**Introduction to Programming**

If you’ve never written a for-loop, or don’t know what a string is in programming, start here. This course is self-paced, allowing you to adjust the number of hours you spend per week to meet your needs.

**Topics covered**: simple programs simple data structures

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Courses** | **Duration** | **Effort** | **Prerequisites** | **Discussion** |
| [Introduction to programming](https://cs.ossu.dev/coursepages/intro-programming/) | 10 weeks | 10 hours/week | none | [chat](https://discord.gg/syA242Z) |

**Introduction to Programming**

If you’ve never written a for-loop, or don’t know what a string is in programming, start here. These courses are self-paced, allowing you to adjust the number of hours you spend per week to meet your needs.

We are currently looking for volunteers to try out both of the following two courses and analyze them in different ways to determine which one is better suited to be included in our curriculum. We suggest that you flip a coin to decide which one to take first, so that you avoid an ordering bias. Once you have completed both courses, please provide your analysis of [this RFC](https://github.com/ossu/computer-science/issues/1164).

If you don’t have time or do not want to volunteer, you are required to do **only ONE** of the following courses.

**CS50P: Introduction to Programming with Python**

This course has been developed by the CS50 team at Harvard University.

An introduction to programming using a language called Python. Learn how to read and write code as well as how to test and “debug” it. Designed for students with or without prior programming experience who’d like to learn Python specifically. Learn about functions, arguments, and return values (oh my!); variables and types; conditionals and Boolean expressions; and loops. Learn how to handle exceptions, find and fix bugs, and write unit tests; use third-party libraries; validate and extract data with regular expressions; model real-world entities with classes, objects, methods, and properties; and read and write files. Hands-on opportunities for lots of practice. Exercises inspired by real-world programming problems. No software required except for a web browser, or you can write code on your own PC or Mac.

**Link**: <https://cs50.harvard.edu/python/>

**Note**: This course is *different* from CS50 or CS50x. CS50 is not part of the OSSU curriculum. That being said, if you have completed CS50, you can skip this course and move on to the next one.

**Instructions**

* If you want to follow along with the instructor, log in to the [CS50 “codespace”](https://cs50.dev) and watch [this video](https://cs50.harvard.edu/python/2022/shorts/visual_studio_code_for_cs50/) to get started.
* Watch each lecture and complete the respective problem set. Read the lecture notes to revise things.
* If you are stuck somewhere, feel free to ask questions. You can join the OSSU chat for this course here: <https://discord.gg/cBkssaJy5g>.
* You can also join the CS50 discord server and ask questions there: <https://discord.gg/cs50>, but note that it is not affiliated with or maintained by OSSU.

**Course Materials**

1. [Functions, Variables](https://cs50.harvard.edu/python/2022/weeks/0/) — [Notes](https://cs50.harvard.edu/python/2022/notes/0/) — [Problem Set](https://cs50.harvard.edu/python/2022/psets/0/)
2. [Conditionals](https://cs50.harvard.edu/python/2022/weeks/1/) — [Notes](https://cs50.harvard.edu/python/2022/notes/1/) — [Problem Set](https://cs50.harvard.edu/python/2022/psets/1/)
3. [Loops](https://cs50.harvard.edu/python/2022/weeks/2/) — [Notes](https://cs50.harvard.edu/python/2022/notes/2/) — [Problem Set](https://cs50.harvard.edu/python/2022/psets/2/)
4. [Exceptions](https://cs50.harvard.edu/python/2022/weeks/3/) — [Notes](https://cs50.harvard.edu/python/2022/notes/3/) — [Problem Set](https://cs50.harvard.edu/python/2022/psets/3/)
5. [Libraries](https://cs50.harvard.edu/python/2022/weeks/4/) — [Notes](https://cs50.harvard.edu/python/2022/notes/4/) — [Problem Set](https://cs50.harvard.edu/python/2022/psets/4/)
6. [Unit Tests](https://cs50.harvard.edu/python/2022/weeks/5/) — [Notes](https://cs50.harvard.edu/python/2022/notes/5/) — [Problem Set](https://cs50.harvard.edu/python/2022/psets/5/)
7. [File I/O](https://cs50.harvard.edu/python/2022/weeks/6/) — [Notes](https://cs50.harvard.edu/python/2022/notes/6/) — [Problem Set](https://cs50.harvard.edu/python/2022/psets/6/)
8. [Regular Expressions](https://cs50.harvard.edu/python/2022/weeks/7/) — [Notes](https://cs50.harvard.edu/python/2022/notes/7/) — [Problem Set](https://cs50.harvard.edu/python/2022/psets/7/)
9. [Object-Oriented Programming](https://cs50.harvard.edu/python/2022/weeks/8/) — [Notes](https://cs50.harvard.edu/python/2022/notes/8/) — [Problem Set](https://cs50.harvard.edu/python/2022/psets/8/)
10. [Et Cetera](https://cs50.harvard.edu/python/2022/weeks/9/) — [Notes](https://cs50.harvard.edu/python/2022/notes/9/) — [Final Project](https://cs50.harvard.edu/python/2022/project/)

**Python for Everybody**

This course has been created by Professor Charles Severance from the University of Michigan.

