Fatec
Araras

Curso de Python

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

"Programar é como cozinhar: em Python, você usa molho à bolonhesa pré-fabricado; em C++, você começa com tomates frescos e carne picada; em Assembly, você tem uma fazenda onde você cultiva seus tomates e cria sua vaca"

@gv_barroso

Fonte: https://twitter.com/CodeWisdom

Objetivos para hoje

Nos conhecermos

Conhecer a linguagem python

Criar primeiros scripts

O que é programa de computador ?

Fatec Araras

Um programa é uma sequência de instruções que especifica como realizar uma computação. O computação pode ser algo matemático, como resolver um sistema de equações ou encontrar as raízes de um polinômio, mas também pode ser uma computação simbólica, como e substituir texto em um documento ou compilar um programa.

Os detalhes parecem diferentes em diferentes linguagens, mas algumas instruções básicas aparecem em todas as linguagens:

- Entrada de dados
- Saída de dados
- Computação e operações matemáticas
- Condição de execução
- Repetição

Python

O que é python ?

Python é uma linguagem de programação de alto nível, interpretada, de *script*, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte.

Foi lançada por Guido van Rossum em 1991. Atualmente possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation.

Fonte: wikipedia

Linguagem de alto nível

Fatec Araras

Linguagem de alto nível:

Uma linguagem de programação projetada para ser fácil humanos para ler e escrever.

Linguagem de baixo nível:

Uma linguagem de programação projetada para ser fácil para um computador executar; também chamado de "linguagem de máquina" ou "linguagem de montagem".

Portável e interpretada

Fatec Araras

Portabilidade

Uma propriedade de um programa que pode ser executado em mais de um tipo de computador.

