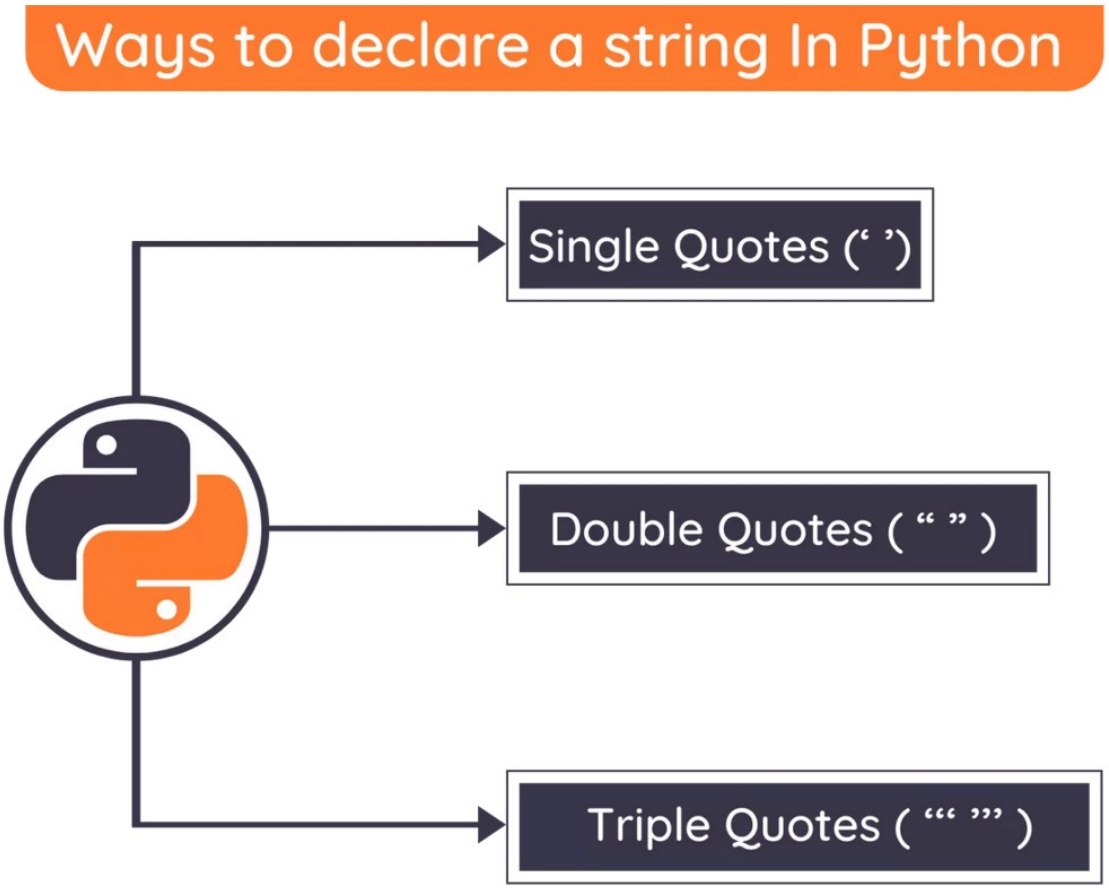
Python – Strings e Docstrings



https://www.fireblazeaischool.in/blogs/python-strings/

Python – Strings e Docstrings

String - tipo de dados que representa uma sequência de caracteres

```
string1 = 'ola mundo'              # single quotes

string2 = "ola' mundo"             # double quotes

string3 = '''ola' "mundo"'''       # triple quotes

print(string1)
print(string2)
print(string3)
```

```
ola mundo
ola' mundo
ola' "mundo"
```

Python – Strings e Docstrings

- O uso de strings em Python funciona de forma idêntica às listas;
- O acesso aos vários caracteres que constituem a string é feito através de um índice;
- Nas strings, tal como nas listas, a primeira posição corresponde ao índice 0:
- Podem-se usar intervalos de índices e definir um *passo*;
- Podem-se usar índices negativos (-1 corresponde à última posição da string).

```
string1 = 'ola mundo'

print( string1[1:8:3] )

print( string1[-1:-10:-1] )
```

```
lmd
odnum alo
```

