



Algoritmos e Programação de Computadores

Aula 03 - Python

Prof. Dr. Rodolfo Carneiro Cavalcante
rodolfo.cavalcante@arapiraca.ufal.br
Universidade Federal de Alagoas
Campus de Arapiraca



Aulas Anteriores

- O que é um computador?
- O que há dentro de um computador?
- Qual o propósito de um sistema de computação?
- Hardware e Software
- Algoritmos

Hoje: Linguagem Python



Aulas Anteriores

- O que é um computador?
- O que há dentro de um computador?
- Qual o propósito de um sistema de computação?
- Hardware e Software
- Algoritmos

Hoje: Linguagem Python

Introdução



- Programar é:
 - Comunicar com o computador
 - Passar instruções sobre como resolver problemas
 - Projetar solução para um problema
- O computador é responsável por executar a solução projetada
 - É preciso que as partes se entendam

Introdução



- Solução de problemas
 - Pensar criativamente sobre como o problema deve ser resolvido
 - Solução deve ser expressa de forma clara e precisa
- Para programar é preciso combinar uma linguagem comum entre programador e máquina

Introdução

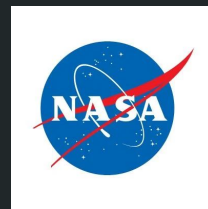


- Centenas de linguagens de programação já foram criadas desde que o computador eletrônico foi inventado
 - Algumas nem existem mais
- Linguagens famosas
 - Python, C, C++, Java, Javascript, PHP

Python



- Python é uma linguagem muito simples de se entender por humanos
- Nome faz referência ao grupo humorístico inglês “Monty Python”
- Utilizada em vários projetos importantes



Python



- Linguagem de alto nível
 - Mais próxima do programador do que da máquina
 - Similar ao inglês
- Sintaxe clara e concisa
 - Favorece a legibilidade do código-fonte
 - Linguagem de propósito geral
 - Menos linhas de código
- Portável
- Orientada a Objetos

Python



Tipos de aplicação

- aplicações da Web: Django, Flask
- desenvolvimento de software de modo geral
- automação de testes de software
- ciência de dados e machine learning (ML)

