The Programming Historian

Creating and Viewing HTML Files with Python (/en/lessons/creating-and-viewing-html-files-with-python)

William J. Turkel and Adam Crymble

Here you will learn how to create HTML files with Python scripts, and how to use Python to automatically open an HTML file in Firefox.

Peer-reviewed

Support PH (https://programminghistorian.org/en/support-us#donate-to-us)

edited by

- Miriam Posner

reviewed by

- Jim Clifford

published

2012-07-17

modified

2020-01-20

difficulty

Medium

This lesson has been translated into Spanish: Crear y ver archivos HTML con Python (/es/lecciones/crear-y-ver-archivos-html-con-python)

This lesson is part of a series. You might want to check out the previous lesson (counting-frequencies).

Contents

Lesson Goals

This lesson uses Python to create and view an HTML file. If you write programs that output HTML, you can use any browser to look at your results. This is especially convenient if your program is automatically creating hyperlinks or graphic entities like charts and diagrams.

Here you will learn how to create HTML files with Python scripts, and how to use Python to automatically open an HTML file in Firefox.

Files Needed For This Lesson

- `obo.py`

If you do not have these files from the previous lesson, you can download programming-historian-5, a zip file from the previous lesson (/assets/python-lessons5.zip).

Creating HTML with Python

At this point, we’ve started to learn how to use Python to download online sources and extract information from them automatically. Remember that our ultimate goal is to incorporate programming seamlessly into our research practice. In keeping with this goal, in this lesson and the next, we will learn how to output data back as HTML. This has a few advantages. First, by storing the information on our hard drive as an HTML file we can open it with Firefox and use Zotero (https://zotero.org) to index and annotate it later. Second, there are a wide range of visualization options for HTML which we can draw on later.

If you have not done the W3 Schools HTML tutorial (https://www.w3schools.com/html/default.asp) yet, take a few minutes to do it before continuing. We’re going to be creating an HTML document using Python, so you will have to know what an HTML document is!

“Hello World” in HTML using Python

One of the more powerful ideas in computer science is that a file that seems to contain code from one perspective can be seen as data from another. It is possible, in other words, to write programs that manipulate other programs. What we’re going to do next is create an HTML file that says “Hello World!” using Python. We will do this by storing HTML *tags* in a multiline Python *string* and saving the contents to a new file. This file will be saved with an `.html` extension rather than a `.txt` extension.

Typically an HTML file begins with a doctype declaration (https://www.w3schools.com/tags/tag_doctype.asp). You saw this when you wrote an HTML “Hello World” program in an earlier lesson. To make reading our code easier, we will omit the doctype in this example. Recall a multi-line string is created by enclosing the text in three quotation marks (see below).

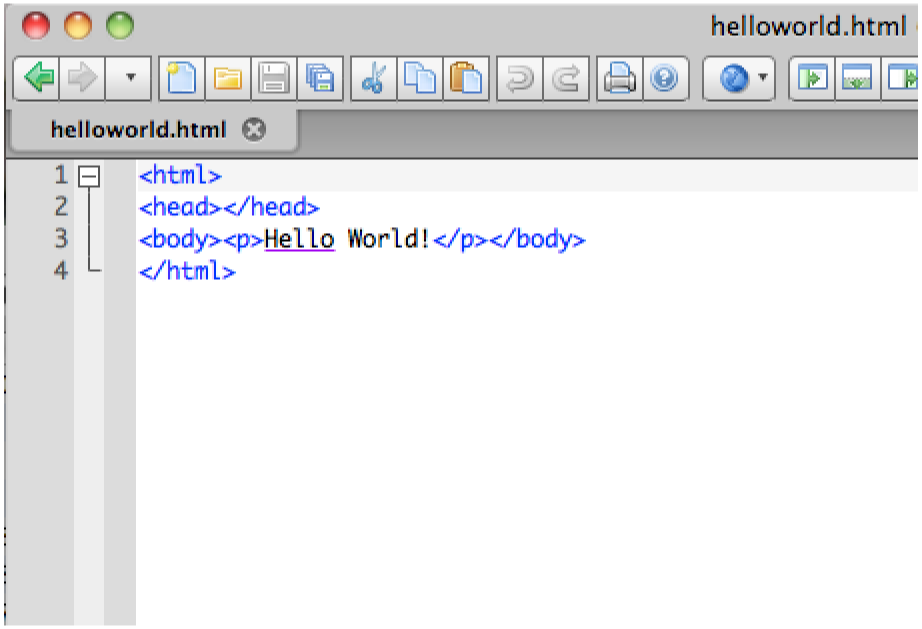
```
# write-html.py

f = open('helloworld.html','wb')

message = """<html>
<head></head>
<body><p>Hello World!</p></body>
</html>"""

f.write(message)
f.close()
```