Learn to Program and Analyze Data with Python. Develop programs to gather, clean, analyze, and visualize data.

**Link**: <https://www.py4e.com/lessons>

**Textbook**: [PDF](http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf) / [EPUB](http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.epub) / [HTML](https://www.py4e.com/html3) / [Buy hardcopy](https://www.py4e.com/book)

**Note**: This course is also offered on Coursera, Edx. Those versions require you to pay to get the full version of the course. We suggest doing the course on its website, which is completely free.

**Instructions**

* You need to [sign in](https://www.py4e.com/) to the course website using your Google account to access the assignments.
* Watch all the videos of a lesson and then do its assignments.
* If you prefer reading books, you can read the HTML version of the chapter related to the lesson linked on the lesson’s page, or you can download the whole book in different formats from [this page](https://www.py4e.com/book).
* If you face any problems, feel free to ask questions. You can join the OSSU chat for this course here: <https://discord.gg/syA242Z>.
* You only need to complete the course up to the Regular Expressions lesson. The rest of the course is optional.

**Course Materials**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Course | location | progress |
| 01 | [Installing Python](https://www.py4e.com/lessons/install) |  |  |
| 02 | [Why Program?](https://www.py4e.com/lessons/intro) |  |  |
| 03 | [Variables, expressions and statements](https://www.py4e.com/lessons/memory) | [GO TO](#Bookmark1) |  |
| 04 | [Conditional Execution](https://www.py4e.com/lessons/logic) |  |  |
| 05 | [Functions](https://www.py4e.com/lessons/functions) |  |  |
| 06 | [Loops and Iterations](https://www.py4e.com/lessons/loops) |  |  |
| 07 | [Strings](https://www.py4e.com/lessons/strings) |  |  |
| 08 | [Files](https://www.py4e.com/lessons/files) |  |  |
| 09 | [Lists](https://www.py4e.com/lessons/lists) |  |  |
| 10 | [Dictionaries](https://www.py4e.com/lessons/dictionary) |  |  |
| 11 | [Tuples](https://www.py4e.com/lessons/tuples) |  |  |
| 12 | [Regular Expressions](https://www.py4e.com/lessons/regex) |  |  |
| 13 | [Network Programming](https://www.py4e.com/lessons/network) (Optional) |  |  |
| 14 | [Using Web Services](https://www.py4e.com/lessons/servces) (Optional) |  |  |
| 15 | [Object-Oriented Programming](https://www.py4e.com/lessons/Objects) (Optional) |  |  |
| 16 | [Databases](https://www.py4e.com/lessons/database) (Optional) |  |  |
| 17 | [Data Visualization](https://www.py4e.com/lessons/dataviz) (Optional) |  |  |

**Fixes**

1. If you’re doing the BeautifulSoup4 lesson, there is an issue with Python 3.10+ that will give you an error referencing the Collections library. We have a fix for you. We don’t expect you to understand it, just put this in front of your code in the imports block:

import collections

collections.Callable = collections.abc.Callable

from bs4 import BeautifulSoup

Doing this should fix the compatibility issue and allow your code to run.

03 [Variables, expressions and statements](https://www.py4e.com/lessons/memory)

CLI = command line interface.

Bug = a mistake, problems that I need to solve

print("hello,world")

\*Return values and variables:

- input: (it is expecting a string)

name = input("what is your name")

print("hello"+name)

Now we have a return value from name that we can put in the print function

\*Comments and pseudocode

name = input("what is your name")

'''

this is how to write multi-line comment

'''

print("hello"+name)

We use the hash ## to comment in the code so the interrupter to ignore

- Pseudocode = is to express our thoughts or to-do-list in the code

there is two ways to write the print() with more than one thing in it

We could do this

We are using one argument so we have to put a space after hello

print("hello "+name)

Or

But here we are using two arguments, thus we dont need to put a space

print("hello",name)

\*Named parameters

We have the print() this function it has more than one argument

It is like this print(\*objects, sep=” “ , end=”\n”)

Objects are arguments

Sep and end are named parameters

\*f-Strings

name = input("what is your name")

#f-formatting

print(f"hello {name}")

\*string methods

sometimes we hit the space bar by mistake it will look ugly

So we need to use the method strip() to remove the spaces before and after.

* Deleting spaces right and left the value David Malan

name = input("what is your name")

#f-formatting

#maybe by mistake we did hit space we can use the function strip

name = name.strip()

print(f"hello {name}")

* Capitalize the strings but only the first word

name = input("what is your name")

#f-formatting

#maybe by mistake we did hit space we can use the function strip

name = name.strip()

name = name.capitalize()

print(f"hello {name}")

* Capitalize the strings all the first letters

name = input("what is your name")

#f-formatting

#maybe by mistake we did hit space we can use the function strip

name = name.strip()

name = name.title()

print(f"hello {name}")

* We can put the two methods together in one line

name = name.strip().title()

we can also put the two methods in one line with the input

name = input("what is your name").strip().title()

#f-formatting

#maybe by mistake we did hit space we can use the function strip

print(f"hello {name}")