de vários prêmios.

Blender - images



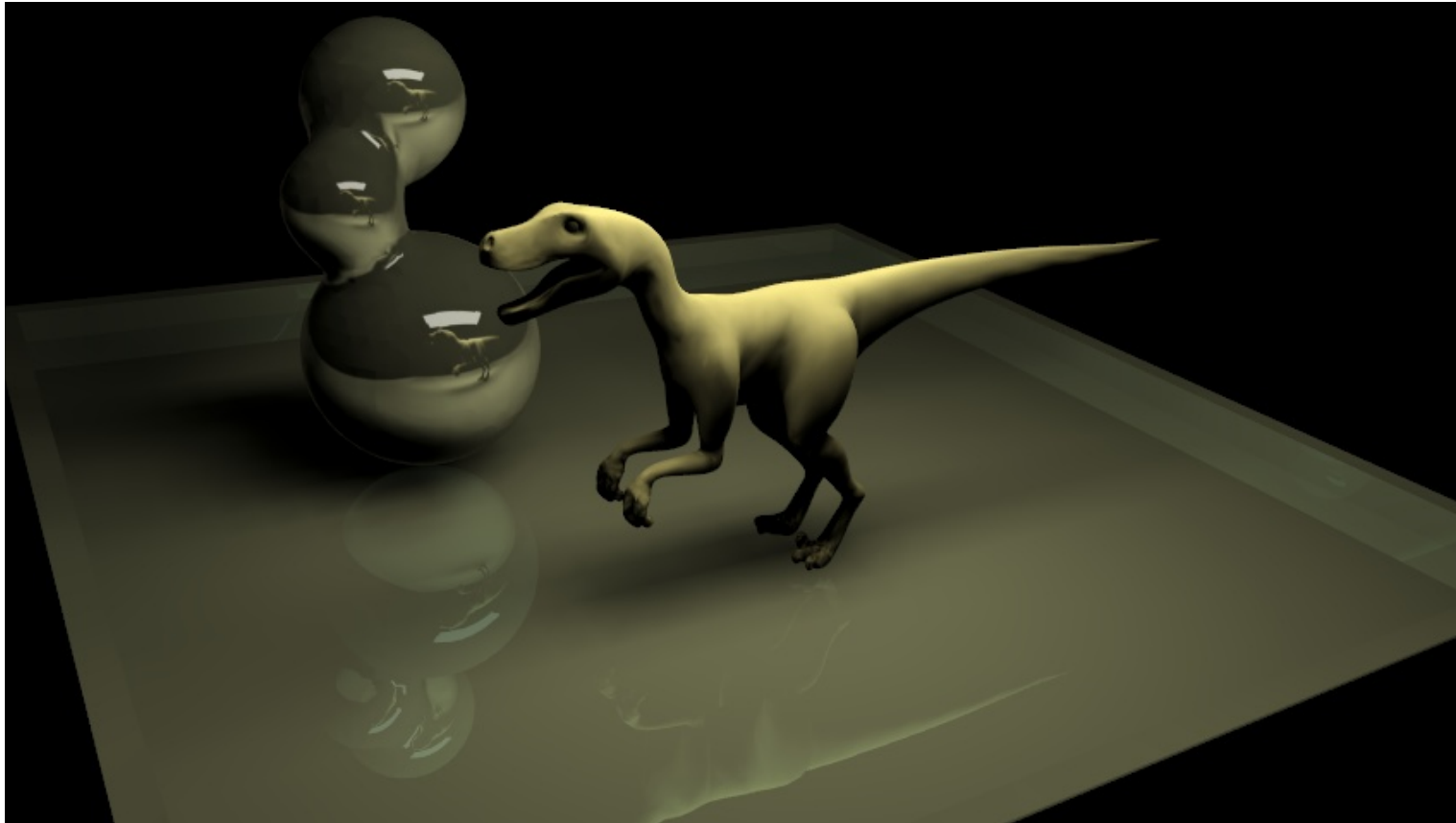
teapot - um exemplo

Blender - images



unixblocks - ©Carlos M. dos Santos

Blender - images



Raptor - um exemplo de radiosity

Blender - images



Snog the frog - ©Andy Goralczyk



Blender

Hey, ainda não acabou!

