

Atividade S2 A1

Nome: Adalberto Caldeira Brant Filho

Repositório GitHub: <https://github.com/adalbertobrant/lipai>

Código das Videoaulas

1. Instalação do ambiente alternativo ao vscode

Instalação do vscodium para ubuntu usando o snap:

```
snap install codium --classic
```

2. Criação da pasta de estudo para a atividade A1

```
mkdir estudo-python
```

3. Criação da pasta ./estudo-python/src

Usando o vscodium GUI

4. Instalação das extensões

Nome da extensão	Funcionalidade
ms-python	Extensão para utilizar o python no vscodium
pylint	Extensão para boas práticas da linguagem
autopep8	Extensão de condutas e normas da linguagem python

5. Upload do projeto para o github

Aula 02 - Keywords e Identificadores

Keywords

São palavras reservadas apenas utilizadas pelo python e não podem ser utilizadas pelo usuário para a declaração de variáveis e outras funcionalidades.

```
# Algumas palavras reservadas  
true, false, none, import, if, lambda
```

Identificadores

São as palavras de nome de variáveis, funções, classes. Eles são case sensitive, ou seja, existe diferença entre letras maiúsculas ou minúsculas.

Todos os identificadores podem começar com underline `_` ou uma letra, não podem ter espaços em branco entre um identificador composto. Por exemplo:

```
# nome de variável
# no caso o certo seria nome_de_variavel
nome = 'Maria'
idade = 30
Nome = 'João'
nome_completo = 'Maria da Silva'
```

Não se pode usar caracteres especiais nos nomes (identificadores) tais como `#`, `!`, `@`, `$`.

Uma boa prática da linguagem é utilizar sempre identificadores claros e que tenham significado para o que a variável está fazendo.

Na definição de constantes (ou seja, variáveis que não mudam), o correto é usar letras maiúsculas:

```
PI = 3.14

idade = 18
MAIORIDADE = 18

if idade >= MAIORIDADE:
    print('Maior de idade')
else:
    print('Menor de idade')
```

Aula 03 - Comentários

Podem ser apenas de uma única linha:

```
# comentário com uma única linha
```

Ou comentários com várias linhas:

```
'''
Comentários com várias linhas
comentários com várias linhas
comentários com várias linhas
'''
```

O comentário deve ser usado sempre que o código não estiver autoexplicativo.

Aula 04 - Variáveis, Constantes e Literais

A variável é um local de armazenamento de dados ou local que indica onde os dados estão sendo guardados, funcionando como um indicador (ponteiro).

Em algumas linguagens como C, é necessário falar qual o tipo da variável. No Python não é necessário pois ele faz a inferência do tipo automaticamente.

```
numero = 10
print(numero, type(numero))
# retorna: 10 <class 'int'>
```

O Python também aceita tipagem dinâmica e também pode alterar valores já dentro de uma variável:

```
valor_inicial = 10
valor_inicial = 20
print(valor_inicial)
# vai imprimir o valor de 20
```

O Python também consegue identificar múltiplas atribuições:

```
maria, idade, endereco = "maria", 20, "Rua 10 numero 1"
print(maria)
# maria
print(idade)
# 20
print(endereco)
# Rua 10 numero 1
```

Atribuição do mesmo valor para múltiplas variáveis:

```
nome1 = nome2 = nome3 = "João"
# todas as variáveis terão "João" como seu valor armazenado
```

Notação de variáveis compostas usando o snake_case

```
idfuncionario = 1234
id_funcionario = 1234 # uso do snake_case que é o padrão da pep8
```

As constantes em Python sempre são maiúsculas (UPPER_CASE).

Literais

Os literais em Python são os dados sem necessidade de ter uma variável junto dele:

```
# literais
print(27) # 27 é o literal
IDADE = 27
print(IDADE, type(IDADE)) # type mostra o tipo da variável ou do literal
```

Outros tipos em Python

```
# Strings
NOME = "Maria da Silva"
print(NOME, type(NOME))

# Booleano
print(True, type(True))
print(False, type(False))

# None
print(None, type(None))
```

Coleções

```
# Lista
numeros = [1, 2, 3]
print(numeros, type(numeros))

# Tupla (tuple)
emails = ('joao@mail.com', 'maria@mail.com')
print(emails, type(emails))

# Conjunto (set)
nomes = {'maria', 'maria', 'joao', 'pedro', 'maria', 'Pedro'}
print(nomes, type(nomes))

# Dicionário (dictionary)
aluno = {
    'prontuario': 1234,
    'nome': 'Maria da Silva',
    'idade': 34
}
print(aluno, type(aluno))
```

Aula 05 - Tipos de Dados

Existem diferentes tipos de dados em Python, alguns estudados são:

