BEINEX

Full Stack Internship

Strings

Strings

- String is a sequence of characters.
- String may contain alphabets,numbers and special characters.
- Usually strings are enclosed within a single and double quotes.
- **Example:**

```
a=”hello world”
b=’python’
```

Accessing Values in Strings

Example:

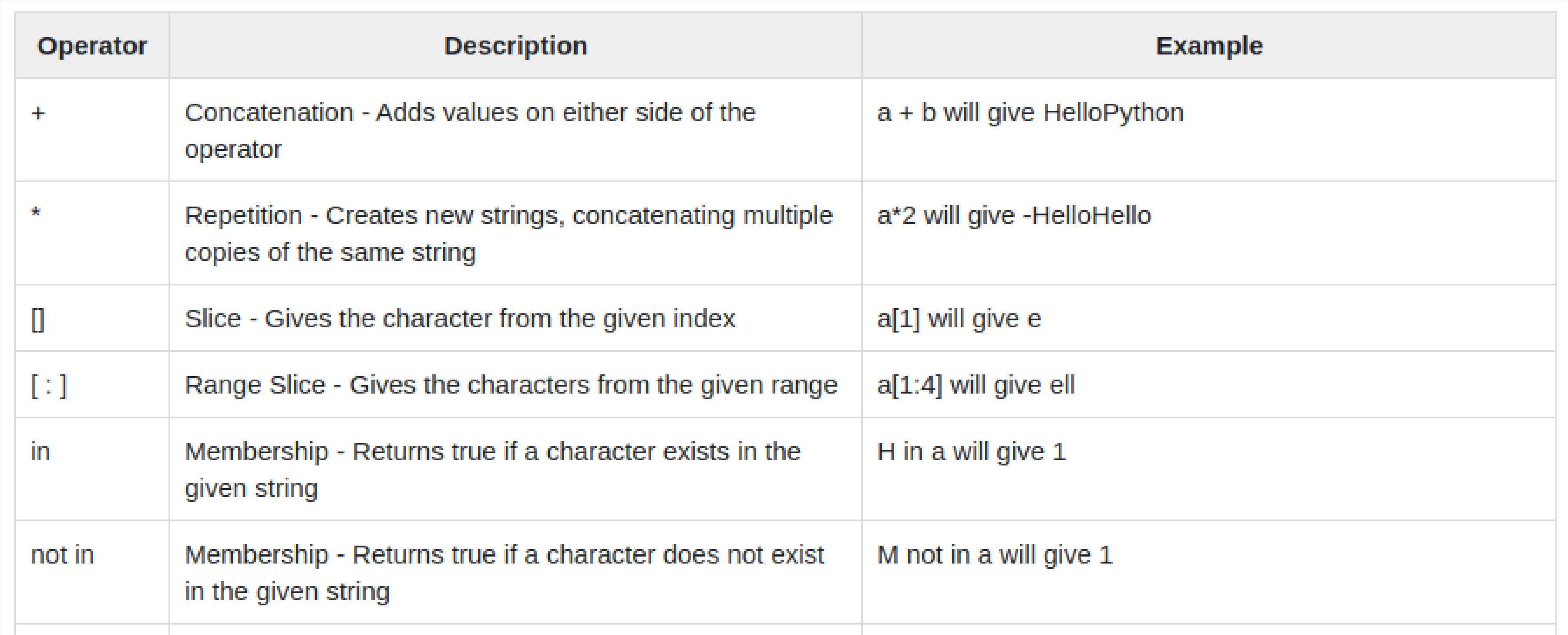
```
var2 = "Python Programming"
print (var2[0])
print (var2[1:5])
```

Strings are immutable – You cannot change any character within the string

Example:

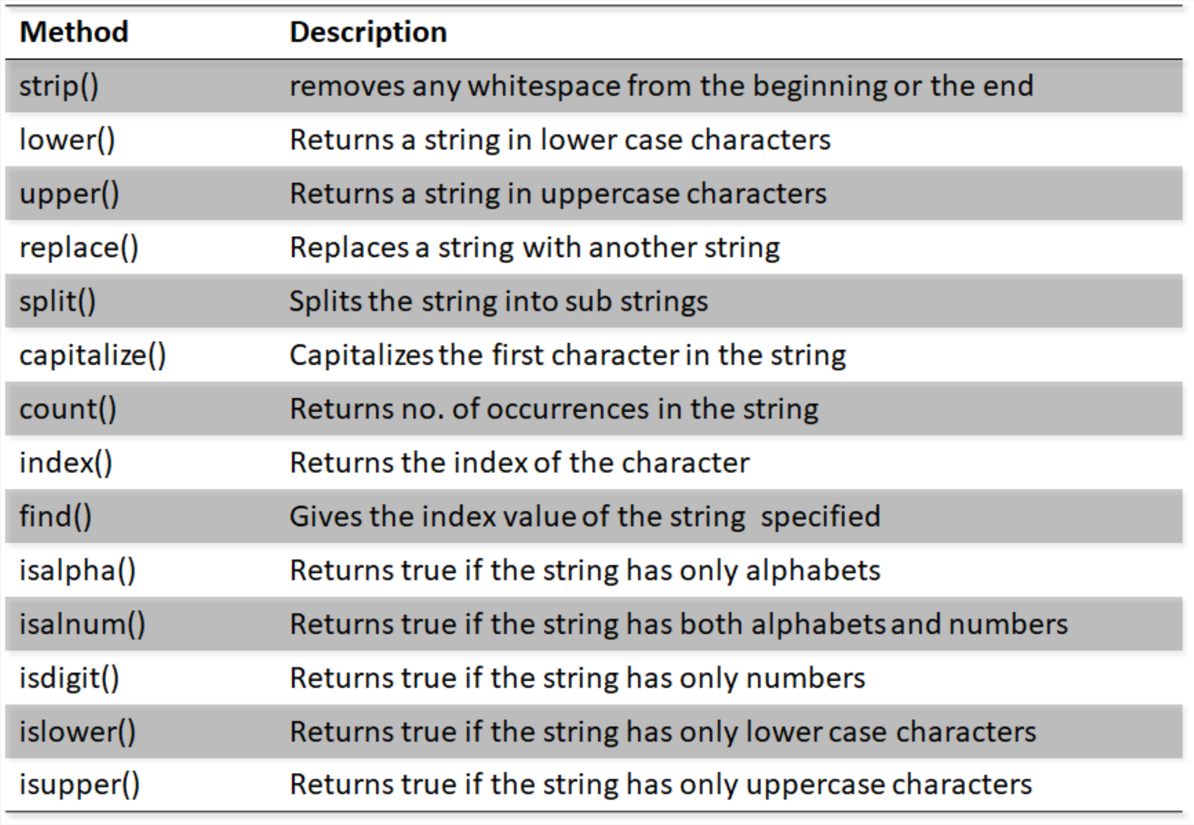
```
var1 = 'Hello World!'
Var1[1] = ‘i’
TypeError: 'str' object does not support item assignment
```

OPERATORS ON STRING



Operator	Description	Example
+	Concatenation - Adds values on either side of the operator	a + b will give HelloPython
*	Repetition - Creates new strings, concatenating multiple copies of the same string	a*2 will give -HelloHello
[]	Slice - Gives the character from the given index	a[1] will give e
[:]	Range Slice - Gives the characters from the given range	a[1:4] will give ell
in	Membership - Returns true if a character exists in the given string	H in a will give 1
not in	Membership - Returns true if a character does not exist in the given string	M not in a will give 1

Built-in String Methods



Method	Description
strip()	removes any whitespace from the beginning or the end
lower()	Returns a string in lower case characters
upper()	Returns a string in uppercase characters
replace()	Replaces a string with another string
split()	Splits the string into sub strings
capitalize()	Capitalizes the first character in the string
count()	Returns no. of occurrences in the string
index()	Returns the index of the character
find()	Gives the index value of the string specified
isalpha()	Returns true if the string has only alphabets
isalnum()	Returns true if the string has both alphabets and numbers
isdigit()	Returns true if the string has only numbers
islower()	Returns true if the string has only lower case characters
isupper()	Returns true if the string has only uppercase characters

String Formatting Operator

- **Using .format()**

  Example:

  ```
  name = 'YoungWonks'
  year = 2014

  string = '{} was started in {}'.format(name, year)

  string = '{0} was started in {1}'.format(name, year)

  string = "{name} was started in {year}.".format(name=name, year=year)
  ```

- **F strings**

  Example:

  ```
  string = f'{name} was started in {year}'
  ```

- **Using % operator**

Example:

```
name = 'YoungWonks'
year = 2014

string = 'Hello, %s ' % name

string = '%s was started in %d' % (name, year)
```