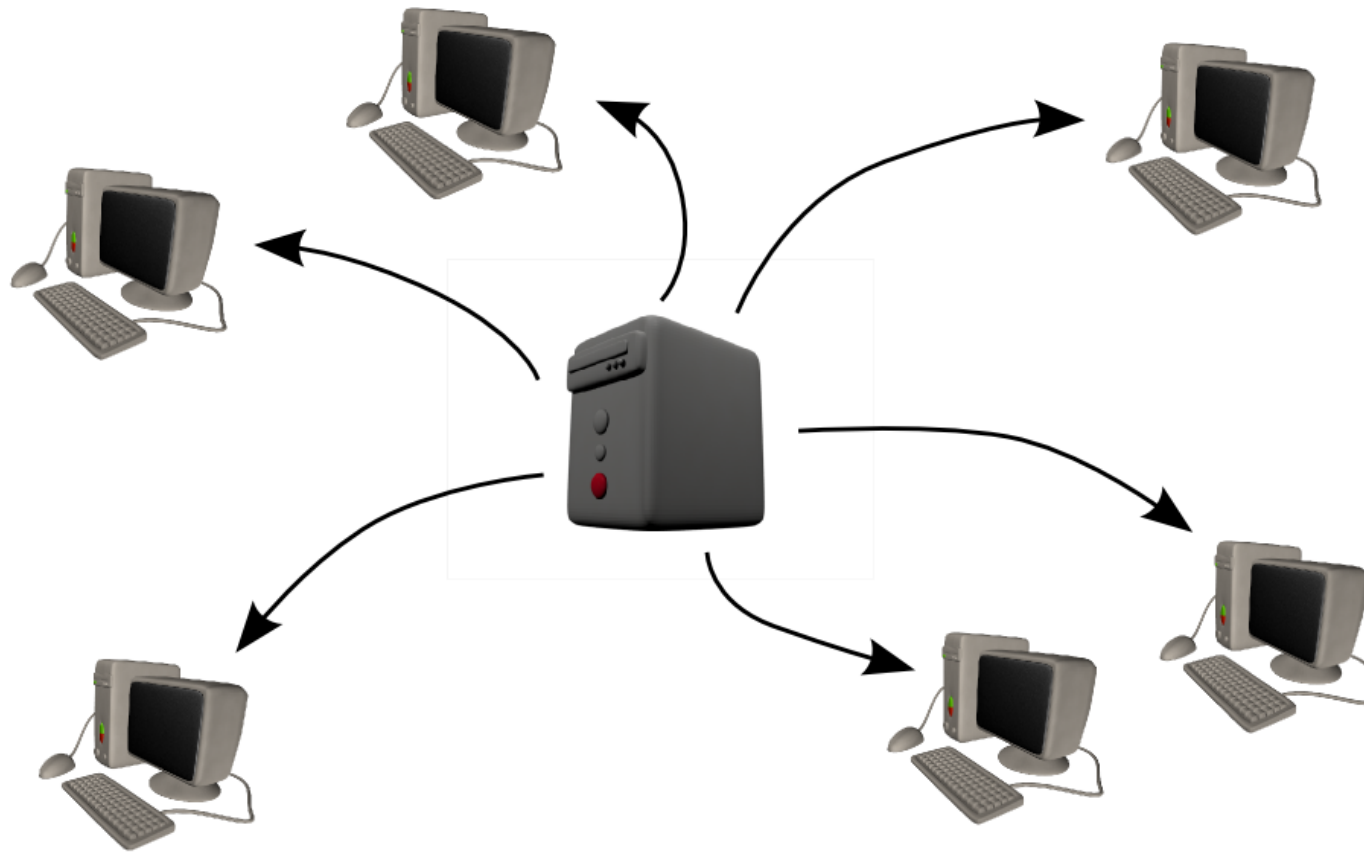
BitTorrent

“BitTorrent is a free speech tool.”

