

MiniCurso Python



SECOMP 2019 - UTFPR Cornélio Procópio

Roteiro

- Apresentação do curso e dos alunos;
- História e características da linguagem;
- Áreas de aplicação;
- Sintaxe básica;
- Estruturas de dados;
- Módulos importantes;
- Criação de um *web scraper*.

0 curso

História da linguagem

Guido Van Rossum



Python

- Linguagem de propósito geral;
- Fácil e intuitiva;
- Multiparadigmas;
- Multiplataforma;
- Interpretada;
- Biblioteca padrão poderosa;
- Código aberto.

Áreas de aplicação

- Inteligência artificial;
- Web;
- Automação residencial;
- Análise de dados;
- Ensino de lógica de programação.

Python 2 e 3

PYTHON 2

`print "hello"`

Python 2 print statement



`5/2=2`

It rounds your calculation down to the nearest whole number

0100
0001

ASCII

Strings are stored as ASCII by default



Library

Many older libraries built for Python 2 are not forwards-compatible



Legacy

It is still entrenched in the software at certain companies

≠

`print ("hello")`

The print statement has been replaced with a `print ()` function

≠

`5/2=2.5`

The expression `5 / 2` will return the expected result



+

Unicode

Text strings are Unicode by default

0000
0000
0100
0001

≠

Library

Many of today's developers are creating libraries strictly for use with Python 3



>

Future



It will take over Python 2 by 2020

PYTHON 3

Sintaxe básica

Python vs. C: “Ola Mundo”

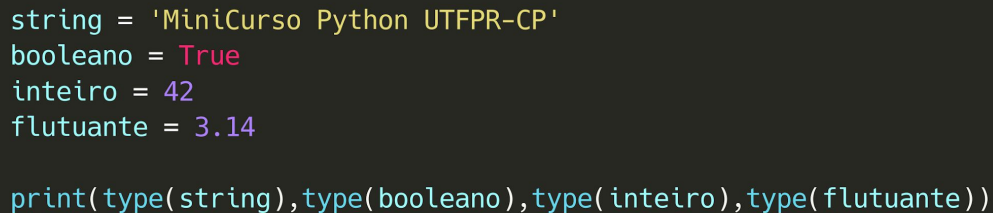


```
def main():  
    print('Ola Mundo')  
  
if __name__ == "__main__":  
    main()
```

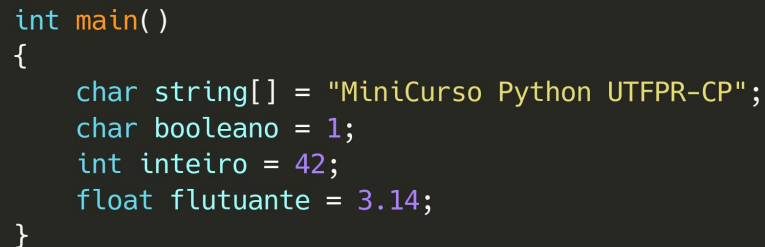


```
#include <stdio.h>  
  
int main()  
{  
    printf("Ola Mundo\n");  
}
```

Python vs. C: Variáveis

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains Python code for variable declaration and type checking.

```
string = 'MiniCurso Python UTFPR-CP'  
booleano = True  
inteiro = 42  
flutuante = 3.14  
  
print(type(string), type(booleano), type(inteiro), type(flutuante))
```

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains C code for variable declaration within a main function.

```
int main()  
{  
    char string[] = "MiniCurso Python UTFPR-CP";  
    char booleano = 1;  
    int inteiro = 42;  
    float flutuante = 3.14;  
}
```

Python vs. C: Entrada e saída

[illegible]

Python vs. C: Entrada e saída



```
1 nome = input("Digite o seu nome!\n")
2 idade = int(input("Digite a sua idade!\n"))
3 altura = float(input("Digite sua altura(metros)!\n"))
4
5 print(f'Olá {nome} você tem {idade} anos e {altura:.2f}m de altura')
```

Atividades

Entrada e saída

- Faça um programa que leia uma *string* e imprima na tela;
- Faça um programa que leia dois números inteiros, faça a soma e mostre o resultado;
- Faça um programa que leia dois números reais, faça a divisão e mostre o resultado.