Python – Strings e Docstrings

Outros exemplos:

```
string1 = 'ola mundo'

for c in range(4, 9):
    print( string1[c] )
```

```
m
u
n
d
o
```

```
for c in string1:
    print( c, end=' ' )
```

```
o l a   m u n d o
```

```
for c in string1:
    if c in 'aeiou':
        print( c, end=',' )
```

```
o,a,u,o,
```

Python – Strings e Docstrings

Métodos (funções) pré-definidos para operações com strings:

- capitalize()
- casefold()
- center()
- count()
- find()
- endswith()
- expandtabs()
- index()
- isalnum
- isalpha()
- isdecimal()
- isdigit()
- isidentifier()
- islower()
- isnumeric()
- isprintable()
- isspace()
- istitle()
- isupper()
- join()
- lower()
- lstrip()
- replace()
- rfind()
- rindex()
- rjust()
- rsplit()
- rstrip()
- split()
- rsplit()
- rstrip()
- split()
- splitlines()
- startswith()
- swapcase()
- title()
- upper()
- zfill()

Ver lista completa e detalhes das funções em
https://www.w3schools.com/python/python_ref_string.asp

Python – Strings e Docstrings

Exemplo: criar um programa que use um subprograma para contar o número de ocorrências de um carácter numa string

```
def contaOcorrencias( string, c ):
    contador = 0
    for a in string:
        if a == c:
            contador += 1
    return contador

def main():
    s = input("Insira uma sequencia de caracteres: " )
    c = input("Insira o carácter a considerar: " )
    o = contaOcorrencias( s, c )
    print( "Ocorrencias de", c, "na string", '''"''', s, '''"''', ": ", o)

if __name__ == '__main__':
    main()
```

```
Insira uma sequencia de caracteres: algoritmo
Insira o carácter a considerar: o
Ocorrencias de o na string " algoritmo " :  2
```

Python – Strings e Docstrings

Docstrings (documentation strings):

- São um tipo especial de comentários que, ao contrário dos outros comentários, podem ser usados pelo interpretador de Python;
- Servem, entre outros, para documentar uma função em Python, descrevendo o que a função faz e quais as características principais (parâmetros formais, valor de retorno);
- São definidas usando *triple single quotes* (‘’’) ou *triple double quotes* (“””) e colocadas logo abaixo da declaração da função (primeira linha);
- Podem ser acedidas usando o método `__doc__`, associado ao nome da função ou usando a função pré-definida `help` (por exemplo, a partir da *shell*);
- De uma forma geral, destinam-se a documentar o código-fonte de um programa em Python, existindo vários formatos diferentes para as escrever.

Python – Strings e Docstrings

Como deve ser escrita uma docstring?

- A primeira linha da docstring deve começar com uma letra maiúscula e terminar com um ponto final;
- A primeira linha deve ser uma descrição breve do que a função faz;
- Se houver mais linhas na docstring, a segunda linha deve estar em branco, separando visualmente o resumo do restante da descrição;
- As linhas seguintes devem conter um ou mais parágrafos que descrevam as convenções de chamada do objeto (nomes e tipos dos parâmetros formais), efeitos colaterais (valor de retorno), etc.

Python – Strings e Docstrings

Como deve ser escrita uma docstring?

```
def multiply_numbers(a, b):
    """
    Multiplies two numbers and returns the result.

    Args:
        a (int): The first number.
        b (int): The second number.

    Returns:
        int: The product of a and b.
    """
    return a * b
```

(não é obrigatório seguir exatamente este formato ...)

