

FUNDAMENTOS DE PROGRAMAÇÃO

Introdução ao Python
Prof. Bruno Góis Mateus



Índice

- Introdução
- Python
- Python em ação
- O nosso primeiro programa
- Variáveis, expressões e instruções

INTRODUÇÃO

Relembrando

- Resolver problemas é um ponto central da ciência da computação
 - As soluções criadas através do processo de resolução são também importantes
 - Essas soluções são chamadas de **Algoritmos**

Algoritmos

- Conjunto de passos que devem ser seguidos para resolver um problema
- Nosso objetivo é pegar um problema e desenvolver um **algoritmo**
 - Solução geral
 - Pode ser automatizada pelo computador

Programação

- É a habilidade de transformar um algoritmo e representá-lo em uma notação (programa) que o computador pode executar
- Esses programas são escritos em uma **Linguagem de programação**

PYTHON

Python

- É uma linguagem de programação de alto nível
 - Outras linguagens de programação de alto nível: C, Java, PHP ...
- Foi lançada por Guido van Rossum em 1991.
 - Projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional
 - O nome Python teve a sua origem no grupo humorístico britânico Monty Python



Python

- **Python** é a linguagem do Interpretador **Python** e aqueles que conseguem se comunicar com ele. Um indivíduo que consegue “falar” **Python** é conhecido como um **Pythonista**. É uma habilidade bem incomum, e pode ser hereditária. Praticamente todos os **Pythonistas** usam o software inicialmente desenvolvido por **Guido van Rossum**

Linguagens de baixo nível

- Código de máquina ou assembler
 - Código máquina
 - Basicamente a codificação binária de cada instrução
 - Assembler
 - Um formato um pouco mais fácil para se referir a instruções de baixo nível

Linguagens de baixo nível

Um computador só é capaz de executar programas escritos em linguagem de máquina

Linguagens de alto nível

- Programas escritos em linguagem de alto nível devem ser processados antes de ser executados
- Desvantagem
 - Processamento extra
- Vantagens
 - Facilidade para escrever programas
 - Menos demorado
 - Mais curto
 - Mais legível
 - Portável

Linguagens de programação

Linguagens de alto nível

- Usada para construir a grande maioria dos programas existentes

Linguagens de baixo nível

- Usada para construir algumas programas específicos

Dois tipos de programas transformam linguagens de alto nível em linguagem de baixo nível

Compiladores

Interpretadores

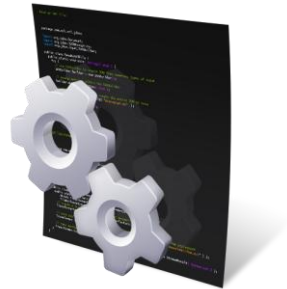
Compilador

- Lê o programa e o **traduz** completamente antes de executá-lo
 - Programa em alto nível -> código fonte
 - Programa traduzido -> código objeto ou executável
- Um programa traduzido pode ser executado diversas vezes



Interpretador

- Lê o programa em alto nível e o executa
 - Realiza o que o programa especifica
- Processa o programa pedaço por pedaço
 - Lê uma linha
 - Realiza algumas computações
- O programa é interpretado cada vez que for executado



Interpretador



Híbrida

- Várias linguagens modernas utilizam os dois processos anteriores
 - Primeiramente o código é compilado para **bytecode**
 - Depois é ele interpretado pela **máquina virtual**



Programas

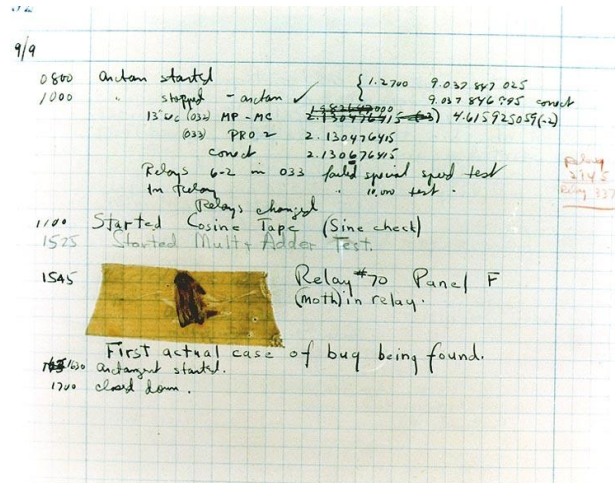
- Sequência de instruções
 - Especificam como a computação devem ser realizadas
 - Ex: Renderizar uma página html, procurar e substituir um texto em um documento de texto
- Os detalhes são diferentes entre linguagem, no entanto possuem bastante coisas em comum

