



Módulo 2.1 - String

5000



Temos um vídeo referente a este conteúdo, você pode acessá-lo [aqui](#).

Olá, desenvolvedores! O nosso conteúdo do dia englobará algumas das estruturas fundamentais do Python, as strings. Esperamos que vocês possam tirar o máximo possível de proveito do conteúdo que estamos disponibilizando, não esqueçam de praticar bastante para fixar bem.

Strings

Um objeto string em Python tem informações de caracteres e pode conter qualquer número de caracteres, inclusive zero. Muitas vezes você verá esse tipo de dado abreviado como *str*.

As strings são indicadas usando aspas simples, aspas duplas ou aspas triplas. As aspas simples são utilizadas para maioria dos casos. Se quisermos inserir aspas simples dentro da string, teremos duas alternativas, a primeira alternativa é usar o operador de escape \ antes das aspas, e a segunda alternativa é envolver sua string com aspas duplas.

O mesmo vale para strings que contêm aspas duplas, precisamos ou anular esse caractere com o operador de escape \, ou envolver a string com aspas simples. As aspas triplas são usadas para strings longas que podem se estender por várias linhas. O tipo de aspa usada no final da string deve corresponder ao tipo de aspa do começo. As aspas simples podem conter aspas duplas dentro delas, e vice-versa. Aqui vai alguns exemplos.

```
my_name = 'Kenzie' # string com aspas simples
mixed_quote = "Kenzie won't let me down" # string com aspas duplas, contendo uma aspas simples
multi_line = """Red sky at night:
sailor's delight. Red sky at morning:
sailor's warning."""
escaped_string = 'Here\'s looking at you, kid'
empty_string = ''
```

Indexação e Fatiamento

As strings são consideradas *listas* em Python, portanto, os caracteres dentro delas podem ser acessados via índice ou *fatiamento*.

Índices:

Para acessar um caractere em uma posição específica dentro de uma string, coloque um inteiro que representa a posição do caractere entre colchetes, []. Os números positivos começam pela esquerda, pelo 0, e os números negativos começam pela direita, pelo -1.



Módulo 2.1 - String

💰 5000



Temos um vídeo referente a este conteúdo, você pode acessá-lo [aqui](#).

Olá, desenvolvedores! O nosso conteúdo do dia englobará algumas das estruturas fundamentais do Python, as strings. Esperamos que vocês possam tirar o máximo possível de proveito do conteúdo que estamos disponibilizando, não esqueçam de praticar bastante para fixar bem.

Strings

Um objeto string em Python tem informações de caracteres e pode conter qualquer número de caracteres, inclusive zero. Muitas vezes você verá esse tipo de dado abreviado como *str*.

As strings são indicadas usando aspas simples, aspas duplas ou aspas triplas. As aspas simples são utilizadas para maioria dos casos. Se quisermos inserir aspas simples dentro da string, teremos duas alternativas, a primeira alternativa é usar o operador de escape `\` antes das aspas, e a segunda alternativa é envolver sua string com aspas duplas.

O mesmo vale para strings que contêm aspas duplas, precisamos ou anular esse caractere com o operador de escape `\`, ou envolver a string com aspas simples. As aspas triplas são usadas para strings longas que podem se estender por várias linhas. O tipo de aspa usada no final da string deve corresponder ao tipo de aspa do começo. As aspas simples podem conter aspas duplas dentro delas, e vice-versa. Aqui vai alguns exemplos.

```
my_name = 'Kenzie' # string com aspas simples
mixed_quote = "Kenzie won't let me down" # string com aspas duplas, contendo uma aspas simples
multi_line = """Red sky at night:
sailor's delight. Red sky at morning:
sailor's warning."""
escaped_string = 'Here\'s looking at you, kid'
empty_string = ''
```

Indexação e Fatiamento

As strings são consideradas *listas* em Python, portanto, os caracteres dentro delas podem ser acessados via índice ou *fatiamento*.

Índices:

Para acessar um caractere em uma posição específica dentro de uma string, coloque um inteiro que representa a posição do caractere entre colchetes, `[]`. Os números positivos começam pela esquerda, pelo 0, e os números negativos começam pela direita, pelo -1.



