Introdução à linguagem Python

Fernando Henrique Vieira Trindade

Python - Comentários

```
# Comentário de uma linha
"""
Comentário
de várias linhas
"""
```

Relembrando os tipos

- Tipagem dinâmica
- Principais tipos:
  - **int, float, str, boll**

```
idade = 20
altura = 1.75
nome = "Ana"
ativo = True
```

Conversão de Tipos

```
int("10")          # string → int
float("3.14")      # string → float
str(100)           # int → string
bool(0)            # int → boolean
```

Entrada de dados

```
nome = input("Digite seu nome: ")
idade = int(input("Digite sua idade: "))
print(f"Olá {nome}, você tem {idade} anos.")
```

Revisando operadores

- **Aritméticos:** `+`, `-`, `*`, `/`, `//`, `%`, `**`
- **Comparação:** `==`, `!=`, `<`, `>`, `<=`, `>=`
- **Lógicos:** `and`, `or`, `not`
- **Pertencimento:** `in`, `not in`

Controle de Fluxo

```
if idade >= 18:
    print("Maior de idade")
elif idade > 12:
    print("Adolescente")
else:
    print("Criança")
```

Estruturas de Repetição - while

```
contador = 1
while contador <= 5:
    print(contador)
    contador += 1
```

Estruturas de Repetição - for

- range(inicio, fim, passo)
- Pode iterar listas, strings e outros iteráveis

```
for i in range(1, 6):
    print(i)
```

Controle de loop

- **break -** interrompe
- **continue -** pula para a próxima iteração

```
for i in range(1, 10):
    if i == 5:
        break
    if i % 2 == 0:
        continue
    print(i)
```

Controle de loop

- **break -** interrompe
- **continue -** pula para a próxima iteração

```
for i in range(1, 10):
    if i == 5:
        break
    if i % 2 == 0:
        continue
    print(i)
```

Listas

- **Mutáveis**
- **Funções úteis: -** append, remove, pop, sort, reverse, count, index

```
frutas = ["maçã", "banana", "uva"]
frutas.append("laranja")
frutas.remove("banana")
print(frutas[0])
```

Listas

- **Pode conter valores repetidos**
- **Pode conter tipos mistos (não recomendado para dados estruturados)**

```
vazia = []
numeros = [1, 2, 3, 4]
mistura = ["texto", 10, True, 3.14]
```

Listas - Acessando elementos

- Índice começa em 0
- Índices negativos contam de trás para frente

```
print(frutas[0])     # Primeiro elemento
print(frutas[-1])    # Último elemento
```

Listas - Alterando elementos

- **Basta atribuir um valor ao índice desejado**

```
frutas[1] = "laranja"
print(frutas)  # ["maçã", "laranja", "uva"]
```

Listas - Inserindo elementos

- **append() -** insere no final
- **insert(posição, valor) -** insere na posição específica

```
frutas.append("abacaxi")          # Adiciona no final
frutas.insert(1, "pera")          # Adiciona na posição 1
```

Listas - Removendo elementos

- **remove() -** apaga a primeira ocorrência do valor
- **pop() -** apaga pelo índice e retorna o item removido
- **clear() -** lista vazia()

```
frutas.remove("uva")        # Remove pelo valor
item = frutas.pop(0)        # Remove pelo índice e retorna
frutas.clear()              # Remove todos os elementos
```

Listas - Removendo elementos

- **remove() -** apaga a primeira ocorrência do valor
- **pop() -** apaga pelo índice e retorna o item removido
- **clear() -** lista vazia()

```
frutas.remove("uva")        # Remove pelo valor
item = frutas.pop(0)        # Remove pelo índice e retorna
frutas.clear()              # Remove todos os elementos
```

Listas - Fatiando listas

- **[início:fim:passo]**
- **Não inclui o índice final**

```
numeros = [10, 20, 30, 40, 50]
print(numeros[1:4])    # [20, 30, 40]
print(numeros[:3])     # [10, 20, 30]
print(numeros[2:])     # [30, 40, 50]
print(numeros[::2])    # [10, 30, 50]
```

Listas - Copiando listas

- **.copy() ou [:]**
- **Atribuição direta (lista2 = lista1) mantém referência**

```
lista1 = [1, 2, 3]
lista2 = lista1.copy()
lista3 = lista1[:]      # Também funciona
```

Listas - Iterando listas

- **for** valor **in** lista - percorre valores
- **enumerate(lista) -** índice e valor

```
for fruta in frutas:
    print(fruta)

for i, fruta in enumerate(frutas):
    print(i, fruta)
```

Listas - Ordenando listas

- **.sort()** - altera a lista
- **sorted(lista)** - retorna nova lista ordenada

```
numeros.sort()                  # Ordem crescente
numeros.sort(reverse=True)      # Ordem decrescente
```

Listas - Funções úteis

```
len(frutas)                # Quantidade de elementos
frutas.count("maçã")       # Quantas vezes aparece
frutas.index("uva")        # Índice da primeira ocorrência
```

Tuplas

- **Imutáveis**
- **Mais rápidas e mais seguras para dados fixos.**

```
cores = ("vermelho", "azul", "verde")
print(cores[1])
```

Criando Tuplas

```
vazia = ()
numeros = (1, 2, 3)
mistura = ("texto", 10, True)
um_elemento = (5,)  # vírgula obrigatória
```

Acessando elementos das Tuplas

```
print(cores[0])     # vermelho
print(cores[-1])    # verde
```

Iterando em Tuplas

```
for cor in cores:
    print(cor)
```

Convertendo Tuplas - Listas

```
lista = list(cores)      # tupla → lista
lista[0] = "amarelo"
cores = tuple(lista)     # lista → tupla
```

Tuplas - Funções úteis

```
len(cores)                  # tamanho
cores.count("azul")         # ocorrências
cores.index("verde")        # índice
```

Manipulando Strings

```
texto = "  Python é incrível!  "
print(texto.lower())
print(texto.upper())
print(texto.strip())
print(texto.replace("Python", "Java"))
print(texto.split())
```

Dicionários

- **Chave:Valor**
- **Chaves únicas (strings, números, tuplas imutáveis)**
- **Funções úteis:** keys(), values(), items(), get()

```
pessoa = {"nome": "Ana", "idade": 25}
pessoa["cidade"] = "São Paulo"
print(pessoa["nome"])
```

Exercícios

1. Faça os exercícios da lista_01 no classroom;
2. Crie um arquivo .py para cada exercício. Coloque o enunciado da questão como comentário e resolva com o código logo abaixo do comentário;
3. Empacote todos os exercícios em uma pasta .Zip e envie para o email: fernando.trindade@souunit.com.br OBS: Lembre de colocar seu nome completo e o nome da turma no assunto do email.