POWERING
POTENTIAL

Python Functions

LESSON OBJECTIVES

In this chapter, you'll learn about:

- Python functions
- How to call functions
- Pass parameters to functions
- How to capture the result

PYTHON FUNCTIONS

There are three types of python functions:

- **Built-in Functions**
  - Print, input, number and 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 parameter

- 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!

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)
```

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
```

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
```

FORMATTING STRINGS

```
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
```

PYTHON LIBRARIES

- There are hundreds 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
```

USER DEFINED FUNCTIONS

The syntax of a Python function is the following:

```
def function_name( parameters ):
    statement1
    statement2
    …
    …
    return [expr]
```

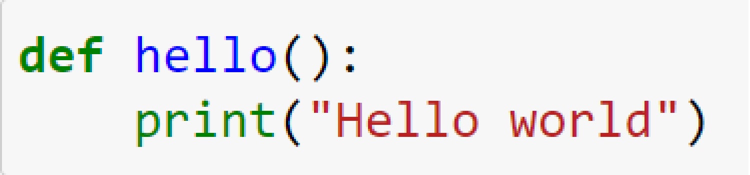
✓ **def** is a keyword that defines a function

✓ A function may or may not have parameters

✓ A function may or may not return a value

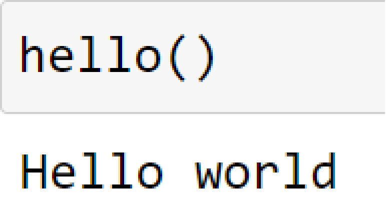
USER DEFINED FUNCTIONS

No parameters, no return value



```
def hello():
    print("Hello world")
```

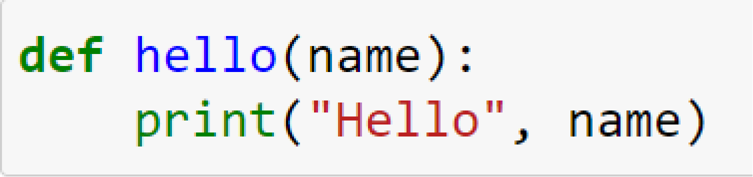
Calling the function and output:



```
hello()
```

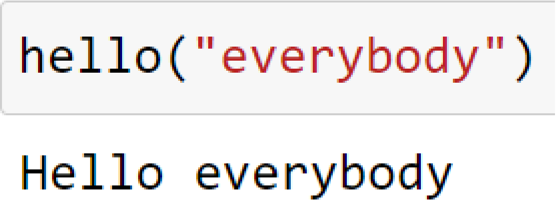
```
Hello world
```

Function with parameters, no return value



```
def hello(name):
    print("Hello", name)
```

Calling the function and output:

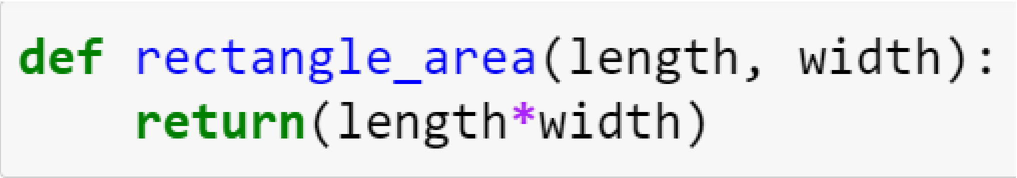


```
hello("everybody")
```

```
Hello everybody
```

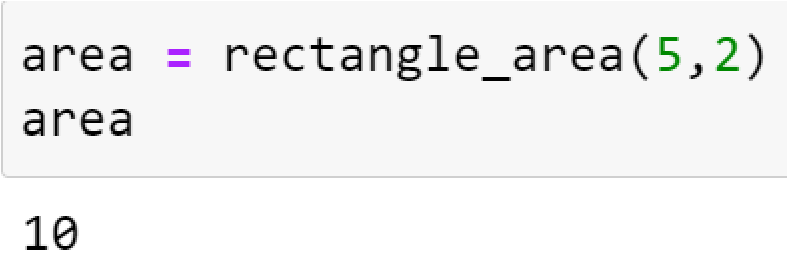
USER DEFINED FUNCTIONS

Function with parameters and return value



```
def rectangle_area(length, width):
    return(length*width)
```

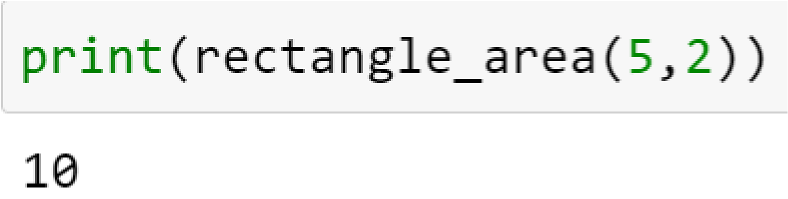
We can save the return value into a variable:



```
area = rectangle_area(5,2)
area
```

```
10
```

... or we can print it:



```
print(rectangle_area(5,2))
```

```
10
```

MODULES

User defined functions can be stored in a file called **module.**

To use functions from that module, it is necessary to **import** it (same as you would do with a library).

# to import module

import random

Now you can use any function from that module.

You need to call **module_name.function_name()**

# import module and give it a short name (alias)

import random as rn

import pandas as pd

You need to call **alias.function_name()**

SUMMARY

In this chapter, you learned about:

- Python functions
- How to call functions
- Pass parameters to functions
- How to capture the result

Further Reading

https://www.python.org/