



Bem-Vindx à Comunidade DS

Fundamentos de Programação com Python

Aula #4 - Funções, POO e
Operações em Objetos

TÓPICOS DA AULA #4

1. Revisão da Aula Anterior
2. Resolução dos Exercícios Anteriores
3. Funções
4. POO - Programação Orientada à Objetos
5. Operação em Objetos Funções Built-in
6. Exemplos
7. Exercícios

1. REVISÃO DA AULA ANTERIOR

REVISÃO DA AULA ANTERIOR

Estruturas de Dados - Variáveis “Complexas”:

- Lista
- Dicionário
- Tupla
- Range

REVISÃO DA AULA ANTERIOR

Input/Output - I/O:

- Abertura e Fechamento de Arquivos
- Leitura de Arquivos

REVISÃO DA AULA ANTERIOR

Estruturas de Repetição:

- *for*
- *while*
- Como Iterar sobre estruturas

2. RESOLUÇÃO DOS EXERCÍCIOS

RESOLUÇÃO DOS EXERCÍCIOS

IT'S CODING TIME!

—

3. FUNÇÕES

FUNÇÕES

- Blocos de Códigos que são “executáveis” e possuem um nome!
- Quase tudo que se repete, pode ser uma Função

VARIÁVEIS COMPLEXAS

IT'S CODING TIME!

4. POO - PROGRAMAÇÃO ORIENTADA À OBJETOS

POO - PROGRAMAÇÃO ORIENTADA À OBJETOS

- Paradigma de Desenvolvimento de Software
- Menos Repetição de Blocos de Código
- Mais Próximo do Mundo Real
- Manutenção Facilitada

POO - PROGRAMAÇÃO ORIENTADA À OBJETOS

A POO é baseada em:

- Classes
- Objetos
- Atributos
- Métodos

POO - PROGRAMAÇÃO ORIENTADA À OBJETOS

IT'S CODING TIME!

5. OPERAÇÕES EM OBJETOS E BUILT-IN FUNCTIONS

OPERAÇÕES EM OBJETOS E BUILT-IN FUNCTIONS

Tudo dentro do Python são Objetos!

```
class int([x])
```

```
class int(x, base=10)
```

Return an integer object constructed from a number or string x, or return 0 if no arguments are given. If x defines `__int__()`, `int(x)` returns `x.__int__()`. If x defines `__index__()`, it returns `x.__index__()`. If x defines `__trunc__()`, it returns `x.__trunc__()`. For floating point numbers, this truncates towards zero.

OPERAÇÕES EM OBJETOS E BUILT-IN FUNCTIONS

Modificação de tipo = Instanciar uma classe!

```
1  # Criando uma variável com o valor 20
2  numero = '20'
3  print(numero)
4  print(type(numero))
5
6  # Pula Linha
7  print()
8
9  # Utilizando a conversão para transformar o dados em inteiro
10 num_convertido = int(numero)
11 print(num_convertido)
12 print(type(num_convertido))
```

executed in 3ms, finished 15:26:20 2021-10-01

```
20
<class 'str'>
```

```
20
<class 'int'>
```

OPERAÇÕES EM OBJETOS E BUILT-IN FUNCTIONS

IT'S CODING TIME!

6. EXEMPLOS

EXEMPLOS

Exemplo 1:

Considere um arquivo de entrada no formato CSV (comma separated values) com informações relativas a acidentes na aviação civil brasileira nos últimos 10 anos (arquivo anv.csv)

As informações estão separadas pelo caracter separador ~ e entre “” (aspas) conforme o exemplo abaixo:

```
"201106142171203"~"PPGXE"~"AERoclUBE"~"AVIÃO"~"NEIVA INDUSTRIA  
AERONAUTICA"~"56-C"~"PAUL"~"PISTÃO"~"MONOMOTOR"~"660"~"LEVE"~"2"~"1962"~"BRASIL"~"BRASIL"~"PRI"~"INSTRUÇÃO"~  
"SDPW"~"SDPW"~"INDETERMINADA"~"UNKNOWN"~"VOO DE INSTRUÇÃO"~"SUBSTANCIAL"~"0"~"2018-07-09"
```

```
"201707111402595"~"PPNCG"~"OPERADOR PARTICULAR"~"AVIÃO"~"PIPER  
AIRCRAFT"~"PA-46-350P"~"PA46"~"TURBOÉLICE"~"MONOMOTOR"~"1950"~"LEVE"~"6"~"1990"~"NULL"~"BRASIL"~"TPP"~"PARTI  
CULAR"~"SBBR"~"SBGR"~"POUSO"~"LANDING"~"VOO PRIVADO"~"NENHUM"~"0"~"2018-07-09"
```

O arquivo é composto das seguintes colunas por 25 Colunas de descrevem cada uma das variáveis.

