Aula de revisão

Variáveis

Declarando e nomeando variáveis

- Use nomes que revelem a intenção da variável
- Evite desinformação
- Use nomes pronunciáveis
- Use nomes pesquisáveis
- Não use trocadilhos

Declarando e nomeando variáveis

Podem conter apenas letras, números e sublinhado/underscore

- Espaços não são permitidos
- Evite palavras-chave Python
- Curto e descritivo
- Cuidado com l e O, parecem 1 e 0

idp

Tipos das variáveis

Tipos de variáveis

Para cada tipo de variável, as funções tem comportamento diferente.

- Texto (str)
- Número inteiro (int)
- Número real (float)
- Booleano (bool), True ou False

Tipos de variáveis

Ao declarar uma variável, o Python determina seu tipo dinamicamente, uma variável pode mudar de tipo apenas declarando-a novamente:

```
idade = “Trinta e dois” # tipo texto
idade = 32 # tipo inteiro
```

Listas

Declarando listas

Estrutura de dados indexável

O índice começa em 0

Declarando uma lista vazia:

```
minhaLista = []
```

Declarando listas

Uma lista pode ter vários argumentos de diferentes tipos.

```
minhaLista = [‘eu’, ‘amo’, ‘Python’]
minhaLista[0]
```

Declarando listas

Modificando elementos de uma lista

```
minhaLista = [‘eu’, ‘amo’, ‘Python’]
minhaLista[2] = ‘software livre’
```

idp

Dicionários

Dicionários

Estrutura de dados relacional

Acessa valores através de palavras-chave

Declarando um dicionário vazio:

```
meuDict = {}
```

Dicionários

Declarando um dicionário com duas entradas e acessando cada uma delas:

```
meuDict = {‘linguagem’: ’Python’, ’idade’: ‘28 anos’}
meuDict[‘linguagem’]
meuDict[‘idade’]
```

Funções e bibliotecas

Acessando funções de variáveis

É possível visualizar as funções disponíveis para cada tipo de variável utilizando o operador .

- Exemplo
- nomes = []
- nomes. # Aperte Tab depois do .

Controle de fluxo

Laços

Em português, chamamos “loop” de “laço”

- Palavra-chave **for**

```
for cliente in clientes:
    print(cliente)
    if cliente == “Álvaro”:
        print(“Te peguei!”)
```

Laços for ou “para cada”

Com **for**, o bloco irá executar para cada elemento do objeto.

```
for cliente in clientes:
    print(cliente)
```

idp

Laços while ou “enquanto”

O **while** executará o bloco “enquanto” a condição a direita dele resultar em verdadeiro, ou booleano True.

```
while nome == “Álvaro”:
    print(“Olá, professor!”)
```

idp

Blocos if... else ou “se... senão”

O **if** ou “se”, executa o bloco se a condição for satisfeita.

```
if nome == “Álvaro”:
    print(“Bom dia, professor”)
else:
    print(“Bom dia, querido colega”)
```

idp

INSTITUTO BRASILEIRO DE ENSINO,
DESENVOLVIMENTO E PESQUISA