Hello Python! - Reading Material

Some good programming resources available online:

- The official Python tutorial. This tutorial is designed to help people teach themselves Python. While it goes in a different order than the one we're taking here, it covers a lot of the same subjects that we explore in this course. You can refer to this resource for extra information on these subjects.
- The Think Python book. This book aims to teach people how to program in Python. It's available online in PDF and browsable forms. Again, you can use this resource to learn more about some of the subjects we cover.

The official language reference. This is a technical reference of all Python language components. At first, this resource might be a little too complex, but as you learn how Python works and how it's built, this can be a useful reference to understand the details of these interactions.

More About Python

Using Python on your own

The best way to learn any programming language is to practice it on your own as much as you can. If you have Python installed on your computer, you can execute the interpreter by running the python3 command (or just python on Windows), and you can close it by typing exit() or Ctrl-D.

If you don't already have Python installed on your machine, that's alright. We'll explain how to install it in an upcoming course.

In the meantime, you can still practice by using one of the many online Python interpreters or codepads available online. There's not much difference between an interpreter and a codepad. An interpreter is more interactive than a codepad, but they both let you execute code and see the results.

Below, you'll find links to some of the most popular online interpreters and codepads. Give them a go to find your favorite.

- https://www.python.org/shell/
- https://www.onlinegdb.com/online_python_interpreter
- https://repl.it/languages/python3
- https://www.tutorialspoint.com/execute_python3_online.php
- https://rextester.com/l/python3_online_compiler
- https://trinket.io/python3

Additional Python resources

While this course will give you information about how Python works and how to write scripts in Python, you'll likely want to find out more about specific parts of the language. Here are some great ways to help you find additional info:

- Read the official Python documentation.
- Search for answers or ask a question on Stack Overflow.
- Subscribe to the Python tutor mailing list, where you can ask question and collaborate with other Python learners.
- Subscribe to the Python-announce mailing list to read about the latest updates in the language.

Python history and current status

Python was released almost 30 years ago and has a rich history. You can read more about it on the History of Python Wikipedia page or in the section on the history of the software from the official Python documentation.

Python has recently been called the fastest growing programming language. If you're interested in why this is and how it's measured, you can find out more in these articles:

- The Incredible Growth of Python (Stack Overflow)
- Why is Python Growing So Quickly - Future Trends (Netguru)
- By the numbers: Python community trends in 2017/2018 (Opensource.com)
- Developer Survey Results 2018 (Stack Overflow)

First Programming Concepts Cheat Sheet

Functions and Keywords

Functions and keywords are the building blocks of a language's syntax.

Functions are pieces of code that perform a unit of work. In the examples we've seen so far, we've only encountered the print() function, which prints a message to the screen. We'll learn about a lot of other functions in later lessons but, if you're too curious to wait until then, you can discover all the functions available here.

Keywords are reserved words that are used to construct instructions. We briefly encountered for and in in our first Python example, and we'll use a bunch of other keywords as we go through the course. For reference, these are all the reserved keywords:

False	class	finally	is	return

None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

You don't need to learn this list; we'll dive into each keyword as we encounter them. In the meantime, you can see examples of keyword usage here.

Arithmetic operators

Python can operate with numbers using the usual mathematical operators, and some special operators, too. These are all of them (we'll explore the last two in later videos).

- **a + b** = Adds a and b
- **a - b** = Subtracts b from a
- **a * b** = Multiplies a and b
- **a / b** = Divides a by b
- **a ** b** = Elevates a to the power of b. For non integer values of b, this becomes a root (i.e. a**(1/2) is the square root of a)
- **a // b** = The integer part of the integer division of a by b
- **a % b** = The remainder part of the integer division of a by b

Some good programming resources available online:

- The official Python tutorial. This tutorial is designed to help people teach themselves Python. While it goes in a different order than the one we're taking here, it covers a lot of the same subjects that we explore in this course. You can refer to this resource for extra information on these subjects.
- The Think Python book. This book aims to teach people how to program in Python. It's available online in PDF and browsable forms. Again, you can use this resource to learn more about some of the subjects we cover.

The official language reference. This is a technical reference of all Python language components. At first, this resource might be a little too complex, but as you learn how Python works and how it's built, this can be a useful reference to understand the details of these interactions.