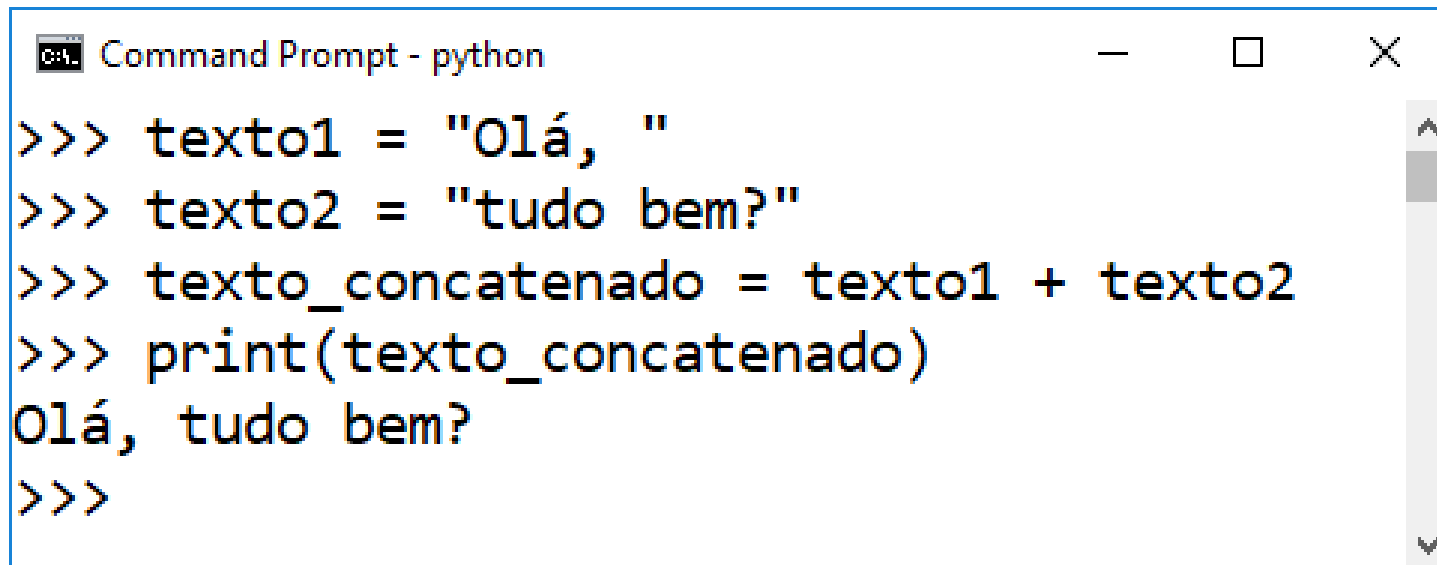


Variáveis e tipos de dados

Aprendendo mais sobre strings

Concatenação

Concatenar é unir strings. Como se estivéssemos “somando” os textos. Para concatenar usamos o operador de adição “+”.

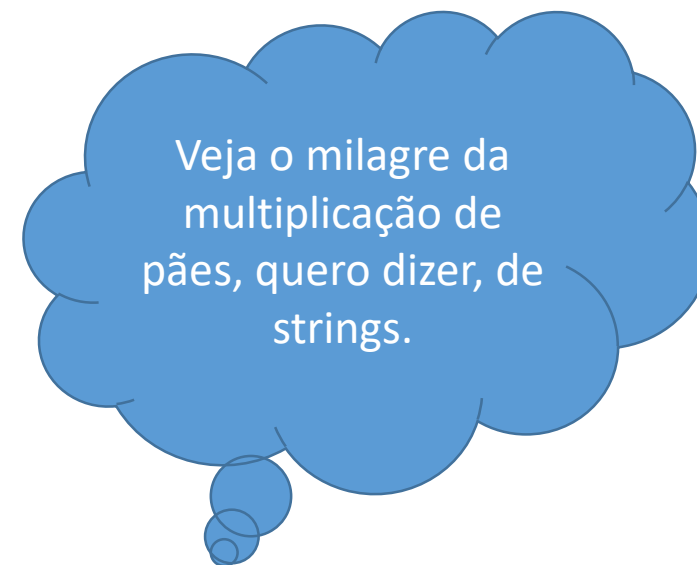
A screenshot of a Windows Command Prompt window titled "Command Prompt - python". The window contains the following Python code and its output:

```
>>> texto1 = "Olá, "  
>>> texto2 = "tudo bem?"  
>>> texto_concatenado = texto1 + texto2  
>>> print(texto_concatenado)  
Olá, tudo bem?  
>>>
```

