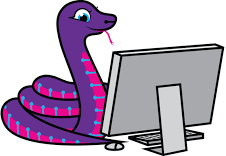
Introduction to Computer Programming



Objective

Learn the basics of computer programming using the Python language.

Requirements

Familiarity with computers, Mac or PC (navigating folders; opening, editing files, etc.) Bring your own laptop.

Course Material

This presentation can be found at "https://resq-it.com/Python". Please note this address so you can refer to the material later on.

Your instructor

Yours truly, Kal Maiwand, can be reached at [kal@highcoTech.com](mailto:kal@highcoTech.com). I have been using Python for about 10 years, and am still discobvering new features and tricks :)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Contents

* [About the Course](http://localhost:8082/resq-it/Python/about.html)
* [Table of Contents](http://localhost:8082/resq-it/Python/TOC.py)
* [Why program?](http://localhost:8082/resq-it/Python/program.html)
* [Installing Python](http://localhost:8082/resq-it/Python/install.html)
* [Using IDLE](http://localhost:8082/resq-it/Python/using-idle.html)
* [Variables](http://localhost:8082/resq-it/Python/vars.html)
* [Numbers](http://localhost:8082/resq-it/Python/nums.html)
* [Strings](http://localhost:8082/resq-it/Python/strings.html)
* [Flow Control](http://localhost:8082/resq-it/Python/flow.html)
* [Conditions](http://localhost:8082/resq-it/Python/condition.html)
* [Collections](http://localhost:8082/resq-it/Python/collections.html)
* [Functions](http://localhost:8082/resq-it/Python/functions.html)
* [Modules](http://localhost:8082/resq-it/Python/modules.html)
* [Working with Files](http://localhost:8082/resq-it/Python/files.html)
* [Resources](http://localhost:8082/resq-it/Python/resources.html)
* [Next Steps](http://localhost:8082/resq-it/Python/next_steps.html)

[PreviousPrevious](http://localhost:8082/resq-it/Python/TOC.py)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Why program?

To give a computer [specific instructions](https://en.wikipedia.org/wiki/Computer_programming) to perform a task. Instructions must be formulated such that a computer can understand them.

[What is Programming (3:30)](https://www.youtube.com/watch?v=3tWMQ3ZMjbg)[Explore: Purpose Of Programming](http://wiki.c2.com/?PurposeOfProgramming)

Why Python? python

Python is a powerful programming language, yet simple to learn. It was introduced by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) in 1990. Python has a large and dedicated comminity that develops tools for and around the language. We will probably just scratch the surfice, but learn enough to write small programs/scripts.

[Language of the year](https://www.zdnet.com/article/programming-language-of-the-year-python-is-standout-in-latest-rankings/)[Explore: Super popular](https://www.kdnuggets.com/2017/07/6-reasons-python-suddenly-super-popular.html)[Previous](http://localhost:8082/resq-it/Python/program.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/program.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Installing Python

Go to: <https://www.python.org/downloads/>, and click on 'Download Python 3.7.3' for your platform.http://localhost:8082/resq-it/Python/Assets/img-download.png

Windows

* Find the downloaded program (typically in your 'Downloads' folder): python-3.7.3-macos10.9.pkg, and double click it
* Select the default option at all prompts (‘Continue’ ,‘Agree’ and your password when needed)
* Click Start, type idle in the search window, then select IDLE Python
* [Explore: Windows Info](https://docs.python.org/3/using/windows.html)

Mac

* Find the downloaded program (typically in your 'Downloads' folder) and double click it
* Select ‘Add Python 3.7 to PATH, and click ‘Install Now’
* Select the default option at all prompts (‘Continue’ , ‘Agree’)
* in Spotlight ( command space) type idle
* [Explore: Mac Info](https://docs.python.org/3/using/mac.html)

[Previous](http://localhost:8082/resq-it/Python/install.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/install.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Using IDLE

The Python language comes with and IDLE = Integrated Development and Learning Environment

When you start IDLE, you should see something like:

Python 3.7.3 (v3.7.3:9a3ffc0492, Dec 24 2018, 02:44:43)

Type "help", "copyright", "credits" or "license()" for more information.

>>>

Type in:

>>>3+5¬

If your computer responded 8 then we are ready to dive in!

Later we will learn how to create and modify programs outside IDLE.

[Previous](http://localhost:8082/resq-it/Python/using-idle.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/using-idle.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Variables

Let's start by looking at how data is represnted. A variable is a named placeholder for a value.

>>>myVar = 7¬

The value assigned to a variable can be accessed and changed.

