

Introdução à Programação Python

github.com/kaylani2/iHUB_Python

Kaylani Bochie

iHUB

Conteúdo

- 1 Conceitos Básicos
 - Introdução ao Pensamento Computacional
 - Programação em Linguagens Estruturadas
- 2 Entrada e Saída de Dados
- 3 Estruturas de Controle e Repetição
 - if / else
 - for e while
- 4 Funções, Recursividade e Modularização de Código
- 5 Estruturas de Dados
 - Listas e Tuplas
 - Matrizes
 - Dicionários
- 6 Manipulação de Arquivos
- 7 Programação Orientada a Objetos
- 8 Tratamento de Exceção

- Já usou um computador antes?
- Tem computador em casa com acesso à internet?
- Já programou antes?
- Entende inglês?
- Qual seu objetivo ao aprender a programar?

- Internet Explorer, Firefox;
- Windows Explorer, Microsoft Word;
- iTunes, Windows Media Player;
- Angry Birds, WhatsApp.

Como escrever os programas para que o computador entenda?

Níveis de linguagem

- Linguagem natural;
- **Linguagens de alto nível;**
- Linguagem de montagem (*Assembly*);
- Linguagem de máquina;

High Level Language (HLL)

- Fácil para o programador entender;
- Menos eficiente (memória);
- Simples de "debugar";
- É portátil;
- Independe da plataforma;
- Fácil de manter;
- Precisa de um compilador ou interpretador;

Low Level Language (LLL)

- Fácil para a máquina entender;
- Mais eficiente (memória);
- Difícil de "debugar";
- Não é portátil;
- Depende da plataforma;
- Difícil de manter;
- Precisa de um *assembler*;

```
print ( 'Hello , _World! ' )
```

```
#include <stdio.h>
```

```
int main ()  
{  
    printf ("Hello , _World!\n");  
    return 0;  
}
```


Assembly

```
global _start
```

```
section .text
```

```
_start:
```

```
    mov rax, 1  
    mov rdi, 1  
    mov rsi, msg  
    mov rdx, msglen  
    syscall
```

```
    mov rax, 60  
    mov rdi, 0  
    syscall
```

```
section .rodata
```

```
msg: db "Hello ,_World!", 10  
msglen: equ $ - msg
```

Hierarquia de Dados

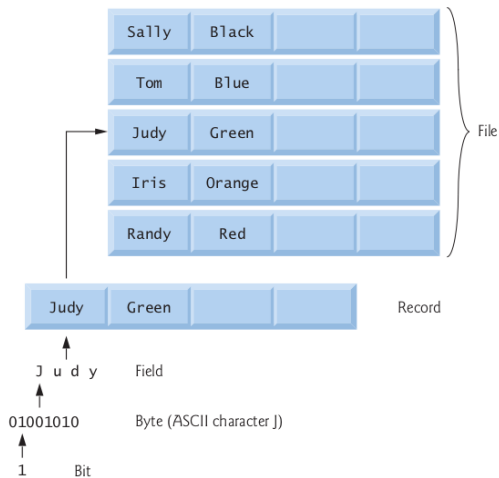


Figura 1: Hierarquia de dados [1].

Linguagens Compiladas X Linguagens Interpretadas

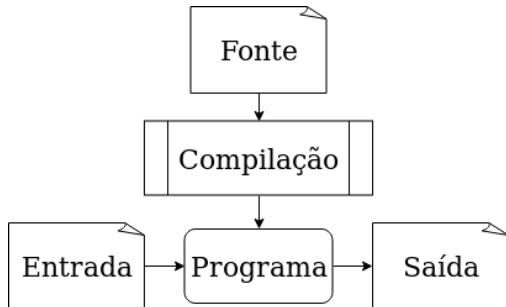


Figura 2: Compiladas

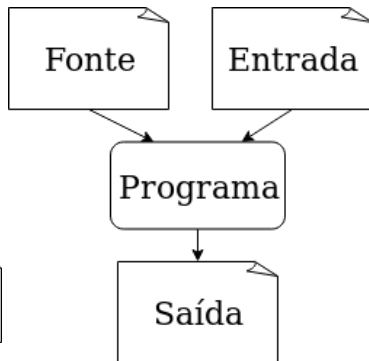


Figura 3: Interpretadas

Paradigmas

- Imperativo:
 - **Procedural** (COBOL, C, Python [2, 3, 4]);
 - **Orientado a objetos** (Python, C++, Java).
- Declarativo:
 - Funcional (Lisp, Python);
 - Lógico;
 - Matemático.

Linguagens multi-paradigmas



Hello, World!