Tipos mutáveis e não-mutáveis

Tipos mutáveis e não-mutáveis

| Class | Description | Immutable? |
|------------------|--------------------------------------|------------|
| bool | Boolean value | ✓ |
| int | integer (arbitrary magnitude) | ✓ |
| float | floating-point number | ✓ |
| list | mutable sequence of objects | |
| tuple | immutable sequence of objects | ✓ |
| str | character string | ✓ |
| set | unordered set of distinct objects | |
| frozenset | immutable form of set class | ✓ |
| dict | associative mapping (aka dictionary) | |

Estruturas de controle



```
value = True
```

```
if(value):  
    print("Value eh True")  
else:  
    print("Value nao eh True")
```

Estruturas de repetição



```
for item in lista_itens:  
    print(f'Item: {item}')
```



```
contador = 0  
  
while contador < 10:  
    #Faz outras coisas  
    print(f'Contador está em {contador}')
```

Atividades

Controle e Repetição

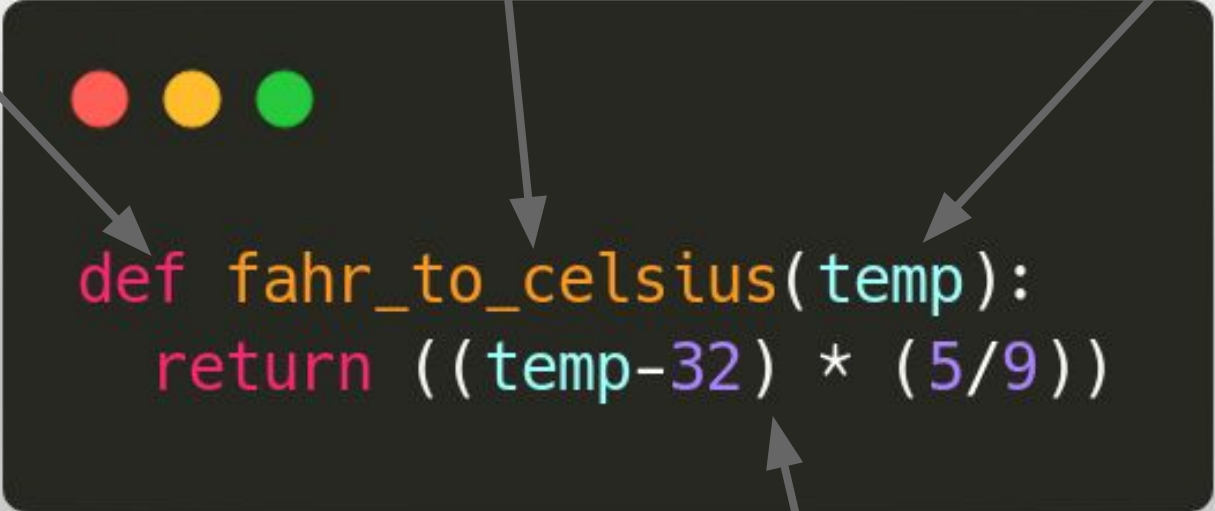
- Faça um script que, dependendo do estado de uma variável (True ou False), mostre uma mensagem diferente na tela;
- Faça um script que mostre se um número digitado pelo usuário é positivo, negativo ou igual a 0;
- Usando a estrutura FOR faça um script que mostre uma sequência numérica na tela;
- Crie uma lista e, usando a estrutura FOR, mostre essa lista na tela.

Funções

Parâmetro de
definição

Nome da Função

Parâmetros



```
def fahr_to_celsius(temp):  
    return ((temp-32) * (5/9))
```

Valor de Retorno

Funções

- Argumentos nomeáveis;
- Retorno de mais de um valor.

Atividades

Funções

- Crie uma função que receba 2 números e retorne a soma deles;
- Crie uma função que receba 2 números e retorne a soma e a subtração ao mesmo tempo;
- Crie uma função que receba 2 números e retorne a soma deles, chame essa função passando o nome dos parâmetros;
- Crie uma função que receba 2 números e retorne a soma deles, porém essa função deve ter valores padrões para os parâmetros.

Arquivos



```
nome_arquivo = 'nome_do_arquivo.txt'
```

```
modo_leitura = 'r'
```

```
with open(nome_arquivo, modo_leitura) as arquivo:  
    print(arquivo.read())
```

Atividades

Arquivos

- Faça um programa que cria um arquivo e escreve no mesmo;
- Utilizando o arquivo criado na primeira atividade, leia o conteúdo do arquivo;

Estruturas de dados

Estruturas de dados

- Listas;
- Tuplas;
- Conjuntos;
- Dicionários;
- Classes.

Estrutura de Dados - Lista

58,14,2,69,87,68,45,21,0

- Crie uma lista com os números acima e imprima cada número em uma linha acompanhado da sua posição na lista;
 - EX:
 - 1 - 58
 - 2 - 14
 - 3 - 2
- Ordene a lista e imprima a nova lista;
- Adicione o número “-1” à lista;
- Concatene a lista **[10,20,30,40]** à lista anterior;
- Imprima o maior e o menor item da lista;

Estrutura de Dados - Tupla e Conjunto

58,14,2,69,87,68,45,21,0

- Crie uma lista com os números acima e transforme essa lista em uma tupla e tente alterar um elemento (SPOILER: Vai dar erro);
- Concatene a lista **[58,69,45,2,-1,-2,-3]** à lista anterior e transforme em um Conjunto, imprima o resultado;

Módulos

Módulos

- PIP;
- CSV;
- Regex;
- JSON;
- Matplotlib;
- requirements.txt (controle de dependências);
- Faça seu próprio módulo!

Obrigado!