Strings in a Nutshell

Strings are literally just a sequence of characters. They are represented by " " or ' '.
Each character in a string has an index position used to identify its place by python. We can use these indexes to get fancy with strings. But before we get to that, lets get to the methods.

.upper()
As the name implies, this will convert all the characters in a string to capital letters.

```
k= "kapalot"
s="shoes"
Kcap=k.upper()
Scap =s.upper()
print(Kcap, Scap)
>>>KAPALOT SHOES
```

.lower()
Opposite of .upper(). Same application.

```
Klow = Kcap.lower()
Slow = Scap.lower()
print(Klow, Slow)
>>> kapalot shoes
```

.replace(old, new)
Once again as the name implies, .replace replaces the old characters with the new characters.

```
s = salim
New_S = s.replace("sal", "maz")
print(New_S)
>>>mazim
```

Slice
Slices a string. Literally.

```
s = 'salim'
Slicer= slice(2)
print(s[Slicer])
>>>sa
```

Lit

Len
The length of a string regardless of index. This is literal length based on no of chars.

```
s = "salim"
print(len(s))
>>>5
```

```
>>>5
```


Index

The position of each individual character. Here's how python reads strings.

S A L I M

0 1 2 3 4

We use [:] to identify chars via index. Python never counts the last index. So if i code:

```
s = 'salim'
print(s[1:3])
>>>al
```

it returns al. It only counted the letters 1-2 instead of 1-3.

Check Strings.py for more examples