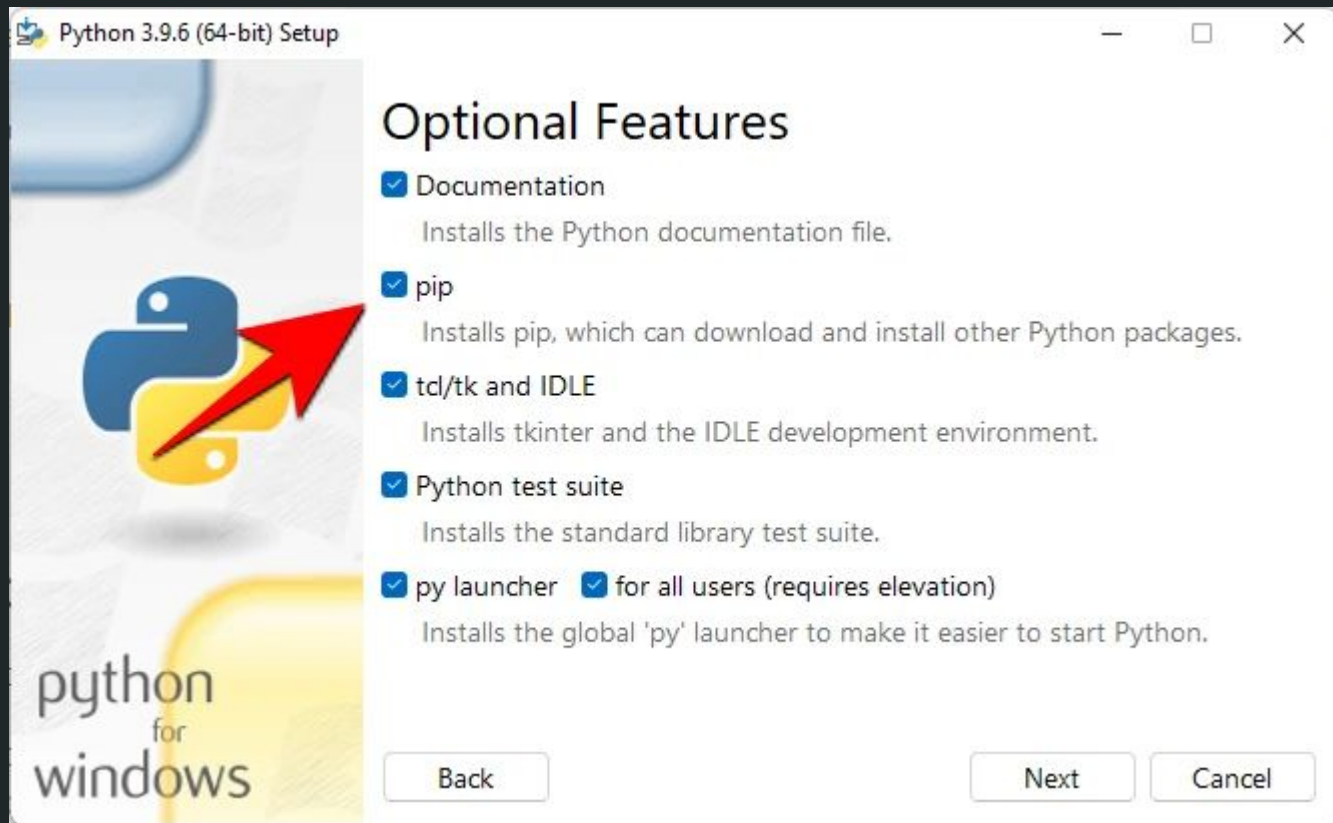
Instalação



Python - Windows

- Baixe o instalador na página oficial
 - <https://www.python.org/downloads/>
- instale utilizando o setup de instalação
- marque pip para instalação

Instalação



Instalação



Python - Linux Ubuntu

- No terminal

```
$ sudo apt-get install python3  
$ sudo apt-get install python3-pip
```

Instalação



Visual Studio Code

- acesse a página oficial e baixe a versão adequada para seu SO
- <https://code.visualstudio.com/download>

Instalação



Flask

- Framework para desenvolvimento de aplicações web
 - Python é a linguagem base
- Um framework “é uma biblioteca de códigos que torna a vida do desenvolvedor mais fácil, quando se fala de aplicativos, escaláveis e fáceis de manter”
 - Existem diversos frameworks para o Python, incluindo Flask, Tornado, Pyramid e Django

Instalação



Flask

- Utilizaremos Flask para motivar o estudo
- Instalação

```
$ pip install -U Flask
```

Instalação



Flask

- Utilizaremos Flask para motivar o estudo
- Instalação

```
$ pip install -U Flask
```


Criando Sistemas Simples



- Sistema de cálculo de aprovação

Programa



- Às vezes considerado sinônimo de algoritmo
- Algoritmo é mais abstrato
- Relacionado com projeto de solução do problema
- Programa é uma sequência de instruções que especificam como executar uma computação
- Concreto
- Escrito em uma linguagem de programação

Programa



- Começaremos com programas simples
 - sem interface gráfica

Primeiro Código



```
from flask import Flask #importe bibliotecas

app = Flask (__name__) #iniciar aplicação

@app.route('/') #criar rota principal
def main():
    return 'Alô, mundo'

if __name__ == '__main__':
    app.run() #executa a aplicação
```

- acesse no browser: localhost:5000

Primeiro Código



```
from flask import Flask #importe bibliotecas

app = Flask (__name__) #iniciar aplicação

@app.route('/') #criar rota principal
def main():
    return 'Alô, mundo'

if __name__ == '__main__':
    app.run() #executa a aplicação
```

- acesse no browser: localhost:5000



Primeiro Código

- Você pode dar qualquer nome para seu programa
 - É preciso que ele tenha a extensão “.py”
- Vamos simplificar e não usar o Flask por enquanto

```
def main():  
    print('Alô, mundo')  
  
if __name__ == '__main__':  
    main()
```

- execute no terminal `python alomundo.py`

Primeiro Código



- Quais as saídas se adicionarmos no main?

```
print('meu nome é Pedro')  
print(2+3)  
print(2*10)  
print(2**5)
```



Variáveis

- Espaços de memória para guardar valores
 - Para manipulação no programa
- Variáveis podem guardar valores de vários tipos
 - inteiro
 - float
 - caractere
 - String
 - ...
- “Mesa do bolo”

