@PETTELEUFF

Mini Curso Python

Aula 02

SeTel 2023
Universidade Federal Fluminense

AULA 02

Tópicos

7 Manipulação de String

8 Vetores

9 Listas

10 Matrizes

11 Funções

PET Tele

Strings

- Representam formação textual;

```
nome = ‘Silvana’
curso = ‘Engenharia de Telecomunicação’
faculdade = ‘Universidade Federal Fluminense’
```

7 Manipulação de String

Acesso a conteúdo das strings

- Acesso é feito pela variável que contem a string
- Caracteres podem ser acessados pela posição dentro da string;
- A primeira posição tem indice zero;

```
nome= ‘Pet Tele’
print(nome[0]) -> P
print(nome[4]) -> T
```

0	1	2	3	4	5	6	7
P	e	t		T	e	l	e

7 Manipulação de String

Acesso a conteúdo das strings

- Substring
  - [inicio:fim]
- substring retornanda vai de inicio (inclusive) até fim -1;
- Se início for omitido, significa 0
- se fim for omitido, significada len(string)

0	1	2	3	4	5	6	7
P	e	t		T	e	l	e

7 Manipulação de String

Alteração

- conteudo da posição de string é inalteravel (são sequencias imutaveis)

```
nome= ‘Pet Tele’
nome[2]=’o’
```

```
Traceback (most recent call last):
  File "/home/runner/EXEMPLOS/main.py", line 2, in <module>
    nome[2]='o'
TypeError: 'str' object does not support item assignment
```

Operadores

- in
  - `substring in string`
    - retorna True ou False

```
nome= ‘Pet Tele’
print(‘P’ in nome) -> True
print(‘p’ in nome) -> False
print(‘a’ in nome) -> False
print(‘Tele’ in nome) -> True
```

- len
  - **len**(string)
    - retorna a quantidade de caracteres da string

```
nome= ‘Pet Tele’
print(len(nome) -> 7
```

7 Manipulação de String

Operadores

- + (concatenação)
  - `string1 + string2`
    - concatena duas strings
- * (repetição)
  - `string*int`
    - repete a string int vezes

```
nome= ‘Pet Tele’
print(‘P’ in nome) -> True
print(‘p’ in nome) -> False
print(‘a’ in nome) -> False
print(‘Tele’ in nome) -> True
```


```
nome= ‘Pet Tele’
print(‘P’ in nome) -> True
print(‘p’ in nome) -> False
print(‘a’ in nome) -> False
print(‘Tele’ in nome) -> True
```


PET Tele

7 Manipulação de String

Operadores

- in
  - `substring in string`
    - retorna True ou False
- len
  - `len(string)`
    - retorna a quantidade de caracteres da string
- + (concatenação)
  - `string1 + string2`
    - concatena duas strings
- * (repetição)
  - `string*int`
    - repete a string int vezes

0	1	2	3	4	5	6	7
P	e	t		T	e	l	e

PET Tele

Percorrendo uma string

- Os elementos de uma string podem ser acessados usando uma estrutura de repetição

7 Manipulação de String

F-String

- Tipo especial de String que facilita a incorporação de variáveis;
- É criada com f’...’
- Permite acessar expressões usando {...}
- Exemplo:

7 Manipulação de String

F-String

```
7  print ('Meu nome é', nome ,'tenho',
   idade,'anos,','tenho', altura,'m,','peso', peso,
   'kg,','sou', nacionalidade) #saida de dados
```

```
#Faça um programa que leia o nome, a idade, aaltura, o peso e a nacionalidade do
usuário e escreva essas informações na forma de um parágrafo de apresentação

nome = input('digite seu nome: ')
idade = (int(input('digite sua idade: ')))
altura = (float(input('digite sua altura: ')))
peso = (float(input('digite seu peso: ')))
nacionalidade = input('digite sua nacionalidade: ')
print (f'Meu nome é {nome}, tenho {idade} anos, tenho {altura} m de altura, peso
{peso} kg e sou {nacionalidade}')
```

PET Tele

7 Manipulação de String

F-String

```
palavra= ‘PetTele’
texto= f’A palavra{palavra} tem {len(palavra)} letras!’
print(texto)

Saída:
A palavra PetTele tem 7 letras!
```

7 Manipulação de String

F-String

```
numerador = 5
denominador = 3
divisao = numerador / denominador
print(f'{numerador} / {denominador} = {divisao}!')
```

→ 5 / 3 = 1.6666666666666667!