Save the above program as `write-html.py` and execute it. Use *File* -> *Open* in your chosen text editor to open `helloworld.html` to verify that your program actually created the file. The content should look like this:



```
<html>
<head></head>
<body><p>Hello World!</p></body>
</html>
```

HTML Source Generated by Python Program

Now go to your Firefox browser and choose *File* -> *New Tab*, go to the tab, and choose *File* -> *Open File*. Select `helloworld.html`. You should now be able to see your message in the browser. Take a moment to think about this: you now have the ability to write a program which can automatically create a webpage. There is no reason why you could not write a program to automatically create a whole website if you wanted to.

Using Python to Control Firefox

We automatically created an HTML file, but then we had to leave our editor and go to Firefox to open the file in a new tab. Wouldn't it be cool to have our Python program include that final step? Type or copy the code below and save it as `write-html-2.py`. When you execute it, it should create your HTML file and then automatically open it in a new tab in Firefox. Sweet!

Mac Instructions

Mac users will have to specify to the precise location of the `.html` file on their computer. To do this, locate the `programming-historian` folder you created to do these tutorials, right-click it and select "Get Info".

You can then cut and paste the file location listed after "Where:" and make sure you include a trailing slash (/) to let the computer know you want something inside the directory (rather than the directory itself).

```
# write-html-2-mac.py
import webbrowser

f = open('helloworld.html','wb')

message = """<html>
<head></head>
<body><p>Hello World!</p></body>
</html>"""

f.write(message)
f.close()

#Change path to reflect file location
filename = 'file:///Users/username/Desktop/programming-historian/' +
'helloworld.html'
webbrowser.open_new_tab(filename)
```

If you’re getting a “File not found” error you haven’t changed the filename path correctly.

Windows Instructions

```
# write-html-2-windows.py

import webbrowser

f = open('helloworld.html','wb')

message = """<html>
<head></head>
<body><p>Hello World!</p></body>
</html>"""

f.write(message)
f.close()

webbrowser.open_new_tab('helloworld.html')
```

***

Not only have you written a Python program that can write simple HTML, but you've now controlled your Firefox browser using Python. In the next lesson, we turn to outputting the data that we have collected as an HTML file.

Suggested Readings

- Lutz, Learning Python
  - Re-read and review Chs. 1-17

Code Syncing

To follow along with future lessons it is important that you have the right files and programs in your "programming-historian" directory. At the end of each lesson in the series you can download the "programming-historian" zip file to make sure you have the correct code. If you are following along with the Mac / Linux version you may have to open the `obo.py` file and change "file:///Users/username/Desktop/programming-historian/" to the path to the directory on your own computer.

- python-lessons6.zip zip sync (/assets/python-lessons6.zip)

Great! Now you're ready to move on to the next lesson (output-data-as-html-file).

About the authors

William J. Turkel is Professor of History at the University of Western Ontario.

Adam Crymble is a senior lecturer of digital history.

Suggested Citation

William J. Turkel and Adam Crymble, "Creating and Viewing HTML Files with Python," *The Programming Historian* 1 (2012), https://programminghistorian.org/en/lessons/creating-and-viewing-html-files-with-python.

This project is administered by ProgHist Limited, Company Number 12192946 (https://beta.companieshouse.gov.uk/company/12192946).

ISSN 2397-2068 (English) (/)

Hosted on GitHub (https://github.com/programminghistorian/jekyll)

ISSN 2517-5769 (Spanish) (/es)

ISSN 2631-9462 (French) (/fr)

Site last updated 01 April 2020 (https://github.com/programminghistorian/jekyll/commits/gh-pages)

RSS feed subscriptions (https://programminghistorian.org/feed.xml)

See page history (https://github.com/programminghistorian/jekyll/commits/gh-pages/en/lessons/creating-and-viewing-html-files-with-python.md)

Make a suggestion (/en/feedback)