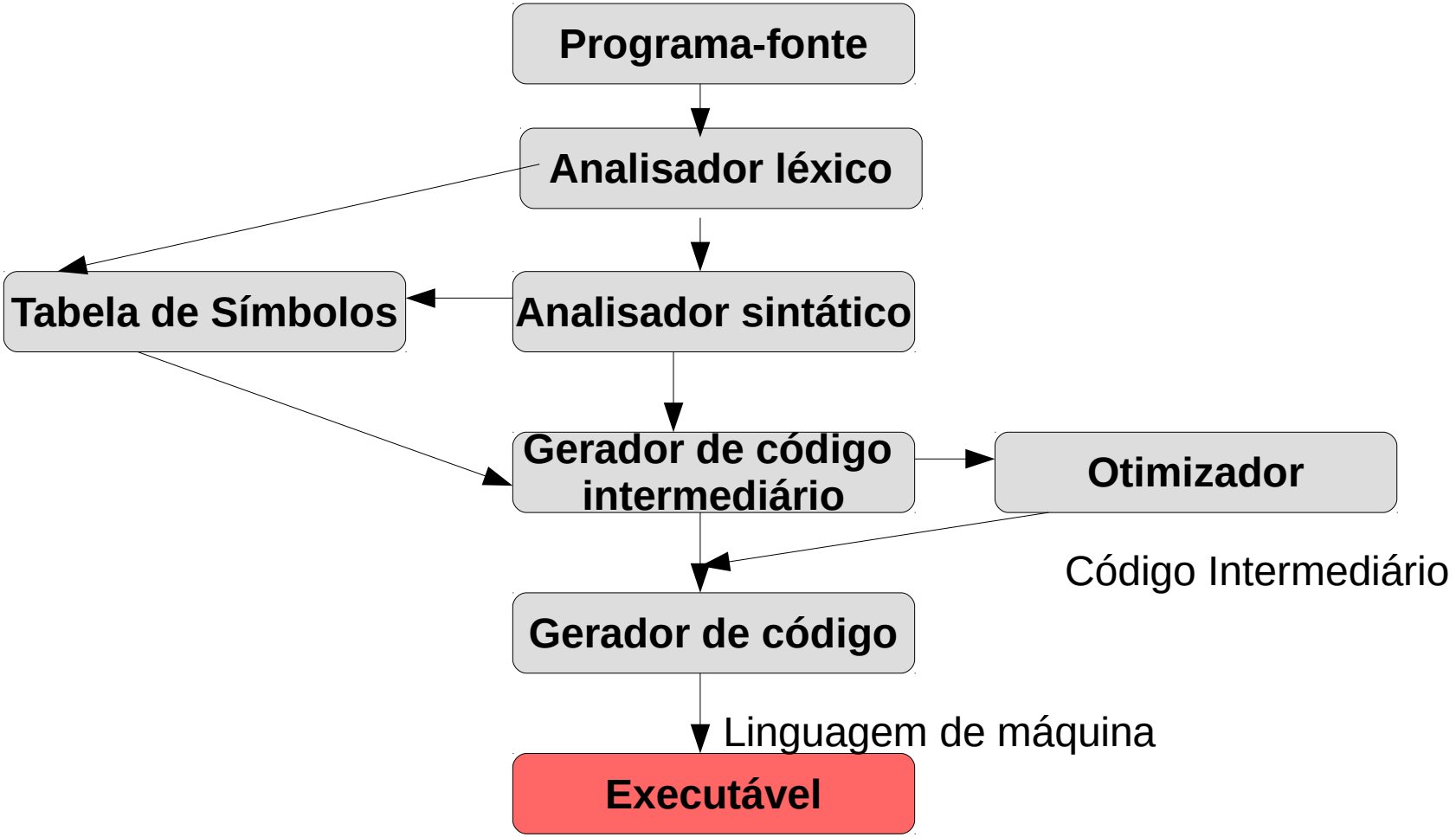
Linguagem Interpretada

Linguagem que faz uso de um interpretador para executar um programa em uma linguagem de alto nível, traduzindo-o uma linha por vez.

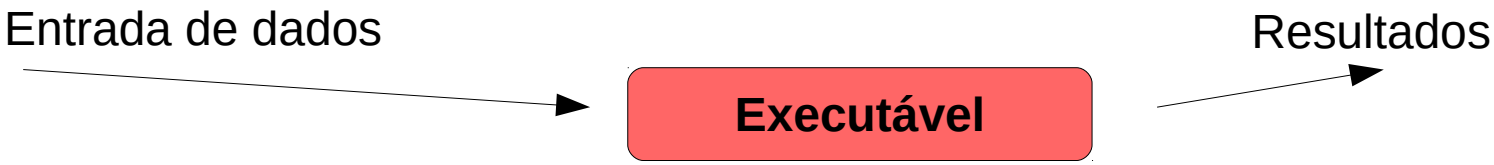
Linguagem Compilada

Linguagem que faz uso de um compilador. O compilador traduz um programa escrito em uma linguagem de alto nível em uma linguagem de baixo nível. O produto gerado em baixo nível é executado posteriormente.

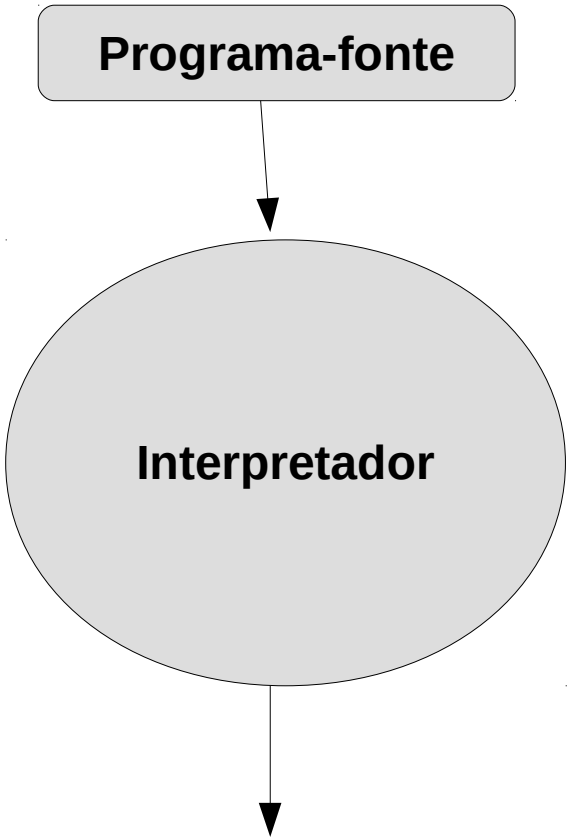
Como se compila um programa



Programa Compilado



Programa Interpretado



Paradigmas de programação

Programação imperativa

É um paradigma de programação que descreve a computação como ações, enunciados ou comandos que mudam o estado (variáveis) de um programa.

Programação Funcional

É um paradigma de programação que trata a computação como uma avaliação de funções matemáticas,evitando mutação de dados e estados.

Programação Orientada a Objetos

É um paradigma de programação baseado na composição e interação entre diversas unidades chamadas de 'objetos'.

Tipagem de dados

Fatec Araras

Python não permite mudança de tipo para possibilitar a atribuição de um valor. Esta é a característica das linguagens de tipagem forte.

Não é necessário declaração prévia dos tipos das variáveis. Esta é a característica das linguagens de tipagem dinâmica.