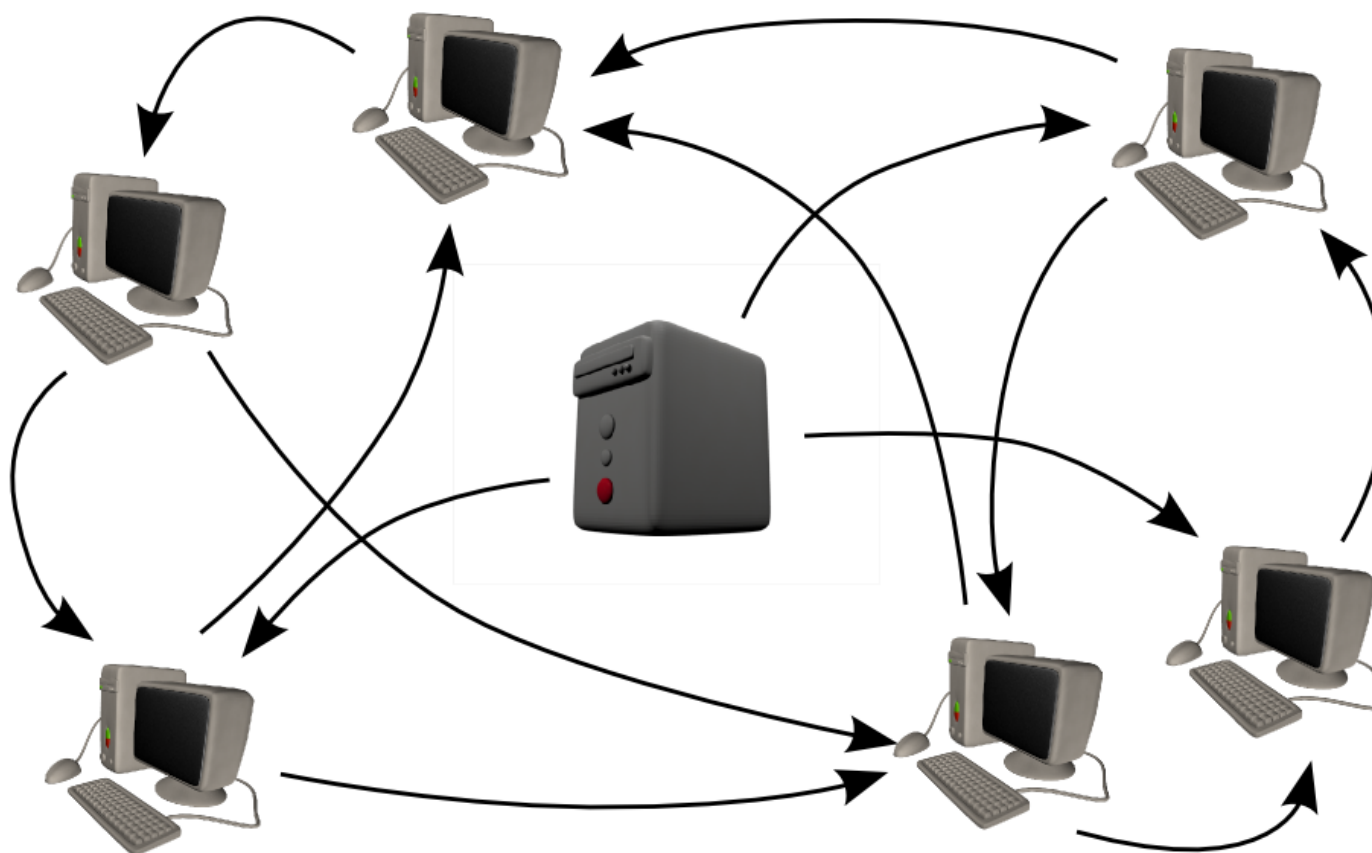
BitTorrent é usado para publicar arquivos em rede; porém:

- O que você quer publicar não está em um único lugar
- Todos que querem o arquivo cooperam para sua circulação

O método clássico



O método BitTorrent



Series60

Series60 é uma plataforma smartphone desenvolvida pela [Nokia](#).

- Usa o Symbian OS como plataforma
- Possui um port funcional do interpretador python
- Possui API's Java e C++

Series60 screenshots

O prompt interativo.



```
>>> from BeautifulSoup
import *
>>> f=open('e:\\hack\\
random.html')
>>> soup=BeautifulSoup(f.
read())
>>>
Options          Exit
```



```
random.html')
>>> soup=BeautifulSoup(f.
read())
>>> print soup.first('a')
<a href='http://posnteo.
com'>link inside</a>
>>>
Options          Exit
```

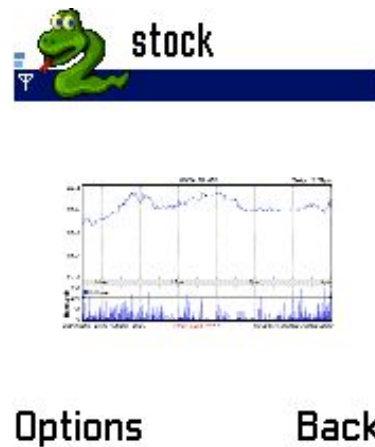
Series60 screenshots

Um dicionário.



Series60 screenshots

Uma aplicação que acessa a bolsa de valores de NY.





Pygame

Pygame é um conjunto de wrappers para a [Simple DirectMedia Layer Library \(SDL\)](#).

- Projetado especialmente para aplicações multimídia
- Altamente portátil
- Utilizado por vários desenvolvedores

Pygame screenshots



Temple of elemental evil(©2003 Troika/Bioware/Atari)

Pygame screenshots



Temple of elemental evil(©2003 Troika/Bioware/Atari)

Pygame screenshots



Temple of elemental evil(©2003 Troika/Bioware/Atari)

Pygame screenshots



Temple of elemental evil(©2003 Troika/Bioware/Atari)

Quem usa?