Programas

- Entrada
 - Dados vindos do teclado, arquivo ou outro dispositivo
- Saída
 - Mostra os dados em uma tela, envia os dados para um arquivo
- Matemática e lógica
 - Realizar operações básica
 - Adição, multiplicação
- Condicional
 - Verifica certas condições para executa um trecho de código
- Repetição
 - Realiza operações sepetidamente

Debugging

- Programar é um processo complexo
 - É normal que erros ocorram
- Erros de programação são chamados de bug
- Existem três tipos de erros
 - Erros de sintaxe
 - Erros de execução
 - Erros de semântica
- O processo de rastrear e corrigir os erros é chamado de **debugging**



Erros de sintaxe

- Em nosso contexto sintaxe está ligado a estrutura do programa e regras sobre estrutura
- Python só pode executar programas sintaticamente corretos
- Durante as primeira semanas você irão perder bastante tempo com isso

Erros de execução

- Esses erros só aparecem quando o programa é executado
- São chamados de exceções (**exceptions**)
 - Geralmente indicam que alguma coisa ruim aconteceu
 - São raros em programas simples

Erros semânticos

- Programas são executados com sucesso mesmo que contenham erros semânticos
 - Programas fazem coisa que você não queria que fizesse
 - Você escreveu um programa diferente do que você precisava
 - O sentido do programa está errado
- Identificar esse tipo de erro pode ser trabalhoso
 - Você precisa parar e olhar todo o processo de execução

Debug

- É uma habilidade essencial para qualquer programador
 - Pode ser frustrante
 - Desafiador
 - Interessante
 - Intellectualmente rico

Debug

- É como um trabalho de detetive
- Ciência experimental
 - Hipótese do que está errado
 - Altera-se o programa
 - Tenta novamente



PYTHON EM AÇÃO

Python em ação

- Nós precisamos aprender a linguagem Python para que possamos comunicar nossas instruções para o Python
 - No começo nós erraremos muito e rabiscaremos como crianças pequenas.

Python em ação

- Quando você erra, o computador não quer saber se você é “sensível”.
 - Ele diz “**syntax error**” (erro de sintaxe) - dizendo que ele *sabe* a linguagem e você está só aprendendo.
 - Parece que Python é cruel e insensível.



Python em ação

- Nós somos inteligentes e podemos aprender
- O computador é simples e muito rápido
 - Mas não aprende
- Então é mais fácil você aprender Python que o computador aprender Português...



Python em ação

- Existem duas maneira de utilizar o interpretador Python
 - Modo Shell
 - Instruções Python são digitadas e interpretadas imediatamente exibindo em seguida o resultado
 - Modo programa
 - As instruções são escritas em um arquivo, **código fonte**
 - Interpretador executa os conteúdo contidos no arquivo
 - Por convenção os arquivos python utilizam a extensão **.py**

Falando com o python

```
$ python
```

```
Python 2.5 (r25:51918, Sep 19 2006, 08:49:13)
```

```
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>>
```



E agora?

Falando com o python

```
$ python
```

```
Python 2.5 (r25:51918, Sep 19 2006, 08:49:13)
```

```
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> x = 1
```

```
>>> print x
```

```
1
```

```
>>> x = x + 1
```

```
>>> print x
```

```
2
```

```
>>> exit()
```

Falando com o python

- Vocabulário / Palavras - Variáveis e palavras reservadas
- Estrutura de frases - padrões de sintaxe válidas
- Estrutura de histórias - construindo um programa para um propósito.

A seguir

- Introdução ao Variáveis, Sentenças e Expressões