Python – Strings e Docstrings

Como deve ser escrita uma docstring?

```
def contaOcorrencias( string, c ):
    '''
        Retorna o numero de ocorrencias de um caracter numa string.

        (string, char) -> int
        >>> contaOcorrencias( "palavra", 'a' )
        3
        >>> contaOcorrencias( "palavra", 'e' )
        0
    '''
    contador = 0
    for a in string:
        if a == c:
            contador += 1
    return contador
```

(neste caso, a docstring apresenta dois exemplos de uso da função)

Python – Strings e Docstrings

```
def main():
    s = input("Insira uma sequencia de caracteres: " )
    c = input("Insira o carácter a considerar: " )

    print(contaOcorrencias.__doc__)

    o = contaOcorrencias( s, c )
    print( "Ocorrencias de", c, "na string", '''"''', s, '''"''', ": ", o)
```

```
Retorna o numero de ocorrencias de um caracter numa string.

(string, char) -> int
>>> contaOcorrencias( "palavra", 'a' )
3
>>> contaOcorrencias( "palavra", 'e' )
0

Ocorrencias de o na string " algoritmo " :  2
```

Python – Strings e Docstrings

Como usar uma docstring?

```
def main():
    s = input("Insira uma sequencia de caracteres: " )
    c = input("Insira o carácter a considerar: " )

    help(contaOcorrencias)

    o = contaOcorrencias( s, c )
    print( "Ocorrencias de", c, "na string", '''"''', s, '''"''', ": ", o)
```

```
Help on function contaOcorrencias in module __main__:

contaOcorrencias(string, c)
    Retorna o numero de ocorrencias de um caracter numa string.

    (string, char) -> int
    >>> contaOcorrencias( "palavra", 'a' )
    3
    >>> contaOcorrencias( "palavra", 'e' )
    0

Ocorrencias de o na string " algoritmo " :  2
```

Python – Strings e Docstrings

Como usar uma docstring?

```
>>> help(contaOcorrencias)
Help on function contaOcorrencias in module __main__:

contaOcorrencias(string, c)
    Retorna o numero de ocorrencias de um caracter numa string.

    (string, char) -> int
    >>> contaOcorrencias( "palavra", 'a' )
    3
    >>> contaOcorrencias( "palavra", 'e' )
    0
```

A função `help` pode ser chamada a partir da *IDLE Shell.*

Python – Strings e Docstrings

Como usar uma docstring?

```
>>> help(print)
Help on built-in function print in module builtins:

print(*args, sep=' ', end='\n', file=None, flush=False)
    Prints the values to a stream, or to sys.stdout by default.

    sep
      string inserted between values, default a space.
    end
      string appended after the last value, default a newline.
    file
      a file-like object (stream); defaults to the current sys.stdout.
    flush
      whether to forcibly flush the stream.
```

A função `help` pode também ser usada para obter informações sobre funções pré-definidas.

Python – Strings e Docstrings

Exemplo: criar um programa que use um subprograma para contar o número de ocorrências de um carácter numa string

```
def contaOcorrencias( string, c ):
    contador = 0
    for a in string:
        if a == c:
            contador += 1
    return contador

def main():
    print()
    s = input("Insira uma sequencia de caracteres: " )
    c = input("Insira o carácter a considerar: " )
    o = contaOcorrencias( s, c )
    print( "Ocorrencias de ", c, " na string ", '''"''', s, '''"''', ": ", o, sep='' )

if __name__ == '__main__':
    main()
```

(comparar com a saída obtida no slide 7)

```
Insira uma sequencia de caracteres: algoritmo
Insira o carácter a considerar: o
Ocorrencias de o na string "algoritmo": 2
```

Python – Strings e Docstrings

Exercícios

```
# Exercicio 1
# contar o numero de ocorrencias de um caracter numa string
# usar uma docstring
# executar na shell os comandos help() e print() para ver a
docstring

# Exercicio 2
# indicar a primeira posicao onde ocorre um caracter numa string

# Exercicio 3
# contar o  numero de espaços existente numa string

# Exercicio 4
# remover espacos existentes numa string

# Exercicio 5
# remover caracteres nao alfabeticos existentes numa string
```

Python – Strings e Docstrings

Exercícios

```
# Exercicio 6
# escrever uma string considerando apenas os caracteres de ordem par
(começa em 0)

# Exercicio 7
# substituir numa string um caracter por outro

# Exercicio 8
# escrever os codigos ascii dos caracteres existentes numa string
# usar funcao pre-definida – ord

# Exercicio 9
# contar o numero de vogais de uma string

# Exercicio 10
# verificar se uma string (palavra) e um palindromo
```