Interagindo com o Programa



Tipos de Dados

- Inteiros (int);
- Reais (float);
- Texto (string);
- Condicionais (boolean, bool);
- Grupos "homogêneos"(arrays, vectors, lists);
- Compostos (dictionary, struct, object).

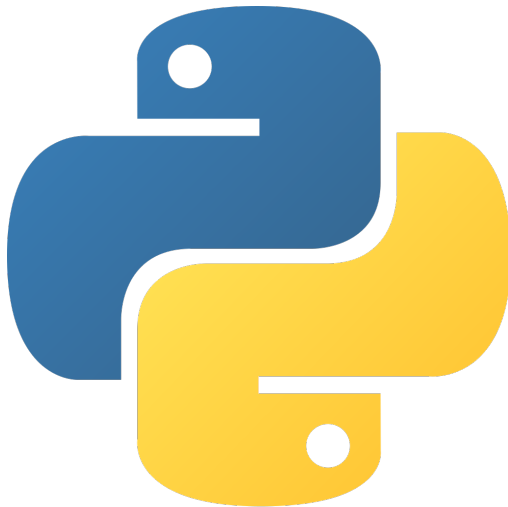
- Hello, Usuário.
- Calcular a área de um retângulo.
- Calcular o rendimento de um investimento (juros simples).



- Compiladores e interpretadores;
- Níveis de linguagem;
- Boas práticas de programação;
- Solução de problemas.



if / else



- Dada a temperatura, dizer se a sala está muito quente, quente, agradável, fria ou muito fria.
- A partir das notas e faltas, dizer a condição do aluno.



Repetição for e while



Usar os dois laços de repetição!

- Calcular a média de N notas.
- Calcular o fatorial de um número N.



- **BOAS PRÁTICAS DE PROGRAMAÇÃO!**

- Indentação;
 - Identificadores com nomes descritivos;
 - Consistência;
 - Legibilidade.
- Laços de repetição;





Escreva uma função que:

- Calcula o fatorial de N.
- Calcula área de um retângulo.
- Calcula o rendimento de um investimento (juros simples).
- Calcula a combinação simples de N elementos P a P.

$$C_p^n = n! / p!(n - p)!$$



help (tipo)

- help (str)
- len ('string')
- str.upper ('string')
- str.lower ('string')
- str.count (umaString, elemento, inicio, fim)
- str.index ('string', 'i')



- **Tuplas;**
- **Listas;**
- Pilhas;
- Filas;
- Vetores e matrizes;
- **Dicionários;**
- Grafos.



- Indexação;
- Iteração;
- Fatiamento (*slicing*);
- Concatenação;
- Replicação;

Imutabilidade de tuplas...

Funções podem retornar tuplas...



Escreva uma função que:

- Calcula a média de N notas.
- Identifica o maior elemento de uma lista.
- Remove elementos duplicados de uma lista.
- Ordena uma lista.

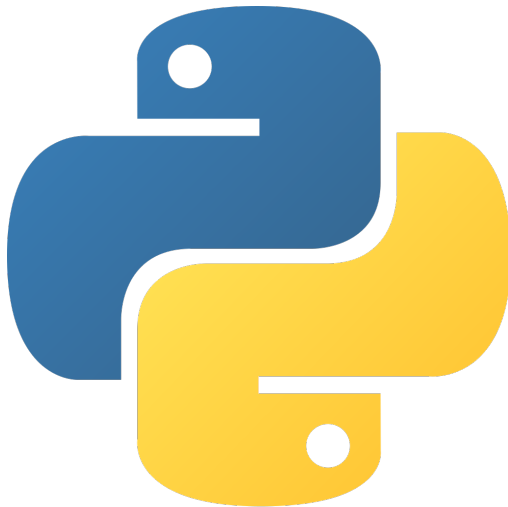












Objetivo: Abstração

Classes:

- Conceito estendido de estrutura de dados;
- Além de organizar dados, oferecem funções de manipulação;
- Encapsulamento;
- Programador não manipula a classe diretamente;
- Elementos públicos e privados;

Objetos

- Instâncias de classes;




- Atributos privados;
- Métodos *set* e *get*;
- Construtor padrão;
- Testes de validade (*set*);
- Template???








Trabalho Final!



 P. Deitel and H. Deitel, *C++ How to Program*.
Pearson, 9 ed., 2014.

 D. Beazley and B. K. Jones, *Automate the Boring Stuff with Python*.
William Pollock, 2015.

 L. Ramalho, *Fluent Python*.
O'Reilly Media, Inc., 2015.

 D. Beazley and B. K. Jones, *Python Cookbook*.
O'Reilly Media, Inc., 2013.