"Talk is cheap. Show me the code"

Linus Torvalds

Programa oi mundo

Fatec Araras

Quais as semelhanças e diferenças entre o programa **oi mundo** nas linguagens apresentadas ?

Primeiro contato com Python

Fatec Araras

Abra o interpretador python.

>>>

Digite

print('oi mundo')

Variáveis

Fatec Araras

Um dos recursos mais poderosos de uma linguagem de programação é a capacidade de manipular variáveis. Uma variável é um nome que se refere a um valor. Com uma declaração de atribuição consegue-se criar novas variáveis.

Nomes variáveis podem ser arbitrariamente longos. Eles podem conter letras e números, mas eles têm que começar com um caractere.

É possível usar letras maiúsculas, mas é uma boa ideia começar os nomes das variáveis com uma letra minúscula.

Variáveis
Prática

Fatec Araras

```
>>> 8teste = 1
  File "<stdin>", line 1
    8teste = 1
SyntaxError: invalid syntax
```

```
>>> nome@ = 1
  File "<stdin>", line 1
    nome@ = 1
SyntaxError: invalid syntax
```

```
>>> class = 1
  File "<stdin>", line 1
    class = 1
SyntaxError: invalid syntax
```

Variáveis Numéricas

Além de int e float, o Python suporta outros tipos de números, como Decimal e Fraction. Python também tem suporte embutido para números complexos

Variáveis Numéricas
Prática

```
>>> 2 + 2
4
>>> a = 2 + 2
>>> a
4
>>> type(a)
<class 'int'>
```

```
>>> id(a)
10914464
>>> a = a / 2
>>>_
1.333333333333333
>>> type(a)
<class 'float'>
```

Variáveis Numéricas
Prática

```
>>> 4 * 2
8
>>> 4 ** 2
16
>>> 20 / 3
6.66666666666
>>> 20 // 3
6
```

```
>>> b = 5 + 2j
>>> b
(5+2j)
>>> type(b)
<class 'complex'>
```

Strings

Fatec Araras

Dados textuais em Python são manipulados com objetos str ou strings. Strings são sequencias imutáveis de pontos de código Unicode.

Strings são escritos de várias maneiras, conforme veremos na prática:

Variáveis String
Prática

```
>>> s1 = 'Permite embutir aspas "duplas" '
>>> s2 = "Permite embutir aspas 'simples' "
>>> s3 = '''Tres aspas simples '''
>>>  s4 = """tres aspas duplas"""
>>>  dir(s1)
>>>  s1 * 2
```

Variáveis String
Prática

Fatec Araras

```
>>> s1
'Permite embutir aspas "duplas" '
>>> s1 + 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Can't convert 'int' object to str implicitly
>>> s1 + str(2)
>>>  s1[0:13]
>>>  s1[23:-2]
```

Variáveis String
Prática

```
>>>numero = 12
>>> numero.isnumeric()
True
>>> s1.isnumeric()
False
>>>>>>> nome = 'Orlando Saraiva Junior'
>>> nome.split(' ')
['Orlando', 'Saraiva', 'Júnior']
>>> help(str)
```

Listas

O Python conhece vários tipos de dados compostos, usados para agrupar outros valores.

A mais versátil é a lista, que pode ser escrita como uma lista de valores separados por vírgula (itens) entre colchetes. As listas podem conter itens de tipos diferentes, mas geralmente os itens têm o mesmo tipo.

Lista é um tipo de sequencias mutáveis.

Variáveis Lista
Prática

```
>>> l1 = []
>>> for x in range(10):
....        l1.append(x)
>>>
>>> l1
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>  l1[3:6]
[3, 4, 5]
>>>  l1.reverse()
```

Variáveis Lista e Tupla

Prática

```
>>> l2 = [-1, 1, 66.25, 333, 333, 1234.5]
>>> del(l2[0])
>>> l2
[1, 66.25, 333, 333, 1234.5]
>>> type(l2)
<class 'list'>
>>> tupla1 = tuple(l2)
>>> tupla1
(1, 66.25, 333, 333, 1234.5)
>>> type(tupla1)
<class 'tuple'>
```

Tuplas

As **tuplas** são sequencias imutáveis, normalmente usadas para armazenar coleções de dados heterogêneos.

O tipo **range** representa uma sequência imutável de números e é comumente usado para looping um número específico de vezes em loops for.

Set é um conjunto é uma coleção não ordenada sem elementos duplicados. Usos básicos incluem testes de associação e eliminação de entradas duplicadas.

