Python Library Functions

PYTHON LIBRARY FUNCTIONS

In this chapter you'll learn about:

- Python inbuilt Functions
- How to call such functions
- Pass Parameters to these functions
- How to capture the result

Python Functions

Python Functions

- **Built-in functions**
  - print, input, number & string functions
- **Library functions**
  - math.min(), math.max(), statistics.median()
- **User defined functions**
  - Functions we write ourselves

Parameters and return value

Functions can take one or more parameters

- *A value to be used in the function*
- *e.g.* `print('Hello World!')`

Return zero or one result

Do stuff!

- *Many useful things*
- *Not there? Write it! (see the next chapter)*

A function doesn't always take parameters. For example, it goes and calculates something for which it doesn't need parameters, e.g. a function **get_pi_fiveplaces()** returns the value of PI to five decimal places – no parameter necessary.

A function may not return a result. A function to delete a file on disk doesn't need to return anything – though it may be better to return the success or otherwise of the operation.

There are hundreds of built-in functions in the Python main library but it is still very common to write our own to do tasks specific to our needs.

About function parameters

A parameter can be ...

- A literal

```
print('Hello World!')
```

- A variable

```
greeting = 'Hello World'
print(greeting)
```

- An expression

```
print(100 * 0.2)
```

A literal can be an actual number or a piece of text.
A variable is a container for a value. The print function prints out the value contained in the variable.
An expression is simply a different way of saying “the result of some calculation”.
In the example the expression evaluates to 10 (2 x 3 = 6 + 4 = 10) so the print function prints 10.

Standard Library functions

- **You've seen a few inbuilt functions**
  - print, input, len, int, str, float, split

- **But there are many more**
  - Numeric functions
    - `abs, min, max, pow, round`
  - `String functions`
    - `capitalize, title, lower, upper`
    - `Zfill, format, ljust, rjust, center`
    - `isdigit, startswith, endswith, replace`

Built-in Numeric functions

```
numbers = [19,63,51,7,99,11,23,15,17,8]

print(min(numbers))

print(max(numbers))

print(pow(2,3))
(or 2**3)

print(abs(-123))
```

```
7

99

8

123
```

Numeric Functions

abs(x)

Returns the absolute (positive) value of x

max(x1, x2,...)

Returns the largest of its parameters.

min(x1, x2,...)

Returns the smallest of its parameters.

pow(x, y)

Returns the value of x raised to the power of y. E.g., pow(3,3) = 27 (3x3x3)

round(x [,n])

Returns **x** rounded to n digits from the decimal point. Python rounds 0.5 up to 1.0 and rounds -0.5 down to -1.0.

Rounding floats

```
print(round(5.671))
print(round(5.671,1))
print(round(5.671,2))
print(int(5.671))
```

```
6
5.7
5.67
5
```

Rounding floats – math library

```
import math

print( math.ceil(2000.98))

print( math.floor(2000.98))
```

```
2001
2000
```

`ceil(x)`	the smallest integer >= x
`floor(x)`	the largest integer <= x

Formatting Strings

```
str = "Bob"
print(str.center(10, '-'))
print(len(str))
print(str.ljust(7)+"Smith")
```

```
---Bob----
3
Bob  Smith
```

```
month="2"
print(month.zfill(2))
```

```
02
```

capitalize()
Capitalise first letter of a piece of text.
center(width, fillchar)
Centre text within a string, padding with a specified character.
len(string)
Returns the length of the string.
ljust(width[, fillchar])
Left justify text, padding with a specified character.
lower()
Converts all uppercase letters in string to lowercase.
upper()
Converts lowercase letters in string to uppercase.
zfill (width)
Returns original string left-padded with zeros to a total of *width* characters; intended for numbers

Lowercase and uppercase

```
str = "Bob"

print(str.lower())
```

bob

```
print(str.upper())
```

BOB

```
name = 'Bob'
if name == 'bob':
    print('Hello bob')
else:
    print("You're not bob!")
```

You're not bob!

```
if name.lower() == 'bob':
    print('Hello bob')
else:
    print("You're not bob!")
```

Hello bob

capitalize()
Capitalise first letter of a piece of text.
center(width, fillchar)
Centre text within a string, padding with a specified character.
len(string)
Returns the length of the string.
ljust(width[, fillchar])
Left justify text, padding with a specified character.
lower()
Converts all uppercase letters in string to lowercase.
upper()
Converts lowercase letters in string to uppercase.
zfill (width)
Returns original string left-padded with zeros to a total of *width* characters; intended for numbers.

String function examples

```
str = "bob smith"

print(str.capitalize())
print(str.title())

print(str.replace(' ','_'))
```

```
Bob smith
Bob Smith

bob_smith
```

```
print(str.title().replace(' ','_'))
```

```
Bob_Smith
```

Can chain functions together

Split function and IN command

- **Split a string into a List by a delimiter**

```
cities = "london,birmingham,leeds,manchester,bristol"

cityList = cities.split(',')
```

```
london
birmingham
leeds
manchester
bristol
```

```
city = input('Please enter a city name: ')

if city.lower() in cityList:
    print('Your city is in the list!')
```

Extracting part of a string

- You can extract part of a List

```
data = [1,3,5,7,9,11,13,15]
print(data[1:5])
```

[3, 5, 7, 9]

- And that includes a string

```
word = 'abcdefgh'

print(word[1:5])
```

bcde

Test before casting to int

```
strAge = input('Please enter your age: ')

if strAge.isdigit():
    age = int(strAge)
    print(age + 1)

else:
    print(strAge,'is not a valid age!')
```

What about floats? There is no isfloat() function! One easy way to do this is to remove one decimal point from the string and then test with the isdigit() function.

```
n = "1.2"
if n.replace(".", "", 1).isdigit():
   print('OK')
else:
   print("No")
```

String format function

```
name="Bob"
age=21
city="London"

str = "{} lives in {}. He is {} years old".format(name,city,age)


print (str)
```

```
Bob lives in London. He is 21 years old

Press any key to continue . . .
```

Other Libraries

- There are 100's of libraries
  https://en.wikipedia.org/wiki/Category:Python_libraries

- Here are a few from the `statistics`

Must import

```
import statistics

numbers = [99,63,51,7,99,11,23,15,17,8]

print( statistics.mean(numbers) )        # average

print( statistics.median(numbers) )      # middle value

print( statistics.mode(numbers) )        # most common data
```

Python Functions

In this chapter you learned about:

- Python inbuilt Functions
- How to call such functions
- Pass Parameters to these functions
- How to capture the result

EXERCISE

Please see your Exercise Guide

- 05-Inbuilt Functions.docx

Further Reading

- https://www.python.org/
- https://www.tutorialspoint.com/python/python_strings.htm
- https://docs.python.org/3/tutorial/index.html#tutorial-index

POWERING POTENTIAL

Thank you