



# Python Programming

Nathan Stern  
([nstern@wearelcc.ca](mailto:nstern@wearelcc.ca))

Alex Lai  
([alai@wearelcc.ca](mailto:alai@wearelcc.ca))

March 2018

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Hello, World!</b>	<b>2</b>
<b>3</b>	<b>Variables</b>	<b>2</b>
3.1	Integers . . . . .	3
3.2	Strings . . . . .	3
3.3	Arrays . . . . .	3
3.4	Dictionaries . . . . .	3
3.5	Printing Challenge . . . . .	4
<b>4</b>	<b>Loops</b>	<b>4</b>
<b>5</b>	<b>Functions</b>	<b>4</b>
5.1	Functions Challenge . . . . .	4
<b>6</b>	<b>Activities</b>	<b>4</b>
6.1	Quadratic Equation Solver . . . . .	4
6.2	Number Guessing Game . . . . .	4
6.3	Random Word Picker . . . . .	4
6.4	Hello, World! in PyGame . . . . .	4
6.5	Snake in PyGame . . . . .	4
<b>7</b>	<b>Challenge Solutions</b>	<b>4</b>
7.1	Printing Challenge . . . . .	4
7.2	Functions Challenge . . . . .	5

# 1 Introduction

This document is a quick introduction to programming in Python. This guide will cover everything that you will need to know in order to start programming in Python. Throughout the document, there will be some challenges to test the knowledge that you have gained up until that point. These challenges are designed to really test what you have learned, as well as testing your creative thinking and problem solving.

This document references many different sections throughout it. Whenever you see a reference, you can click on the section number that is provided to take you to the referenced section.

## 2 Hello, World!

This is the most basic program that any programmer will create when they first start learning a new language. It is usually the shortest program that they will ever make.

```
print ("Hello, World")
```

In Python, the print function is very simple to use. You simply have to type print() with the string that you want to print inside of the parenthesis with quotes around it. You can also use the print function to print the value of a variable. To do this, all you have to do is type print(VARIABLE).

## 3 Variables

Variables are a way of storing values in Python. There are many different kinds of variables that you can use in Python. The most common of these variable types are **Integers** and **Strings**.

Variables are defined very simply, and you do not have to specify the type of the variable when it is first created. The Python interpreter will automatically determine the type of the variable for you.

```
myInt = 5
myString = "Hello"
myArray = [1, 2, 4, 5]
myDict = {5: "What's Up?", 8: "Hello"}
```

In that example, there were different kinds of variables created. In future sections of this document, we will explore each of these different variables and how to use them.

### 3.1 Integers

An integer is one of the types of variables in Python. This variable type contains whole numbers. Because the Python interpreter automatically determines the variable type, you do not have to worry about the different types of numbers that you can have in Python. The different types are: Integer, Short, Long, Double, and Float. These different kinds of numbers have different data sizes associated with them, but you do not normally need to worry about that with Python.

Defining an integer is very easy to do in Python.

```
myInt = 5
```

All you have to do is define the variable name and set it equal to a value. This is the basis for defining any variable in Python.

### 3.2 Strings

Strings in Python are normally used when asking the User for an Input, or when you want to relay information to the User. Defining a string is very similar to defining an integer.

```
myString = "Hello"
```

The main difference between defining a string from defining an integer is the fact that when you define a string, you put quotes around the whatever you want to store as a string.

### 3.3 Arrays

Arrays in Python are variables that can store as many **individual** values that you want. When defining an array, you put all of the values, separated with commas, inside of square brackets.

```
myArr = [1, 5, 71, 234, 6]
```

Arrays can be used in many different ways when programming. You can add values to the end of an array, which will be covered in section 4 of this document.

### 3.4 Dictionaries

Dictionaries are very similar to arrays. In a dictionary, the data that is passed when it is first created is similar to that of a real dictionary. This means that when you create a dictionary in Python, you supply a value and a definition.

```
myDict = {1: "This", 2: "is", 3: "a", 4: "dictionary"}
```

The first value in a dictionary is the value that you want to reference when looking at dictionary. The next value, separated by a colon, is the definition of the dictionary value that you referenced. This is especially useful when you want to keep different values in the same place, somewhat like an actual dictionary.

It is very easy to retrieve the definition of a value in a dictionary.

```
print(myDict[2])  
  
# This will return the definition stored in the  
# dictionary under the name "2".
```

### 3.5 Printing Challenge

Your first challenge is to print anything you want into the console using a variable. If you can't remember how to do that, you can reference section 2 for help.

## 4 Loops

## 5 Functions

### 5.1 Functions Challenge

Your challenge is to create a function that lists the values of an array. The answer can be found in section 7.2 of this document.

## 6 Activities

### 6.1 Quadratic Equation Solver

### 6.2 Number Guessing Game

### 6.3 Random Word Picker

### 6.4 Hello, World! in PyGame

### 6.5 Snake in PyGame

## 7 Challenge Solutions

### 7.1 Printing Challenge

The original challenge activity can be found in section 3.5.

Solution:

```
myVar = "This is an example of a string stored in a variable."  
  
print(myVar)
```

## 7.2 Functions Challenge

The original challenge activity can be found in section 5.1.

Solution:

```
myArr = [5, 12, 242, 1, 6]  
  
def ArrayList():  
    for value in myArr:  
        print(value)  
  
ArrayList()
```