Multiplicando strings

Você pode usar o operador de multiplicação “*” sobre uma string e informar quantas vezes ela vai se repetir.

```
Command Prompt - python
>>> alimento = "pão "
>>> multiplica = alimento * 10
>>> print(multiplica)
pão pão pão pão pão pão pão pão pão pão
>>>
```



Composição ou interpolação

Composição é a forma de juntar vários conteúdos em uma única string. Estes conteúdos podem ser outras strings ou até de outros tipos de dados.

A interpolação é mais eficiente no uso de memória que a concatenação convencional.

O operador “%”, denominado “marcador de posição”, é usado para definir o local onde será inserido o conteúdo na string.

Composição ou interpolação

Composição ou Interpolação

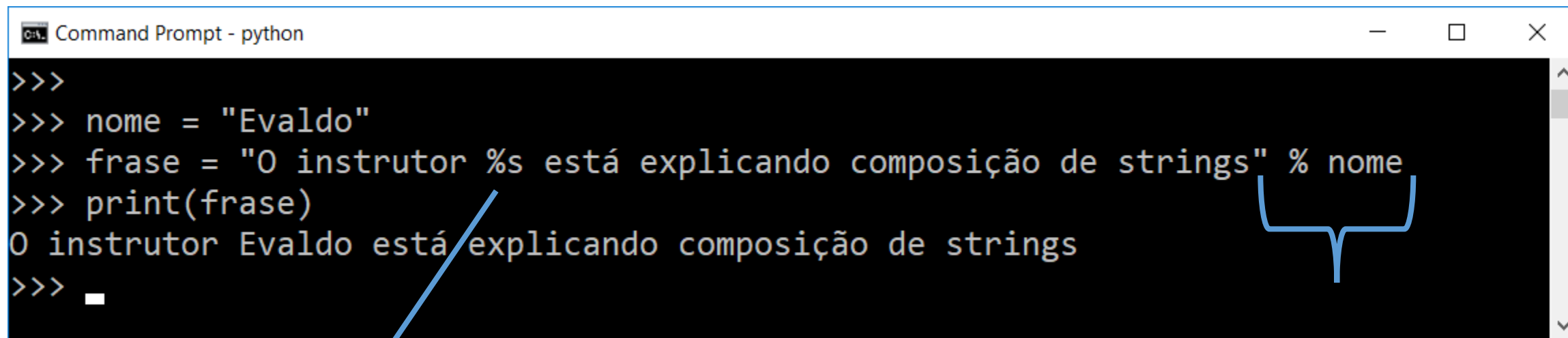
Veja nesta tabela alguns símbolos usados na composição.

%s	String
%d	Inteiro
%f	Real
%%	Sinal de porcentagem

Onde estes símbolos forem colocados dentro de uma string, serão substituídos pelos valores informados.

Composição ou interpolação

Usando %s para compor strings



```
Command Prompt - python
>>>
>>> nome = "Evaldo"
>>> frase = "0 instrutor %s está explicando composição de strings" % nome
>>> print(frase)
0 instrutor Evaldo está explicando composição de strings
>>> _
```

Quero colocar a string aqui.