7 Manipulação de String

F-String

```
divisao = 5 / 3
print(f'{divisao}')      → 1.6666666666666667
print(f'{divisao:.5f}') → 1.66667
print(f'{divisao:.2f}') → 1.67
```

7 Manipulação de String

Operações sobre Strings

- upper
- lower
- find
- strip

7 Manipulação de String

Operações sobre Strings

upper

- `string.upper()`
  - retorna string com letras minúsculas substituídas por maiúsculas;

```
texto = ‘Amo as aulas de Python’
print(texto.upper())

Saída:
‘AMO AS AULAS DE PYTHON’
```

7 Manipulação de String

Operações sobre Strings

lower

- `string.lower()`
  - retorna string com letras maiúsculas substituídas por minúsculas;

```
texto = ‘Amo as aulas de Python’
print(texto.lower())

Saída:
‘amo as aulas de python’
```

Operações sobre Strings

find

- `string.find(substring, inicio, fim)`
  - retorna o indíce da primeira ocorrência da **substring** dentro da **string**, a partir do início, até a posição **fim-1**;
  - retorna **-1** se a substring nao for encontrada;
  - **início** e **fim** são opcionais;
  - não é possivel informar **fim** sem informar **início**

7 Manipulação de String

Operações sobre Strings

find

```
texto = 'Amo as aulas de Python em que o Pet-Tele está
dando na SeTel no Instituto de computação'
print(texto.find('Tel')) -> 36
print(texto.find('Tel', 40, 70)) -> 57
print(texto.find('Tel', 40, 50)) -> -1
```

7 Manipulação de String

Operações sobre Strings

strip

- `string.strip()`
  - retorna uma string com espaço do início e do fim da strig original removidos
  - a string original não é modificada

```
texto = ‘   Pet   ’
print(texto.strip())

Saída:
Pet
```

8 Vetores

- É possível definir variáveis que guardam mais de um valor;
- Essas variáveis são conhecidas por diferentes nomes:
  - Arrays (arranjos)
  - Variáveis compostas
    - unidimensional
      - espaço para armazenar diversos valores;
      - é acessada via índice
  - Variáveis subescritas
    - ex: x1, x2,..., xn
  - Variáveis indexáveis
- Elas podem ser unidimensionais (vetores) ou multidimensionais (matrizes)

9 Listas

- Vetores são implementados por meio de listas
  - Colchetes delimitam início e fim da lista;
  - Vírgula delimita os elementos.
- Elementos podem ser de qualquer tipo
- Ex:
  - lista = [’a’, 1, 2.5, ‘Pet’]
  - notas=[10, 20, 7.8, 9]

Utilização de listas

- comando for
  - comando len informa o tamanho da lista
  - parecido com string

```
notas = [8.0, 5.5, 1.5]
for i in range(len(notas)):
    print(notas[i])
```

```
8.0
5.5
1.5
```

9 Listas

- count
- insert
- index
- del
- pop
- remove

9 Listas

```
notas[0] = float(input('Digite a nota do primeiro aluno: '))
notas[1] = float(input('Digite a nota do segundo aluno: '))
notas[2] = float(input('Digite a nota do terceiro aluno: '))
```

9 Listas

```
notas = []
notas[0] = float(input('Digite a nota do primeiro aluno: '))
notas[1] = float(input('Digite a nota do segundo aluno: '))
notas[2] = float(input('Digite a nota do terceiro aluno: '))
```

9 Listas

```
notas = []
notas.append(float(input('Digite a nota do primeiro aluno: ')))
notas.append(float(input('Digite a nota do segundo aluno: ')))
notas.append(float(input('Digite a nota do terceiro aluno: ')))
```

9 Listas

```
notas = []
print(notas)      → []
notas.append(8)
print(notas)      → [8]
notas.append(9)
print(notas)      → [8, 9]
notas.append(6)
print(notas)      → [8, 9, 6]
notas[1] = 10
print(notas)      → [8, 10, 6]
```

9 Listas

```
#Programa q leia notas dos alunos, faça a media das notas e parabenize que esta acima da media
num_alunos = (int(input('Digite a quantidade de alunos: ')))
nomes = []
notas = []
media = 0
for i in range(num_alunos):
    nomes.append(input(f'Informe o nome do aluno {i+1}: '))
    notas.append(float(input(f'Informe a nota de {nomes[i]}: ')))

media = media + notas[i]
media = media / num_alunos
print(f'A média da turma é {media:.2f}.')

for i in range(num_alunos):
    if notas[i] > media:
        print(f'Parabéns {nomes[i]}!')
```

10 Matrizes

- Variável composta multidimensional
  - É equivalente a um vetor, contudo permite a utilização de diversas dimensões acessadas via diferentes índices
  - Pode ser pensada como um vetor onde cada célula é outro vetor
  - Em diversas situações matrizes são necessárias para correlacionar informações

10 Matrizes

Acesso de valores: [linha][coluna]

print(notas[1][3])