Numéricos (int, float, complexos)

```
IDADE = 20
PESO = 20.10
print(IDADE, type(IDADE))
print(PESO, type(PESO))
```

String

```
NOME = 'JOAO'
SOBRENOME = 'SILVA'
NOME_COMPLETO = NOME + ' ' + SOBRENOME
print(NOME_COMPLETO)

# Interpolação de variáveis
PRODUTO = 'Coca-Cola'
print(f'{NOME} {SOBRENOME} comprou {PRODUTO}')

# Métodos em strings: upper(), lower()
FRASE = 'TUDO ESTÁ ESCRITO EM MAIÚSCULA'
print(f'{FRASE.lower()}')
print(f'{FRASE.upper()}')
```

Dentro da função print podemos usar separadores especiais através da palavra `sep`:

```
print(1, 2, 3, 4, 5, sep=' / ')
```

Boolean

```
LIGADO = True
print(LIGADO, type(LIGADO))

RESULTADO = 10 == 10
print(RESULTADO, type(RESULTADO))
```

Listas

```
frutas = ['banana', 'manga', 'pera', 'maçã', 'laranja', 'abacaxi', 'limão']

print(f'{frutas}, tamanho da lista {len(frutas)}')

# adiciona elemento na lista
frutas.append('abacate')

# iteração sobre itens da lista
for fruta in frutas:
    print(fruta.upper())
```

Tupla

```
codigos = ('spa1', 'spb1', 'spb3')
print(codigos[0]) # imprime o código que estiver na posição inicial da tupla
print(len(codigos)) # imprime o valor de elementos na tupla
```

Set (Conjunto)

```
resultado_sorteio = {4, 5, 6, 7, 10}
print(resultado_sorteio)
resultado_sorteio.add(23)
print(resultado_sorteio)
```

Dicionários (dictionary)

```
funcionario = {
    'codigo': 123,
    'nome': 'José da Silva',
    'salario': 7000.00
}

print(funcionario)
print(funcionario['codigo'])
print(funcionario['nome'])
print(funcionario['salario'])

print(funcionario.keys()) # mostra a lista de chaves
print(funcionario.values()) # mostra a lista de valores

# Alterando valores do dicionário
funcionario['salario'] = 9000.00
print(funcionario)
```

Aula 06 - Conversão de Tipos

Existem dois tipos de conversão: a implícita e a explícita.

```
# Conversão de tipo implícito
LEITURA = 5.53
PESO = 3
TOTAL = LEITURA * PESO
print(TOTAL, type(TOTAL)) # fez a conversão implícita

# Faz a conversão explícita (TYPE CASTING)
TOTAL = int(TOTAL)
print(TOTAL, type(TOTAL))

# Conversão para string
TEXTO = 'Maria tem ' + str(TOTAL) + ' unidades'
print(TEXTO, type(TEXTO))
```

Aula 07 - Entradas e Saídas

A entrada e saída de dados em Python trata-se do input e output de dados I/O.

Pode-se alterar o padrão da saída do print usando `sep` e `end`:

```
print('hello world', 'Maria', True, 1, sep="@ ", end=' !!!\n')
print('hello world', sep=' ### ', end='\n\n\n')
```

Para salvar o resultado do print dentro de um arquivo:

```
arquivo = open('saida.txt', 'w', encoding='utf-8')
print("Saida do arquivo para o arquivo saida.txt", file=arquivo)
arquivo.close()
```

Entrada de dados

Na entrada de dados em Python usamos a palavra reservada `input`:

```
nome = input("Digite o seu nome -> ")
print(nome.upper())

idade = input("Digite a sua idade -> ")
print(idade, type(idade))

idade = int(input("Digite a sua idade -> "))
if idade >= 18:
    print(f"{nome} é maior de idade!!")
else:
    print(f"{nome} é menor de idade!!")
```

Leitura de entrada a partir de um arquivo

```
arq = 'notas.txt'
escrever_arquivo = open(arq, 'w', encoding='utf-8')

nota1 = input("entre a nota1 do aluno de zero a dez -> ")
nota2 = input("entre a nota2 do aluno de zero a dez -> ")
nota3 = input("entre a nota3 do aluno de zero a dez -> ")

print(nota1, nota2, nota3, sep=";", file=escrever_arquivo)

# Deve-se fechar o arquivo
escrever_arquivo.close()

arquivo_notas = open(arq, 'r', encoding='utf-8')
conteudo = arquivo_notas.read()
print(conteudo)
print(conteudo.split(sep=';'))

notas = conteudo.split(sep=';')
media = (float(notas[0]) + float(notas[1]) + float(notas[2])) / len(notas)
print(media)

# Fechar o arquivo no final
arquivo_notas.close()
```