Learn To Program With Python 3

Working With Strings

Working With Strings

- Documentation:
  - Python Docs Informal Tutorial: https://docs.python.org/3/tutorial/introduction.html#strings
  - Python Docs Built-In Types: https://docs.python.org/3/library/stdtypes.html#text-sequence-type-str
  - Python Docs String Methods: https://docs.python.org/3/library/stdtypes.html#string-methods

Working With Strings

- Textual data in Python is handled with str objects, or strings.
- Strings are immutable sequences of Unicode code points
- String literals are written in a variety of ways
  - Single quotes: 'allows embedded "double" quotes'
  - Double quotes: "allows embedded 'single' quotes".
  - Triple quoted: '''Three single quotes''', """Three double quotes"""

Escape Character

- The backslash “\” can be used an escape character

- This tells the compiler to change its normal behavior, such as ignoring a quotation mark in a string instead of assuming it is the end of the string.

- This is very common in programming is used in many different languages.

```
>>> print("He said "Luke, I am your father")
  File "<stdin>", line 1
    print("He said "Luke, I am your father")
                    ^
SyntaxError: invalid syntax
>>> print("He said \"Luke, I am your father")
He said "Luke, I am your father
```

New Line Character

- The backslash followed by an n “\n” can be used to create a new line, such as pressing enter normally does in a text editor

```
>>> print("1. Wake Up. 2. Eat Breakfast.")
1. Wake Up. 2. Eat Breakfast.
>>> print("1. Wake Up.\n2. Eat Breakfast.")
1. Wake Up.
2. Eat Breakfast.
```

Escaping the Escape Character – Raw Strings

- Sometimes you may want to use a backslash but NOT escape a character or create a new line.

- In this case you can use raw strings

- Raw strings are denoted by putting the ‘r’ (for raw) character in FRONT of your OPENING quotation mark

```
>>> print('C:\some\name')  # here \n means newline!
C:\some
ame
>>> print(r'C:\some\name')  # note the r before the quote
C:\some\name
```

Spanning Multiple Lines

- Strings can span multiple lines by using triple-quotes (“””…””” or ‘’’…’’’)

- Any new lines (when you press enter) are automatically included in the string (and don’t need an explicit \n newline character)

```
print("""\
Usage: thingy [OPTIONS]
     -h                        Display this usage message
     -H hostname               Hostname to connect to
""")
```

produces the following output (note that the initial newline is not included):

```
Usage: thingy [OPTIONS]
     -h                        Display this usage message
     -H hostname               Hostname to connect to
```

Spanning Multiple Lines

- Another way to span multiple lines, is that two or more strings next to each other are AUTOMATICALLY concatenated (joined together)

```
>>> text = ('Put several strings within parentheses '
...         'to have them joined together.')
>>> text
'Put several strings within parentheses to have them joined together.'
```

- Note: this method only works with two string literals, not with variables or expressions

Zero Based Index

- What is the first character in the following string “Caution! Wet Floor”
- In Python, strings can be indexed (subscripted), with the first character having index 0
  - Zero-based indexing is normal in programming!
  - Index 0 in “Caution! Wet Floor” is the character “C”
  - Index 1 in “Caution! Wet Floor” is the character “a”
  - Index 2 in “Caution! Wet Floor” is the character “u”
  - Index 3 in “Caution! Wet Floor” is the character “t”

Zero Based Index

- Use “brackets” to “subscript” or access an index
  - “Caution!”[0] – Access the zero index in the string “Caution!”
  - “Caution!”[1] – Access the one index in the string “Caution!”
- Note: Indexing is used widely in programming for more advanced data types such as lists. We will see more of this in the future.

```
>>> "Caution!"[0]
'C'
>>> "Caution!"[1]
'a'
>>> "Caution!"[2]
'u'
>>> "Caution!"[3]
't'
```

0	1	2	3	4	5	6
C	A	U	T	I	O	N

Len() built-in function

- The built-in function len() will return the length of a string

  - Documentation: https://docs.python.org/3/library/functions.html#len

```
>>> len("Caution!")
8
>>> len("Hello World")
11
>>> len("")
0
```

String Methods

- There are many methods (also known as functions) that ALL STRINGS have access to.
  - https://docs.python.org/3/library/stdtypes.html#string-methods
- The syntax is “some string”.some_method()
  - The string itself
  - A Period
  - The method call – method name followed by parentheses and any arguments the method needs

```
>>> "Hello World".upper()
'HELLO WORLD'
>>> "GET TO THE CHOPPER!".lower()
'get to the chopper!'
```

```
>>> "800-123-4567".replace("800", "")
'-123-4567'
```

Working With String Exercises

- Use the link below to find some practice exercises that deal with working with strings.
- https://github.com/ProjectFullStack/Learn-To-Program-Python3/blob/master/exercises/working_with_strings.txt

Learn To Program With Python 3

Learn Full Stack Web Development

www.projectfullstack.com