

Computação 1 - Python

Aula 1 - Teórica: Introdução

Conhecendo a turma

Experiência com programação e uso do computador

- Quantos já programaram antes ?
- Quais linguagens ?
- Quantos tem computador em casa com acesso a Internet ?
- Qual Sistema Operacional ?
- Quantos são calouros ?
- Quem veio de outro curso ?
- Nível de inglês ?

Objetivo da disciplina

Aprender a construir programas de computador.

Exemplos de Programas ?

Objetivo da disciplina

Aprender a construir programas de computador.

Exemplos de Programas ?

- Explorer, Firefox, Google Chrome
- Facebook
- Windows
- Word, Powerpoint
- Media Player, iTunes
- SIGA
- The Sims 4
- ...

Objetivo da disciplina

Aprender a construir programas de computador.

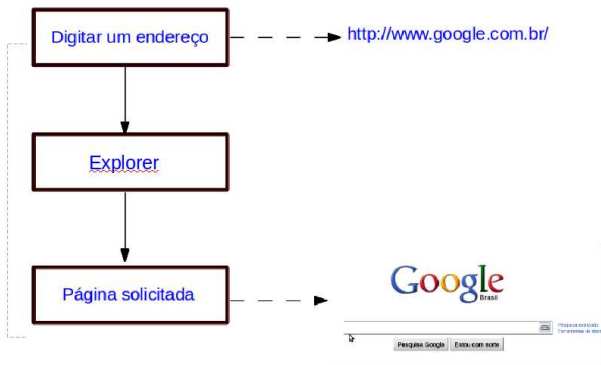
Exemplos de Programas ?

- Explorer, Firefox, Google Chrome

Qual a “tarefa” que o Explorer deve realizar?

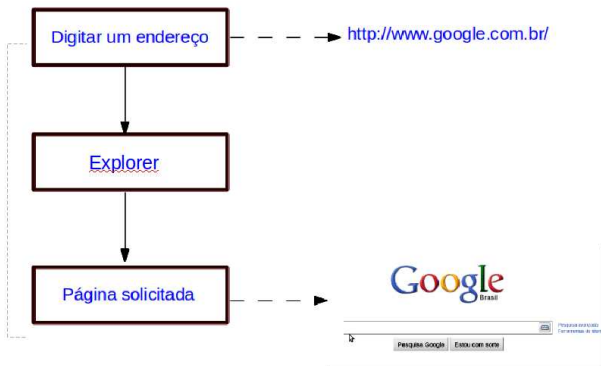
Objetivo da disciplina

Qual a “tarefa” que o Explorer deve realizar?



Objetivo da disciplina

Como o Explorer realiza esta “tarefa”?



Algoritmo

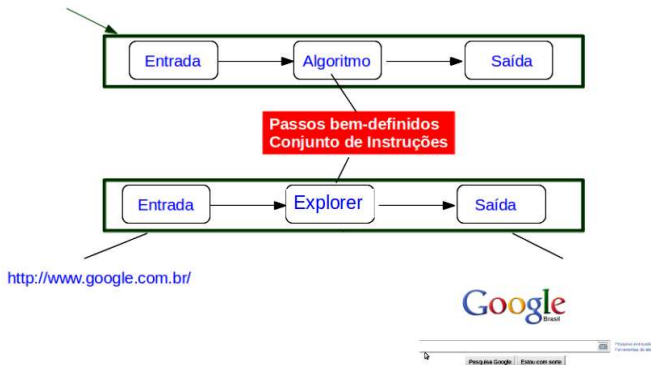
Método efetivo expresso como um conjunto de instruções que devem ser feitas para realizar uma tarefa.

Características

- **Finitude:** deve sempre terminar após um número finito de passos.
- **Bem-definido:** cada passo de um algoritmo deve ser precisamente definido (sem ambiguidades).
- **Entradas:** deve ter zero ou mais entradas (informações que são fornecidas antes do algoritmo iniciar).
- **Saídas:** deve ter uma ou mais saídas (resultado final do algoritmo).
- **Efetividade:** todas as operações devem ser suficientemente básicas de modo que possam ser executadas com precisão em um tempo finito por uma pessoa.

