Strings

1. How to write a string:

```
str = "Hello beautiful World"

str = 'Hello beautiful World'

str = """ Hello
beautiful
World """

str = ''' Hello
beautiful
World '''
```

2- Concatenation

```
string1 =" beautiful "
string2 =" World"

print'Hello', string1, string2

print'Hello'+ string1+ string2

print"Hello {} {}".format(string1,string2)

print "Hello %s"% string2
```

For other data type use:
- %s -> string
- %d -> Integer
- %f -> Float
- %o -> Octal
- %x -> Hexadecimal
- %e -> exponential

Example

```
print "Python %d is great"%3
```

3-Repetition without loop

```
print 'Hello '*10
```

4-String formatting

Break line

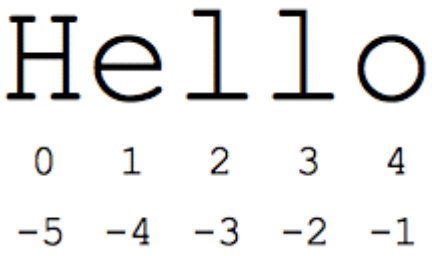
```
print "Print me in 2 \n lines."
```

Print a list

```
print """
Task to do:
\t- Revise
\t- Sleep\n\t- Do exercices
"""
```

5- Indexing

Indexes start from 0.
The last Index in a string can be referred by -1 .
Indexes can be negative : -1 is the last index, -2 is the second last index ...




```
print "Hello"[4]  # gives o
print "Hello"[5]  # gives error : IndexError: string index out of range.
```

6- Slicing

It returns a substring from a string.
The stop index will not be included.

- Syntax string[start:stop:step]
- **step is optional, default value is 1.**
- **if start is not mentioned it means start from 0.**
- **stop index will not be included in the result.**
- **if stop is not mentioned it means until and including last element.**

Examples:

```
print "Hello World"[1:2]        # gives e
print "Hello World"[:]          # gives Hello World
print "Hello World"[-2:-1]      # gives l
print "Hello World"[-2:]        # gives ld
print "Hello World"[6:]         # gives World
print "Hello World"[6::2]       # gives Wrd
```

7-Built-in Functions and methods

- len

returns string length **including** spaces
Example:

```
print len("Hello World")  # gives 11
```

·0 count

It counts the number of a substring inside a string.

Examples:

```
print "Hello World".count('l')        #prints 3
print "Hello World".count('ll')       #prints 1
print "Hello World".count('l',5)      #indicating start index : prints 1
print "Hello World".count('l',5,6)  #indicating start and stop indexes : prints 0
```

1. find

Finds a substring inside a string and returns its index, if not found it returns -1

Examples:

```
print "Hello World".find("l")
#returns 2  Be careful indexing begings from 0

print "Hello World".find("L")
#returns -1 : it means not found :  Be careful python is case sensitive


print "Hello World".find("l",3)  #start
#returns 3

print "Hello World".find("l",1,2)  #start ans stop
#returns -1 : element not found : Be careful we stop looking before the index 2(stop element is not included)
```

- index

It get the indes of a subsrting inside a string

Example:

```
print "Hello World".index('o')    # gives 4
print "Hello World".index('O')   # gives ValueError: substring not found
print "Hello World".index('llo')  # gives 2
```

- capitalize

used to capitalize the first element in the string.

Example:

```
print "hello world".capitalize()
```

- join

Join a string by itself indicating the start and the separator and the end

Example:

```
print 'Hello World'.join('*_#') # gives : *Hello World_Hello World#
```

- split

split a string using a separator into elements and create a list

Example:

```
print "Hello/World".split('/') # gives ['Hello', 'World']
```

```
print "Hello World".split('/') # gives ['Hello World']
print "Hello*World*I*Like*Python".split('*') #  ['Hello', 'World', 'I', 'Like', 'Python']
print "Hello*World*I*Like*Python".split('*',3) #['Hello', 'World', 'I', 'Like*Python']
```

- lower

Example:

```
print "HELLO WORLD".lower() #gives hello world
```

- upper

Example:

```
print "hello world".upper() #gives HELLO WORLD
```

- replace

It replace a substring by another substring inside a string.
By default it replaces all occurences.

Example:

```
print "Hello World".replace('l','a')  # gives Heaao Worad
```

But we can specify the number of replacements:

Example:

```
print "Hello World".replace('l','a',2)  # gives Heaao World: we replaced only the first 2 occurrencesof l
```

- center*

Used to center align the string by giving the field width as parameter.

Example:

```
print "Hello World".center(80)
```

- startswith*

Check if string starts with a substring or a letter.

Example:

```
print "Hello World".startswith('h') #prints False : Python is case sensitive
```

- endswith*

Check if string ends with a substring or a letter.

Example:

```
print "Hello World".endswith('d') #prints true
```

- strip*

Examples:
Removes all occurrences of a substring of both sides of string.

Example
```
print "***Hello  ***  World*".strip("*")  #gives Hello  ***  World
# it removes only from both sides.
```

- swapcase*

It converts all lowercase letters to uppercase, and all upper case to lower case.

Example:
```
str = "Hello Python 3"
print str.swapcase()   # gives hELLO pYTHON
```

Amani Ben Azzouz