EXEMPLOS

Exemplo 1:

Crie uma função que efetue a leitura do arquivo, sem a utilização de bibliotecas externas, e processe o arquivo produzindo dois novos arquivos.

Arquivo 1

O primeiro arquivo deve ter seu conteúdo em formato JSON, com o nome `statistics.json`, e deve possuir as estatísticas:

- fase de operação
- número de total de ocorrências
- percentual de quanto essa fase representa dentro de todos os dados

EXEMPLOS

Exemplo 1:

Exemplo de como deve estar o arquivo:

```
[
  {
    "fase_operacao": "APROXIMAÇÃO FINAL",
    "ocorrencias": 234,
    "percentual": "4,51%"
  },
  {
    "fase_operacao": "INDETERMINADA",
    "ocorrencias": 180,
    "percentual": "2,43%"
  },
  {
    "fase_operacao": "MANOBRA",
    "ocorrencias": 80,
    "percentual": "0,95%"
  }
]
```

EXEMPLOS

Exemplo 1:

Arquivo 2

Crie um arquivo de saída (formato CSV) com nome levels.csv contendo as seguintes informações:

- operation -> aeronave_operador_categoria
- type -> aeronave_tipo_veiculo
- manufacturer -> aeronave_fabricante
- engine_type aeronave_motor_tipo
- engines -> aeronave_motor_quantidade
- year_manufacturing -> aeronave_ano_fabricacao
- seating -> aeronave_assentos
- fatalities -> total_fatalidades

EXEMPLOS

Exemplo 1:

Considerando apenas acidentes cujo nível de dano da aeronave tenha sido LEVE ou NENHUM (coluna `aeronave_nivel_dano`) e que o número de fatalidades (`total_fatalidades`) tenha sido superior à 0 (zero)

OBS: Efetuar apenas uma leitura do arquivo de entrada

EXEMPLOS

IT'S CODING TIME!

7. EXERCÍCIOS

EXERCÍCIOS

Exercício 1:

Crie uma função que efetue a leitura do arquivo produzido no Exemplo 1 (levels.csv) e mostre na tela as seguintes informações estatísticas:

- Quantidade total de acidentes
- Quantidade total de acidentes agrupados por tipo de aeronave (`type`)
- Ano e fabricante da aeronave mais antiga (`year_manufacturing`, `manufacturer`)
- Quantidade de assentos e tipo de motor da aeronave mais nova (`seating`, `engine_type`)
- A quantidade de acidentes que ocorreram com aeronaves do tipo (`type`) HELICÓPTERO, cuja fabricação se deu após o ano 1997
- A quantidade de acidentes que ocorreram com aeronaves do tipo (`type`) AVIÃO fabricadas (`manufacturer`) pela CESSNA AIRCRAFT

EXERCÍCIOS

IT'S CODING TIME!

Os exercícios estarão dentro do Jupyter Notebook!

PRÓXIMA AULA

PRÓXIMA AULA

- Função Assert
- Função try .. except
- Tratamento de Erros

PERGUNTAS E COMENTÁRIOS?

FONTES EXTERNAS

FONTES EXTRAS

- Built-in Types — Python 3.9.7 documentation
- Defining Your Own Python Function – Real Python
- 4. More Control Flow Tools — Python 3.9.7 documentation
- Python Functions (def): Definition with Examples
- PEP 257 -- Docstring Conventions

FONTES EXTRAS

- Python Closures: How to use it and Why?
- Python *args and **kwargs (With Examples)
- Python args and kwargs: Demystified – Real Python
- 10 Examples to Master *args and **kwargs in Python
- Python: entendendo o uso de *args e **kwargs em funções e métodos

FONTES EXTRAS

- [Dunder/Magic Methods in Python | Engineering Education \(EngEd\) Program](#)
- [Python `__str__\(\)` and `__repr__\(\)` functions](#)
- [Built-in Types — Python 3.9.7 documentation](#)
- [Built-in Functions — Python 3.9.7 documentation](#)