Algoritmo - Características

Finitude - (para um problema genérico)



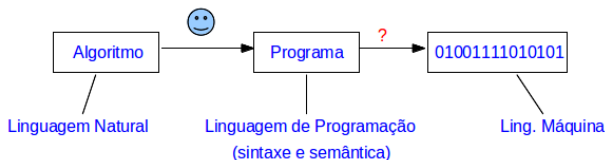
Finitude - (para um problema específico): Dado um endereço na internet, exibir o conteúdo endereçado.

Programa de Computador

Conjunto de instruções que descrevem como uma tarefa deve ser realizada por um computador. Ou seja, o computador deve ser capaz de “entender” as instruções.

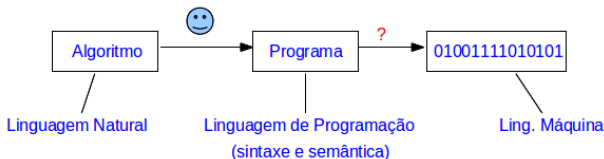


O computador “entende” linguagem de máquina: 01011100110.
Como traduzir um algoritmo para código de máquina?



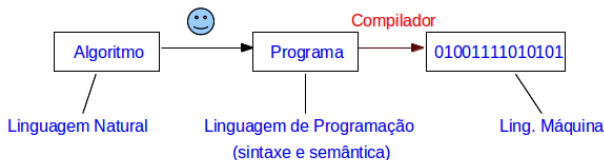
Linguagens e Paradíguas de Programação

- 1 **Programação Imperativa:** define sequências de comandos que um computador deve seguir para realizar uma tarefa.
- 2 **Programação Declarativa:** expressa o que deve ser realizado sem dizer como realizar.
- 3 **Programação Orientada a Objeto**
- 4 **Programação Funcional**

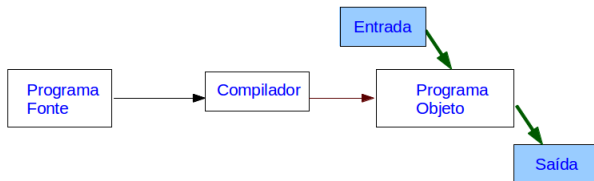


Linguagens e Paradígmats de Programação

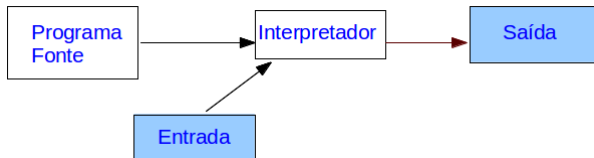
- 1 **Programação Imperativa:** define sequências de comandos que um computador deve seguir para realizar uma tarefa.
- 2 **Programação Declarativa:** expressa o que deve ser realizado sem dizer como realizar.
- 3 **Programação Orientada a Objeto**
- 4 **Programação Funcional**



Compilador



Interpretador



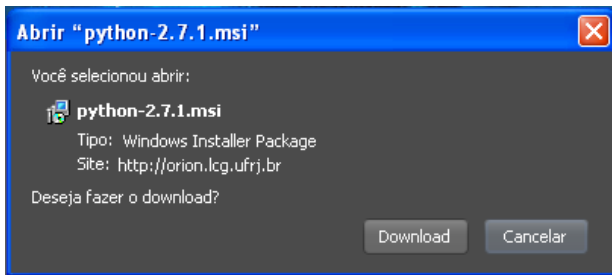
Por que Python ?

- Simples o suficiente para um curso introdutório
- Muitos recursos
 - Orientação a Objetos
 - Escalável (módulos, classes, controle de exceções)
 - Biblioteca embutida extensa e grande número de módulos fornecidos por terceiros
- Grande variedade de aplicações
- Linguagem interpretada (script)
- Multi-plataforma
- Grátis!
- Comunidade bastante grande

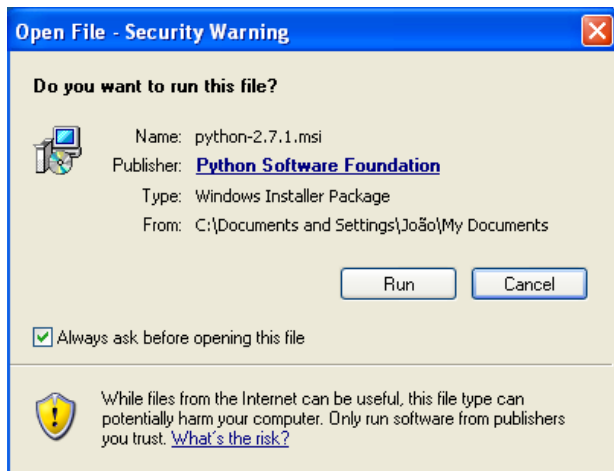
Quem usa Python ?