003EF788			004D4A90	0104A000		0012CCG8		003EF78A
10			4.5	Livro		-3.8		A

Variáveis



- Nomes de variáveis
 - facilitam a manipulação no programa

```
meu_inteiro = 4
print(meu_inteiro)
meu_float = 3.4
print(meu_float)
meu_caract = 'r'
print(meu_caract)
minha_string = 'minha string'
print(minha_string)
```

Variáveis



- Nomes de variáveis
- idade = 10
- altura = 4.5
- item = 'Livro'
- temperatura = -3.8
- conceito = 'A'

idade			altura	item		temperatura		conceito
10			4.5	Livro		-3.8		A

Variáveis



- Quando o computador encontra uma instrução de criação de variável
 - Ele reserva um espaço de memória para aquela variável
 - Guarda o valor atribuído
- Variável pode ter **quase** qualquer nome
 - É interessante que seja um nome significativo para a lógica do programa
- Existem regras para nomeação de variáveis

Variáveis



Regras para nomes de variáveis

- Podem conter letras e números

```
nome1 = 'bob'  
nome2 = 'carlos'  
idade1abc = 5  
nome_completo = 'josé da silva'
```

Variáveis



Regras para nomes de variáveis

- Precisa começar com letra

```
1nome = 'ana' #ERRO
```

- ~\$ `SyntaxError: invalid syntax`

Variáveis



Regras para nomes de variáveis

- Existem algumas palavras reservadas que não podem ser

and	def	exec	if	not
return	break	import	or	for
for	from	class	try	print
except	else	while	elif	del



Entradas do Usuário

- Uma tarefa muito comum dos programas é receber entradas do usuário
- Uma forma de interagir com o programa é digitando entradas por meio do teclado
- Recebendo uma string

```
nome = input('Digite seu nome: ')\nprint(nome)
```



Entradas do Usuário

- Recebendo valor numérico

```
idade = int(input('Digite sua idade: '))  
print(idade)  
nota = float(input('Digite a nota: '))  
print(nota)
```




Entradas do Usuário

- Programa para calcular a área do retângulo

```
lado1 = int(input('Digite lado 1: '))  
lado2 = int(input('Digite lado 2: '))  
area = lado1*lado2  
print('A área do retângulo é',area)
```



Entradas do Usuário

- Programa para calcular a média do aluno

```
nota1 = float(input('digite nota 1: '))  
nota2 = float(input('digite nota 2: '))  
media = (nota1+nota2)/2  
print(media)
```

Comandos



- Instruções que o computador consegue executar
- Vimos alguns:
 - Atribuição: atribuir valores a variáveis
 - Exibir na tela: print
 - Entrada do usuário: input
- Programas são compostos por um conjunto de comandos

Comandos



- Programa que tem duas variáveis e calcula e imprime na tela a soma entre elas

```
a = 5  
b = 8  
soma = a + b  
print(soma)
```

Comandos



- Programa que tem duas variáveis e calcula e imprime na tela a soma entre elas

```
a = int(input('digite o primeiro valor: '))  
b = int(input('digite o segundo valor: '))  
print (a + b)
```

Operadores



- Símbolos utilizados para construção de expressões
- Aritméticos
 - `+`, `-`, `*`, `/`, `//`, `**`, `%`
- Lógicos
 - `and`, `or`, `not`, `==`, `!=`

Operadores



- Símbolos utilizados para construção de expressões
- Aritméticos
 - `+`, `-`, `*`, `/`, `//`, `**`, `%`
- Lógicos
 - `and`, `or`, `not`, `==`, `!=`

Exercícios



- Programa para calcular a área de um triângulo retângulo
 - entradas: base e altura
- Programa para converter temperatura de Fahrenheit para Celsius
 - entrada: temperatura em Celsius
- Programa para converter dolar para real
 - entradas: valor e taxa de câmbio
- Programa para calcular área de uma circunferência
 - entrada: raio

Exercícios



- Programa para calcular o volume de um cilindro
 - entradas: raio e altura
- Programa que calcula o pagamento com atraso de um boleto bancário
 - O valor final a ser pago é dado por
 - $\text{Total} = \text{valor} + \text{multa} + (\text{valor} * \text{juros} * \text{dias})$
 - Sabendo que a multa é de 2 e a taxa de juros é de 0,05



Algoritmos e Programação de Computadores

Aula 03 - Python

Prof. Dr. Rodolfo Carneiro Cavalcante
rodolfo.cavalcante@arapiraca.ufal.br