

CAPÍTULO 8

Toda string é uma sequência de caracteres e, por isso, podemos trabalhar de diversas formas particulares com strings que não podemos com os outros tipo de variáveis.

```
>>>nome = 'Betinho'  
>>>letra = nome[n]
```

n seria o índice correspondente a letra na palavra, sendo que a primeira posição equivale a n = 0. Então, para n = 0, letra = B, n = 1, letra = e, e assim por diante. Existe também a possibilidade de ao invés de aumentar o valor de n a partir de 0, que corresponde a primeira letra, até 6, diminuirmos o valor de n a partir de -1, que corresponde a última letra, -2, a penúltima, e assim por diante.

Uma das funções exclusivas das strings é len(string) que recebe como parâmetro a string e retorna o número de caracteres que ela possui. Outra particularidade é o operador [n:m] que retorna um pedaço da string apenas, do n-ésimo carácter, incluso, até o m-ésimo carácter, exclusivo.

```
>>>frase = 'Johnny Mittos'  
>>>frase[0:6]  
'Johnny'
```

Caso o valor de n seja omitido ([:m]) a função retornará desde o primeiro carácter e caso o m seja omitido ([n:]) a função retornará até o último carácter.

Além disso, existem também os métodos para strings. Métodos são parecidos com funções, porém com a particularidade de serem exclusivos de strings. Alguns exemplos são frase.upper(), que muda todas as letras minúsculas para letras maiúsculas, e frase.find(letra), que retorna a primeira posição em que a letra na frase é a mesma que a letra do argumento. O método .find na realidade é mais genérico do que isso podendo ter como argumentos um conjunto de letras, a posição em que ele começa a procurar por esse conjunto e até que posição ele deve procurar, porém ele sempre retorna a posição onde a primeira letra do conjunto foi identificado, tendo o formato frase.find(letras, posição inicial, posição final).

Existe também um operador booleano in que retorna True se uma letra ou um determinado conjunto de letras estiver presente em uma string e False caso contrário.

```
>>>'hnnny' in 'Johnny Mittos'  
True  
>>>'Mito' in 'Johnny Mittos'  
False
```

Em Python as strings podem ser comparadas, assim podemos identificar se duas strings são iguais ou colocá-las em ordem alfabética, porém todas as letras maiúsculas vem antes que as minúsculas.