Tupla, Range e Sets
Prática

```
>>> lista = []
>>> for x in range(1,20,2):
...     lista.append(x)
...     lista.append(x)
...     lista.append(x)
...
>>> lista
```

[1, 1, 1, 3, 3, 3, 5, 5, 5, 7, 7, 7, 9, 9, 9, 11, 11, 11, 13, 13, 13, 15, 15, 15, 17, 17, 17, 19, 19, 19]

Tupla, Range e Sets
Prática

Fatec Araras

```
>>> lista
[1, 1, 1, 3, 3, 3, 5, 5, 5, 7, 7, 7, 9, 9, 9, 11, 11, 11, 13, 13, 13, 15, 15, 15, 17, 17, 17, 19, 19, 19]
>>> set_unico = set(lista)
>>> set_unico
{1, 3, 5, 7, 9, 11, 13, 15, 17, 19}
>>> tupla = tuple(lista)
>>> tupla
(1, 1, 1, 3, 3, 3, 5, 5, 5, 7, 7, 7, 9, 9, 9, 11, 11, 11, 13, 13, 13, 15, 15, 15, 17, 17, 17, 19, 19, 19)
>>>
```

Dicionários

Os dicionários são encontrados em outras linguagens como “memórias associativas” ou “matrizes associativas”.

Ao contrário das sequências, que são indexadas por um intervalo numérico, os dicionários são indexados por chaves, que podem ser de qualquer tipo imutável.

Dicionários
Prática

```
>>> agenda = {}
>>> agenda['orlando'] = ['Orlando Saraiva',123456]
>>> agenda
{'orlando': ['Orlando Saraiva', 123456]}
>>> agenda['lucas'] = ['Lucas Mendonça',98712212, ["Python","Linux"],]
>>> del agenda['lucas'][2][1]
>>> agenda
{'lucas': ['Lucas Mendonça', 98712212, ['Python']], 'orlando': ['Orlando Saraiva', 123456]}
```

Instruções compostas

Fatec Araras

Instruções compostas contêm (grupos de) outras instruções. Estas afetam ou controlam a execução dessas outras declarações de alguma forma.

Em geral, as instruções compostas abrangem várias linhas, embora uma declaração composta inteira possa estar contida em uma linha.

As instruções **if**, **while** e **for** implementam construções de fluxo de controle tradicionais. **try** especifica manipuladores de exceção e / ou código de limpeza para um grupo de instruções

Instrução If
Prática

```
>>> x = int(input("Digite um número: "))
Digite um número: 30
>>> if x < 0:
...     x = 0
...     print('Negativo alterado para zero')
... elif x == 0:
...     print('Zero')
... else:
...     print('Mais de um')
...
Mais de um
>>>
```

Instrução for
Prática

```
>>> lista_nomes = ['Gilmar','Pedro','Michel','Carlos']
>>> for x in lista_nomes:
...     print(x,len(x))
...
Gilmar 6
Pedro 5
Michel 6
Carlos 6
>>>
```

Instrução while
Prática

```
>>> x = 10
>>> while x > 0 :
...     print(x)
...     x = x - 1
...
>>>
```

Lendo e gravando arquivos

A função open() retorna um objeto de arquivo e é mais comumente usado com dois argumentos:

open (filename, mode).

O primeiro argumento é uma string contendo o nome do arquivo. O segundo argumento é outra string contendo alguns caracteres descrevendo a maneira como o arquivo será usado.

'r' → Quando o arquivo só será lido

'w' → Somente por escrito

'a' → Abre o arquivo para acrescentar (*append*)

Arquivos
Prática

```
>>> try:
...     arquivo = open('nome.txt', 'r')
... except FileNotFoundError:
...     print("Arquivo inexistente")
...
Arquivo inexistente
>>>
```

Arquivos
Prática

Fatec Araras

```
>>> try:
...     arquivo = open('nomes.txt', 'r')
... except FileNotFoundError:
...     print("Arquivo inexistente")
...
>>> arquivo
<_io.TextIOWrapper name='nomes.txt' mode='r' encoding='UTF-8'>
>>> for linha in arquivo:
...     print(linha)
...
>>> dir(arquivo)
```

Fatec
Araras

Dúvidas

Prof. Orlando Saraiva Júnior
orlando.saraiva@unesp.br

Fechamento
Exercício

Fatec Araras

Separadando os nomes

Implemente um programa que leia todos os nomes do arquivo nomes.txt e crie um arquivo chamado nomeSeparados.txt com o seguinte formato

Arquivo entrada

```
Orlando
Lucas
Ana
```

Arquivo saída

```
| O | R | L | A | N | D | O |
| L | U | C | A | S |
| A | N | A |
```