Módulo 2.1 - String

💰 5000



Temos um vídeo referente a este conteúdo, você pode acessá-lo [aqui](#).

Olá, desenvolvedores! O nosso conteúdo do dia englobará algumas das estruturas fundamentais do Python, as strings. Esperamos que vocês possam tirar o máximo possível de proveito do conteúdo que estamos disponibilizando, não esqueçam de praticar bastante para fixar bem.

Strings

Um objeto string em Python tem informações de caracteres e pode conter qualquer número de caracteres, inclusive zero. Muitas vezes você verá esse tipo de dado abreviado como *str*.

As strings são indicadas usando aspas simples, aspas duplas ou aspas triplas. As aspas simples são utilizadas para maioria dos casos. Se quisermos inserir aspas simples dentro da string, teremos duas alternativas, a primeira alternativa é usar o operador de escape `\` antes das aspas, e a segunda alternativa é envolver sua string com aspas duplas.

O mesmo vale para strings que contêm aspas duplas, precisamos ou anular esse caractere com o operador de escape `\`, ou envolver a string com aspas simples. As aspas triplas são usadas para strings longas que podem se estender por várias linhas. O tipo de aspa usada no final da string deve corresponder ao tipo de aspa do começo. As aspas simples podem conter aspas duplas dentro delas, e vice-versa. Aqui vai alguns exemplos.

```
my_name = 'Kenzie' # string com aspas simples
mixed_quote = "Kenzie won't let me down" # string com aspas duplas, contendo uma aspas simples
multi_line = """Red sky at night:
sailor's delight. Red sky at morning:
sailor's warning."""
escaped_string = 'Here\'s looking at you, kid'
empty_string = ''
```

Indexação e Fatiamento

As strings são consideradas *listas* em Python, portanto, os caracteres dentro delas podem ser acessados via índice ou *fatiamento*.

Índices:

Para acessar um caractere em uma posição específica dentro de uma string, coloque um inteiro que representa a posição do caractere entre colchetes, `[]`. Os números positivos começam pela esquerda, pelo 0, e os números negativos começam pela direita, pelo -1.



Módulo 2.1 - String

5000



Temos um vídeo referente a este conteúdo, você pode acessá-lo [aqui](#).

Olá, desenvolvedores! O nosso conteúdo do dia englobará algumas das estruturas fundamentais do Python, as strings. Esperamos que vocês possam tirar o máximo possível de proveito do conteúdo que estamos disponibilizando, não esqueçam de praticar bastante para fixar bem.

Strings

Um objeto string em Python tem informações de caracteres e pode conter qualquer número de caracteres, inclusive zero. Muitas vezes você verá esse tipo de dado abreviado como *str*.

As strings são indicadas usando aspas simples, aspas duplas ou aspas triplas. As aspas simples são utilizadas para maioria dos casos. Se quisermos inserir aspas simples dentro da string, teremos duas alternativas, a primeira alternativa é usar o operador de escape \ antes das aspas, e a segunda alternativa é envolver sua string com aspas duplas.

O mesmo vale para strings que contêm aspas duplas, precisamos ou anular esse caractere com o operador de escape \, ou envolver a string com aspas simples. As aspas triplas são usadas para strings longas que podem se estender por várias linhas. O tipo de aspa usada no final da string deve corresponder ao tipo de aspa do começo. As aspas simples podem conter aspas duplas dentro delas, e vice-versa. Aqui vai alguns exemplos.

```
my_name = 'Kenzie' # string com aspas simples
mixed_quote = "Kenzie won't let me down" # string com aspas duplas, contendo uma aspas simples
multi_line = """Red sky at night:
sailor's delight. Red sky at morning:
sailor's warning."""
escaped_string = 'Here\'s looking at you, kid'
empty_string = ''
```

Indexação e Fatiamento

As strings são consideradas *listas* em Python, portanto, os caracteres dentro delas podem ser acessados via índice ou *fatiamento*.

Índices:

Para acessar um caractere em uma posição específica dentro de uma string, coloque um inteiro que representa a posição do caractere entre colchetes, []. Os números positivos começam pela esquerda, pelo 0, e os números negativos começam pela direita, pelo -1.