Links

- Python - <http://www.python.org>
- Zope - <http://www.zope.org>
- ERP5 - <http://www.erp5.org>
- Gentoo Linux - <http://www.gentoo.org>
- Blender - <http://www.blender3d.org>
- BitTorrent - <http://www.bittorrent.org>
- Nokia Series60 - <http://www.series60.com>
- Pygame - <http://www.pygame.org>



- Comunidade Brasileira - <http://pythonbrasil.com.br>
- Dive into python - <http://diveintopython.org>
- Python cookbook
<http://aspn.activestate.com/ASPN/Python/Cookbook>



Copyright

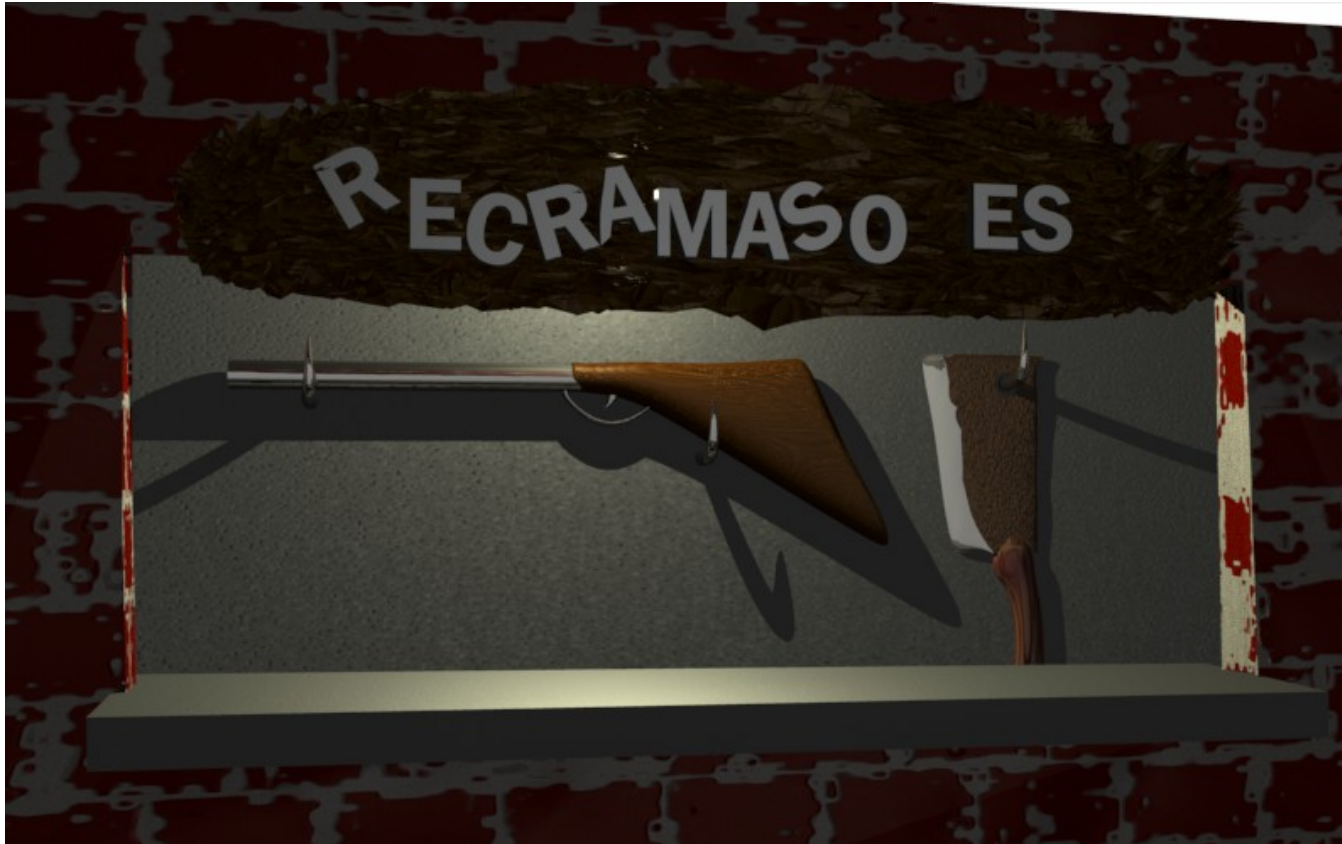
Todas as marcas, registradas ou não, citadas aqui são propriedade de seus respectivos detentores; usadas com propósito acadêmico.

E agora para algo completamente diferente(o filme) é uma marca registrada de Python(Monty) Pictures inc.

Thanks

The producers would like to thank all the fish who have taken part in this film. We hope that other fish will follow the example of those who have participated, so that, in future, fish all over the world will live together in harmony and understanding, and put aside their petty differences, cease pursuing and eating each other and live for a brighter, better future for all fish, and those who love them.

Contate-nos





Contate-nos

- Carlos
knightslayer@gmail.com
<http://knightstalker.freezope.org>
- Eduardo
eduardo.padoan@gmail.com
<http://edcrypt.cjb.net>