

Fundamentos de Programação de Computadores

Python- Strings

Docente: Fátima Leal

DCT DEPARTAMENTO CIÊNCIA
E TECNOLOGIA

Python

- Na última aulas vimos:
- **Funções**
- **ciclos**
- **condições**

Python - Strings

- **Strings** são sequências de caracteres
- No Python, identificamos como strings tudo o que estiver entre “” ou ‘’
- Utilizadas para representar frases ou palavras

```
string1 = "Bem-vindos"
```

```
string2 = string1 - 1
```

```
string3 = "Bem-vindos" / 123
```

```
string4 = string1 * "Bem-vindos"
```

```
string5 = "15" + 2
```

Quais dos comandos que envolvem strings estão corretos?

Python - Strings

- A maior parte dos operadores aritméticos não funciona com strings
- Exceções: **concatenação** de strings (+), **repetição** (*)
- **Precedência**: tal como com **operações numéricas**.
- **Strings** são tipos de **dados compostos**. Ao contrário do **int**, **float** ou **bool** que são tipos de **dados simples ou primitivos**.
- Às strings chamamos também **coleções de dados**.

G	E	E	K	S	F	O	R	G	E	E	K	S
0	1	2	3	4	5	6	7	8	9	10	11	12

Python – Strings concatenação e repetição

- A concatenação de strings é realizada através do comando +

```
a = "Fundamentos"
b = "de"
c = "Programacao"
espaco = " "

print (a + espaco + b + espaco + c)
```

- A repetição de uma string é realizada através do comando *

```
>>> string1 = "Ola"
>>> string1*3
'OlaOlaOla'
>>> |
```

- Como vimos a entrada e saída de dados deve ser feita usando strings como tipos de dados.

Python – Strings indexação

- **Operador de indexação:** parêntesis retos ([])
- **Valor entre parêntesis:** índice
- **Primeiro carater:** índice 0
- **Último carater:** índice -1 (ou n-1)

```
>>> disciplina = "Fundamentos de PRG"  
>>> letra = disciplina[2]  
>>> print(letra)
```

- Qual o resultado do print?

Python – Strings indexação

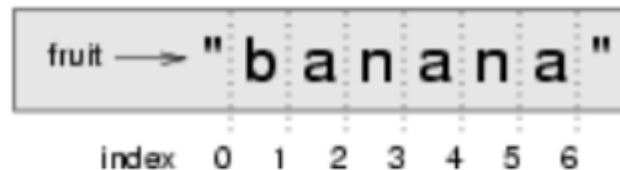
- Podemos obter o tamanho de uma string utilizando a função `len(string)` do Python
- Podemos ter acesso a índices a contar do fim utilizando índices negativos

```
>>> disciplina[-2]  
'R'
```

- Acesso a índices maiores que o tamanho da string origina um erro
- Não se pode alterar valores de caracteres de strings

Python – Fatias de Strings

- Em python podemos extrair partes da string utilizando os índices dos caracteres



```
>>> fruit = "banana"
>>> fruit[:3]
'ban'
>>> fruit[3:]
'ana'
>>> fruit[2:5]
'nan'
```


Python – Substrings

```
>>> phrase = "Pirates of the Caribbean"
>>> print(phrase[0:5])
Pirat
```

- `txt[i:j]` substring entre índices `i` e `j-1` inclusive
- `txt[i:]` substring desde o índice `i` até ao final
- `txt[:j]` substring desde o início até ao índice `j-1` inclusive

- Quando o valor de `j` for superior ao tamanho da string, no acesso a uma fatia isso não dá erro (resulta na string desde `i` até ao final)

Python – testar ocorrências em Strings

`txt1 in txt2` → testa ocorrência de txt1 dentro de txt2

```
>>> 'ana' in 'Banana'
True
>>> 'mana' in 'Banana'
False
>>> 'mana' not in 'Banana'
True
>>> 'Banana' in 'Banana'
True
>>> '' in 'Banana'
True
```

Python – Percorrer strings

```
word="Banana"  
for letter in word:  
    print(letter)
```

Se as strings são espécie de vetores,
podemos percorrê-los

Exercícios

Qual o tamanho da string?

Quantas vezes aparece o carater a?

Quantas vezes aparece o carater a e n?

Substitui o carater a por @ da palavra Banana.

Implementa uma função que verifica quantas vezes um carater introduzido pelo utilizador aparece na palavra “Banana”.

Python – tamanho strings

```
word="Banana"  
for letter in word:  
    print(letter)
```

len(string) – indica o tamanho da string

Python

- Depois de uma aplicação implementada funcionar:
 - Devemos simplificar o código
 - Melhorar a sua leitura
 - Fazer comentários usando o simbolo **#** que nos façam recordar as ideias implementadas