Instalando o interpretador Python 2.7



Instalando o interpretador Python 2.7



Instalando o interpretador Python 2.7



Instalando o interpretador Python 2.7



Instalando o interpretador Python 2.7



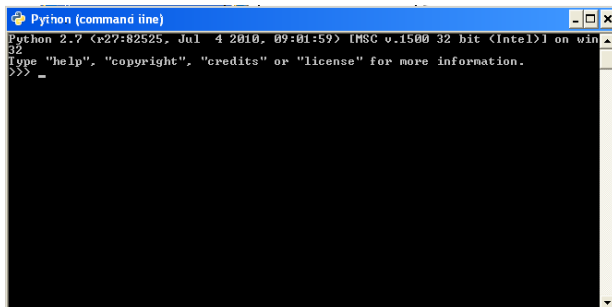
Instalando o interpretador Python 2.7



Instalando o interpretador Python 2.7



Instalando o interpretador Python 2.7

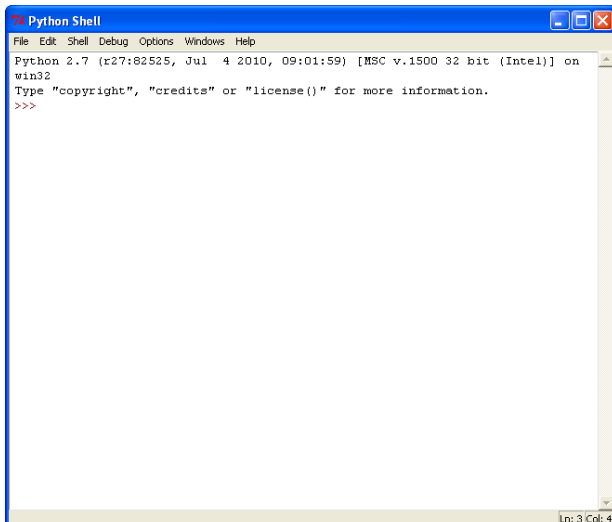


```
Python (command line)
Python 2.7 (r27:82525, Jul 4 2010, 09:01:59) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```


Instalando o interpretador Python 2.7



Instalando o interpretador Python 2.7

A screenshot of a Windows application window titled "Python Shell". The window has a blue title bar with standard minimize, maximize, and close buttons. Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Windows, and Help. The main area of the window is a text editor with a white background and a vertical scrollbar on the right. The text displayed in the editor is: "Python 2.7 (r27:82525, Jul 4 2010, 09:01:59) [MSC v.1500 32 bit (Intel)] on win32", followed by "Type 'copyright', 'credits' or 'license()' for more information.", and then a red prompt ">>>". At the bottom of the window, there is a status bar that reads "Ln: 3 Col: 4". To the right of the status bar, there are several small icons for navigating through the document, including arrows and a magnifying glass.

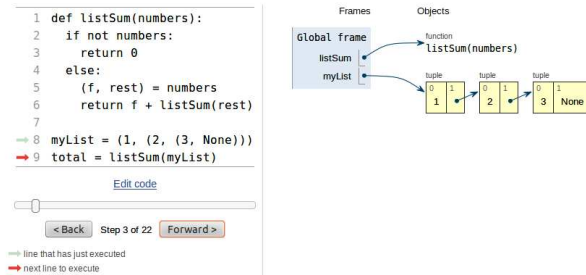
```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.7 (r27:82525, Jul 4 2010, 09:01:59) [MSC v.1500 32 bit (Intel)] on
win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

Python na Nuvem

Pythontutor - <http://pythontutor.com/>

[Start using Online Python Tutor now](#)

For instance, here is a visualization showing a program that [recursively](#) finds the sum of a ([cons-style](#)) linked list. Click the “Forward” button to see what happens as the computer executes each line of code.



Computação 1 - Python

Aula 1 - Teórica: Introdução