Rules for variable names:

* The name consists of letters, numbers and underscores (no spaces!)
* Cannot start with a number
* Cannot be one of the Python reserved words (more about these later later)
* Variable names are case sensitive

>>>anotherVar = 12.5¬

Input & Output

To interact with Python we use the 'print' and 'input' functions.

var = input( prompt)

print()

>>>print( myVar)¬

>>>name = input( 'What is your name? ')¬

>>>print( name)¬

[Previous](http://localhost:8082/resq-it/Python/vars.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/vars.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Numbers

* Examples: 24, 3.5, -1027, 3.14159
* Operators: +, -, \*, /, // (floor div), % (modulo), \*\*
* Comparison: ==, !=, >, <, >=, <=

[LearnPython.org](https://www.learnpython.org/en/Variables_and_Types)

Number methods include: abs, trunc, round, floor, ceil, and many more...

Operators follow the usual precedence rules.Down

number = 1 + 2 \* 3 / 4.0¬  
print( number)¬

Number Exercises

Convert degrees Fahrenheit to Celsius Down

C = (F - 32) \* 5/9

Decide whether a given year is a leap year Down

* Year is divisible by 400 or
* Year is divisible by 4 but not divisible by 100

Explore: variables on [LearnPython](https://www.learnpython.org/en/Variables_and_Types" \t "_blank).

[Previous](http://localhost:8082/resq-it/Python/nums.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/nums.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Strings

A string is a combination of characters surrounded by single quotes (') or double quotes (").

* Examples: 'Vienna', 'Hello world!', "The quick brown FOX"
* Operators: + (concatenate), \* (repeat), slices (substrings)
* Comparison: ==, !=, >, <, >=, <=

String examples

sentence = "The quick brown FOX"¬

sentence = sentence + ' jumps'¬

print( '~' \* 20)¬

name = input("What is your name? ")¬

print( 'Hello ' + name)¬

Python has a multitude of string methods.

print( sentence.upper(), sentence.lower())¬

>>>THE QUICK BROWN FOX the quick brown fox

print( len( sentence))¬

>>>19

Individual characters of a string can be accessed using an 'index' starting at zero.

print( sentence[4])¬

>>>q

print( sentence[ 2:8])¬

>>>e quic

String methods include: find, count, replace, capitalize, swapcase, strip, and many more...

Explore: strings on [docs.python.org](https://docs.python.org/3.7/tutorial/introduction.html#strings).

[Previous](http://localhost:8082/resq-it/Python/strings.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/strings.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Strings

A string is a combination of characters surrounded by single quotes (') or double quotes (").

* Examples: 'Vienna', 'Hello world!', "The quick brown FOX"
* Operators: + (concatenate), \* (repeat), slices (substrings)
* Comparison: ==, !=, >, <, >=, <=

String examples

sentence = "The quick brown FOX"¬

sentence = sentence + ' jumps'¬

print( '~' \* 20)¬

name = input("What is your name? ")¬

print( 'Hello ' + name)¬

Python has a multitude of string methods.

print( sentence.upper(), sentence.lower())¬

>>>THE QUICK BROWN FOX the quick brown fox

print( len( sentence))¬

>>>19

Individual characters of a string can be accessed using an 'index' starting at zero.

print( sentence[4])¬

>>>q

print( sentence[ 2:8])¬

>>>e quic

String methods include: find, count, replace, capitalize, swapcase, strip, and many more...

Explore: strings on [docs.python.org](https://docs.python.org/3.7/tutorial/introduction.html#strings).

[Previous](http://localhost:8082/resq-it/Python/strings.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/strings.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Conditions

A condition is an expression that is either true or false.

var1 == var2

num1 > num2

Below are usage examples for conditional statements:

1. Perform an action only if a condition is true.

if condition:

print( 'True')

2. Based on a condition perform one action or another.

num = 37

if( num % 2 == 0):

print('Even')

else:

print('Odd')

Note:

* Either branch can have multiple statements.
* The actions in either branch have to be indented! Indentation is Python's way of grouping actions. Statements at the same level of indentation belong together.

Explore: Conditional Statements on [LearnPython](https://www.learnpython.org/en/Conditions" \t "_blank).

[Previous](http://localhost:8082/resq-it/Python/condition.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/condition.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Collections

The variables we looked at so far contain a single value, a numbr , a string, etc.

Many problems require dealing with collections of data. Pyhton supports several types of collection (array, tuple, dictionary, set, etc.). We will look at lists, that are most common.