Módulo 2.1 - String

💰 5000



Temos um vídeo referente a este conteúdo, você pode acessá-lo [aqui](#).

Olá, desenvolvedores! O nosso conteúdo do dia englobará algumas das estruturas fundamentais do Python, as strings. Esperamos que vocês possam tirar o máximo possível de proveito do conteúdo que estamos disponibilizando, não esqueçam de praticar bastante para fixar bem.

Strings

Um objeto string em Python tem informações de caracteres e pode conter qualquer número de caracteres, inclusive zero. Muitas vezes você verá esse tipo de dado abreviado como *str*.

As strings são indicadas usando aspas simples, aspas duplas ou aspas triplas. As aspas simples são utilizadas para maioria dos casos. Se quisermos inserir aspas simples dentro da string, teremos duas alternativas, a primeira alternativa é usar o operador de escape \ antes das aspas, e a segunda alternativa é envolver sua string com aspas duplas.

O mesmo vale para strings que contêm aspas duplas, precisamos ou anular esse caractere com o operador de escape \, ou envolver a string com aspas simples. As aspas triplas são usadas para strings longas que podem se estender por várias linhas. O tipo de aspa usada no final da string deve corresponder ao tipo de aspa do começo. As aspas simples podem conter aspas duplas dentro delas, e vice-versa. Aqui vai alguns exemplos.

```
my_name = 'Kenzie' # string com aspas simples
mixed_quote = "Kenzie won't let me down" # string com aspas duplas, contendo uma aspas simples
multi_line = """Red sky at night:
sailor's delight. Red sky at morning:
sailor's warning."""
escaped_string = 'Here\'s looking at you, kid'
empty_string = ''
```

Indexação e Fatiamento

As strings são consideradas *listas* em Python, portanto, os caracteres dentro delas podem ser acessados via índice ou *fatiamento*.

Índices:

Para acessar um caractere em uma posição específica dentro de uma string, coloque um inteiro que representa a posição do caractere entre colchetes, []. Os números positivos começam pela esquerda, pelo 0, e os números negativos começam pela direita, pelo -1.



Módulo 2.1 - String

💰 5000



Temos um vídeo referente a este conteúdo, você pode acessá-lo [aqui](#).

Olá, desenvolvedores! O nosso conteúdo do dia englobará algumas das estruturas fundamentais do Python, as strings. Esperamos que vocês possam tirar o máximo possível de proveito do conteúdo que estamos disponibilizando, não esqueçam de praticar bastante para fixar bem.

Strings

Um objeto string em Python tem informações de caracteres e pode conter qualquer número de caracteres, inclusive zero. Muitas vezes você verá esse tipo de dado abreviado como *str*.

As strings são indicadas usando aspas simples, aspas duplas ou aspas triplas. As aspas simples são utilizadas para maioria dos casos. Se quisermos inserir aspas simples dentro da string, teremos duas alternativas, a primeira alternativa é usar o operador de escape \ antes das aspas, e a segunda alternativa é envolver sua string com aspas duplas.

O mesmo vale para strings que contêm aspas duplas, precisamos ou anular esse caractere com o operador de escape \, ou envolver a string com aspas simples. As aspas triplas são usadas para strings longas que podem se estender por várias linhas. O tipo de aspa usada no final da string deve corresponder ao tipo de aspa do começo. As aspas simples podem conter aspas duplas dentro delas, e vice-versa. Aqui vai alguns exemplos.

```
my_name = 'Kenzie' # string com aspas simples
mixed_quote = "Kenzie won't let me down" # string com aspas duplas, contendo uma aspas simples
multi_line = """Red sky at night:
sailor's delight. Red sky at morning:
sailor's warning."""
escaped_string = 'Here\'s looking at you, kid'
empty_string = ''
```

Indexação e Fatiamento

As strings são consideradas *listas* em Python, portanto, os caracteres dentro delas podem ser acessados via índice ou *fatiamento*.

Índices:

Para acessar um caractere em uma posição específica dentro de uma string, coloque um inteiro que representa a posição do caractere entre colchetes, []. Os números positivos começam pela esquerda, pelo 0, e os números negativos começam pela direita, pelo -1.