Esta é a string que quero colocar no lugar do marcador (%s)

Composição ou interpolação

Usando %d para números inteiros

```
Command Prompt - python
>>> nome = "João da Silva"
>>> idade = 38
>>> filhos = 2
>>> frase = "O jogador %s tem %d anos de idade e tem %d filhos." % (nome, idade, filhos)
>>> print(frase)
O jogador João da Silva tem 38 anos de idade e tem 2 filhos.
>>>
```

Composição ou interpolação

Usando %d para números inteiros

```
Command Prompt - python
>>>
>>> print("Esta é a aula de número %03d" % aula)
Esta é a aula de número 009
>>> print("Esta é a aula de número %3d" % aula)
Esta é a aula de número 9
>>> print("Esta é a aula de número [%3d]" % aula)
Esta é a aula de número [ 9]
>>> print("Esta é a aula de número [%03d]" % aula)
Esta é a aula de número [009]
>>> print("Esta é a aula de número [%-3d]" % aula)
Esta é a aula de número [9 ]
>>> _
```

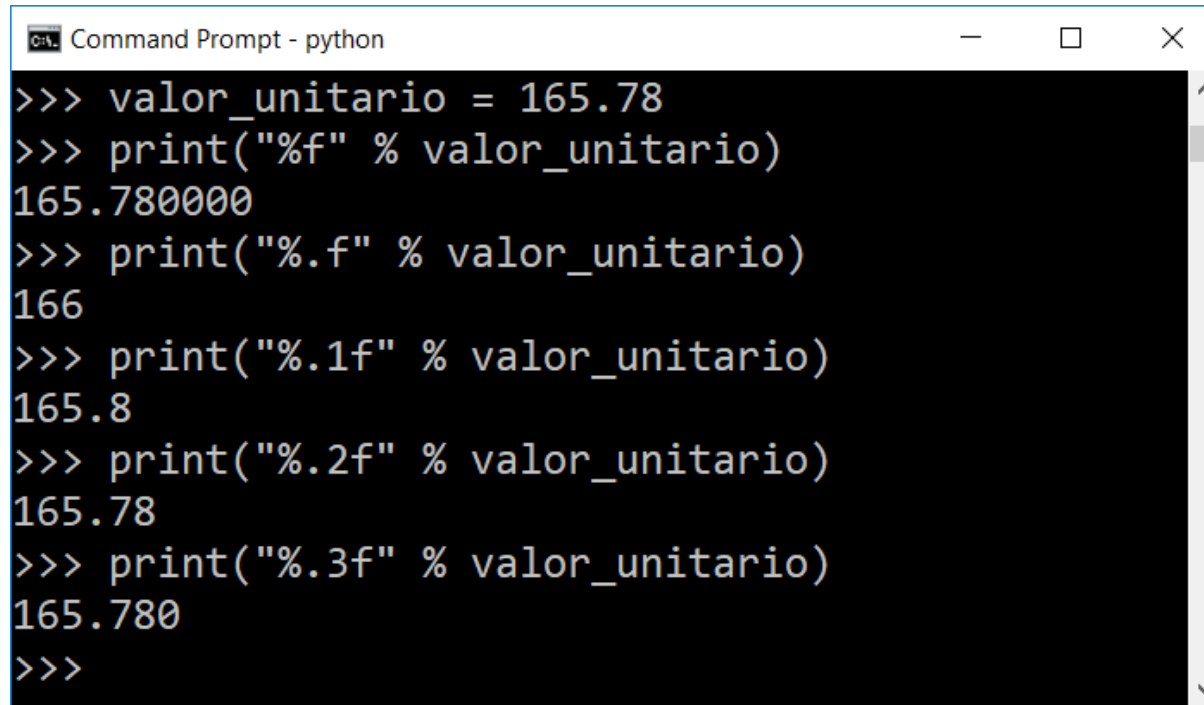

Composição ou interpolação

Usando %f para números reais

```
Command Prompt - python
>>>
>>>
>>> valor_unitario = 165.78
>>> quantidade = 5
>>> valor_total = valor_unitario * quantidade
>>> frase = "O produto custa %f. Eu comprei %d. Paguei ao todo %f" % (valor_unitario, quantidade, valor_total)
>>> print(frase)
O produto custa 165.780000. Eu comprei 5. Paguei ao todo 828.900000
>>> _
```