Working with Lists

>>>myList = [12.5, 4, 102, -35]

>>>childres = ['Heiko', 'Nina']

List operations include: +, insert, append, sort, and many more.

Individual elements of a list can be accessed using an 'index' starting at zero. The last index is -1.

>>>myList[0] = 11

>>>print( myList[2])

>>>print( myList[-1])

To iterate over all values of a list we use the 'for' loop.

for element in myList:

print( element)

Exercises

myList = [ 'spam', 'eggs', 'rice']

Create a string consisting of list elements separated by '\_' Down

Reverse the order of elements Down

Sort elements Down

Explore: lists on [programiz.com](https://www.programiz.com/python-programming/list).

[Previous](http://localhost:8082/resq-it/Python/collections.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/collections.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Functions

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result.

Examples for function definition and invocation:

Function as a shortcut

def my\_function( str):

print("From my\_function: " + str)

my\_function( 'Hello!')

Function with a return value

def avg( lst):

result = 0

for el in lst:

result += el

return result/len(lst)

print( avg( [12, 34, 65]))

[Previous](http://localhost:8082/resq-it/Python/functions.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/functions.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Python Modules

A module is a file containing Python definitions and statements. The file name is the module name with the suffix '.py' appended.

User Modules

If you quit from the Python interpreter and enter it again, the definitions you have made (functions and variables) are lost. Therefore, if you want to write a somewhat longer program, you are better off using a text editor to prepare the input for the interpreter and running it with that file as input instead.

Explore: modules [docs.python.org](https://docs.python.org/3/tutorial/modules.html).

**Exercise**

Create a module containing a function that takes a radius as input and returns the area of the circle.

System Modules

Python comes with an extensive library of [standard modules](https://docs.python.org/3/library). These extend the core language and can be platform dependent.

Below is an example for using a standard library

import random

def gamble():

min, max = 1, 6

roll\_again = "y"

while roll\_again == "y":

print( "Rolling the dices...", random.randint(min, max), ',', random.randint(min, max))

roll\_again = input("Roll the dices again (y/n)?")

gamble()

External Modules

There is a plethora of third-party modules for all kinds of needs and purposes. These typically need to be downloaded/installed before you can use them.

[Previous](http://localhost:8082/resq-it/Python/modules.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/modules.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Working with Files

Reading and writing files:

with open( "input.txt") as f:  
data = f.read()  
  
with open( "table.html", 'w') as f:  
f.write( content)

File Exercises

Examples below illustrate batch manipulation of files, a common task in System Administration.

Rename all files in a directory Down

Create a test folder with some random files to test the script.

Search for a pattern in a directoryDown

Create a test folder with some text files to test the script.

Explore: File Handling on [PythonForBeginners.com](https://www.pythonforbeginners.com/cheatsheet/python-file-handling).

[Previous](http://localhost:8082/resq-it/Python/files.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/files.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Online Resources

There are lots of resources related to Python available online. The list below is a sampling.

* [Python For Beginners](https://www.python.org/about/gettingstarted/)
* [Learning Python](https://docs.python-guide.org/intro/learning/)
* [learnpython.org](https://www.learnpython.org/)
* [programiz.com](https://www.programiz.com/python-programming)
* [pynative.com](https://www.pynative.com/)
* [codecademy.com](hhttps://www.codecademy.com/learn/learn-python-3)
* [snipt.net](https://snipt.net/public/tag/python)

On YouTube:

* [Python Tutorial for Beginners](https://www.youtube.com/watch?v=_uQrJ0TkZlc)
* [Full Course for Beginners](https://www.youtube.com/watch?v=rfscVS0vtbw)
* [Python - Crash Course](https://www.youtube.com/watch?v=yE9v9rt6ziw)

[Previous](http://localhost:8082/resq-it/Python/resources.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/resources.html)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Next Steps

Explore topics we did not have time to cover:

Object Orientation, exception handling, patter matching, databases, etc.

Look on the Web for Python examples (see [Online Resources](http://localhost:8082/resq-it/Python/resources.html)). Learn from other people!

There are many uses for Python, find what interests you:

* Web development - cgi module
* Data analysis - Pandas
* Music, audio synthesis
* Graphics and games - graphics.py, PyGame
* System administration
* and many, many more...

Explore: modules on [wiki.python.org](https://wiki.python.org/moin/UsefulModules).

[Previous](http://localhost:8082/resq-it/Python/next_steps.html)[Contents](http://localhost:8082/resq-it/Python/TOC.py)[Previous](http://localhost:8082/resq-it/Python/next_steps.html)