Composição ou interpolação

Arredondando números reais.



```
Command Prompt - python
>>> valor_unitario = 165.78
>>> print("%f" % valor_unitario)
165.780000
>>> print("%.f" % valor_unitario)
166
>>> print("%.1f" % valor_unitario)
165.8
>>> print("%.2f" % valor_unitario)
165.78
>>> print("%.3f" % valor_unitario)
165.780
>>>
```

Composição ou interpolação

Usando %% para exibir sinal de porcentagem

```
Command Prompt - python
>>>
>>>
>>> percent = 17.5
>>> frase = "A taxa de juros é de %.2f%%" % percent
>>> print(frase)
A taxa de juros é de 17.50%
>>> _
```

Composição ou interpolação

Usando *format* em vez do marcador de posição “%”

```
Command Prompt - python
>>>
>>> nome = "José da Silva"
>>> idade = 50
>>> filhos = 3
>>> frase = "{} tem {} anos de idade e tem {} filhos".format(nome, idade, filhos)
>>> print(frase)
José da Silva tem 50 anos de idade e tem 3 filhos
>>>
```

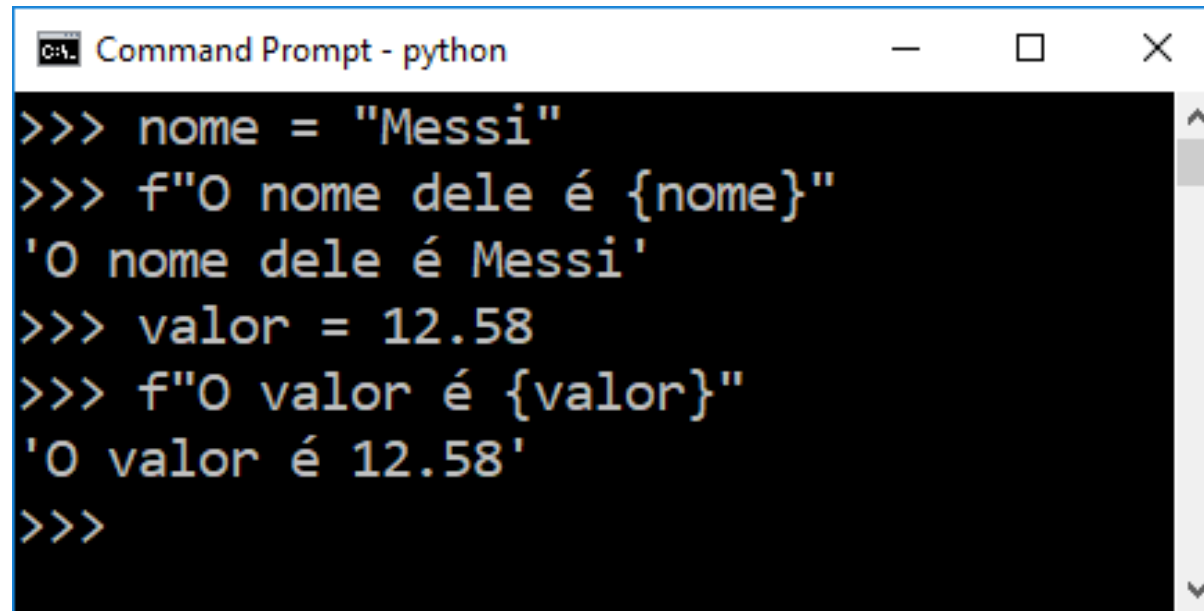
Composição ou interpolação

Outra forma de usar *format*

```
Command Prompt - python
>>> frase = "Olá, eu vou viajar de {3}".format('bicicleta', 'moto', 'trem', 'carro')
>>> print(frase)
Olá, eu vou viajar de carro
>>>
```

Composição ou interpolação

Pode ficar melhor? Sim, no Python 3.6 foi criado o recurso de f-strings ou Literais de strings formatadas.

A screenshot of a Windows Command Prompt window titled "Command Prompt - python". The window has a black background with white text. It shows a Python interactive session with the following code and output:

```
>>> nome = "Messi"
>>> f"O nome dele é {nome}"
'O nome dele é Messi'
>>> valor = 12.58
>>> f"O valor é {valor}"
'O valor é 12.58'
>>>
```

FIM