

CS50's Introduction to Programming with Python

OpenCourseWare

Donate  (<https://cs50.harvard.edu/donate>)

David J. Malan (<https://cs.harvard.edu/malan/>)

malan@harvard.edu

 (<https://www.facebook.com/dmalan>)  (<https://github.com/dmalan>) 

(<https://www.instagram.com/davidjmalan/>)  (<https://www.linkedin.com/in/malan/>)

 (<https://www.reddit.com/user/davidjmalan>) 

(<https://www.threads.net/@davidjmalan>)  (<https://twitter.com/davidjmalan>)

CS50 Shirtificate

Suppose that you'd like to implement a CS50 "shirtificate," a PDF with an image of an [I took CS50](https://cs50.harvardshop.com/collections/print/products/i-took-cs50-unisex-t-shirt) (<https://cs50.harvardshop.com/collections/print/products/i-took-cs50-unisex-t-shirt>) t-shirt, [shirtificate.png](#), customized with a user's own name.

In a file called `shirtificate.py`, implement a program that prompts the user for their name and outputs, using [fpdf2](https://pypi.org/project/fpdf2/) (<https://pypi.org/project/fpdf2/>), a CS50 shirtificate in a file called `shirtificate.pdf` similar to [this one for John Harvard](#), with these specifications:

- The [orientation](https://py-pdf.github.io/fpdf2/PageFormatAndOrientation.html) (<https://py-pdf.github.io/fpdf2/PageFormatAndOrientation.html>) of the PDF should be Portrait.
- The [format](https://py-pdf.github.io/fpdf2/PageFormatAndOrientation.html) (<https://py-pdf.github.io/fpdf2/PageFormatAndOrientation.html>) of the PDF should be A4, which is 210mm wide by 297mm tall.

CS50 Shirtificate



- The top of the PDF should say “CS50 Shirtificate” as [text \(https://py-pdf.github.io/fpdf2/Text.html\)](https://py-pdf.github.io/fpdf2/Text.html), centered horizontally.
- The shirt’s [image \(https://py-pdf.github.io/fpdf2/Images.html\)](https://py-pdf.github.io/fpdf2/Images.html) should be centered horizontally.
- The user’s name should be on top of the shirt, in white [text \(https://py-pdf.github.io/fpdf2/TextStyling.html\)](https://py-pdf.github.io/fpdf2/TextStyling.html).

All other details we leave to you. You’re even welcome to add borders, colors, and [lines \(https://py-pdf.github.io/fpdf2/Shapes.html#lines\)](https://py-pdf.github.io/fpdf2/Shapes.html#lines). Your shirtificate needn’t match John Harvard’s precisely. And no need to wrap long names across multiple lines.

Before writing any code, do read through fpdf2’s [tutorial \(https://py-pdf.github.io/fpdf2/Tutorial.html\)](https://py-pdf.github.io/fpdf2/Tutorial.html) to learn how to use it. Then skim fpdf2’s [API \(https://py-pdf.github.io/fpdf2/fpdf/\)](https://py-pdf.github.io/fpdf2/fpdf/) (application programming interface) to see all of its functions and parameters therefor.

No need to submit any PDFs with your code. But, if you would like, you’re welcome (but not expected) to share a shirtificate with your name on it in any of [CS50’s communities \(https://cs50.harvard.edu/python/communities/\)](https://cs50.harvard.edu/python/communities/)!

▼ Hints

- Note that fpdf2 comes with a `class` called `FPDF`, which has quite a few methods, per [py-pdf.github.io/fpdf2/fpdf/#fpdf.FPDF \(https://py-pdf.github.io/fpdf2/fpdf/#fpdf.FPDF\)](https://py-pdf.github.io/fpdf2/fpdf/#fpdf.FPDF). You can install it with:

```
pip install fpdf2
```

- Note that you can extend `FPDF` and instantiate your own subclass thereof in order to add a header to every page of a PDF, per [py-pdf.github.io/fpdf2/Tutorial.html#tuto-2-header-footer-page-break-and-image \(https://py-pdf.github.io/fpdf2/Tutorial.html#tuto-2-header-footer-page-break-and-image\)](https://py-pdf.github.io/fpdf2/Tutorial.html#tuto-2-header-footer-page-break-and-image). Or you can add it as text yourself.
- Note that you can disable automatic page breaks, which might otherwise cause your PDF to overflow from one page to two, with `set_auto_page_break`, per [py-pdf.github.io/fpdf2/Margins.html \(https://py-pdf.github.io/fpdf2/Margins.html\)](https://py-pdf.github.io/fpdf2/Margins.html).
- Note that a [cell \(https://py-pdf.github.io/fpdf2/Text.html#cell\)](https://py-pdf.github.io/fpdf2/Text.html#cell)’s height can be negative, to move it upward.
- You can open `shirtificate.pdf`, once outputted, by clicking it in VS Code’s file explorer.

Demo

```
$ python shirtificate.py
Name: John Harvard
$
```

Recorded with [asciinema](#)

Before You Begin

Log into [cs50.dev \(https://cs50.dev/\)](https://cs50.dev), click on your terminal window, and execute `cd` by itself. You should find that your terminal window's prompt resembles the below:

```
$
```

Next execute

```
mkdir shirtificate
```

to make a folder called `shirtificate` in your codespace.

Then execute

```
cd shirtificate
```

to change directories into that folder. You should now see your terminal prompt as `shirtificate/ $`. Now execute

```
wget https://cs50.harvard.edu/python/2022/psets/8/shirtificate/shirtificate.
```

to get a copy of the `shirtificate.png` image for your certificate. Finally, execute

```
code shirtificate.py
```

to make a file called `shirtificate.py` where you'll write your program.

How to Test

Here's how to test your code manually:

- Run your program with `shirtificate.py`. Make sure your program prompts you for a name. Enter your own name and press Enter. Your program should create a file, `shirtificate.pdf`, containing the name you entered as input overlaid on a rendering of `shirtificate.png`.
- Try a few other names for good measure, too!

You can execute the below to check your code using `check50`, a program that CS50 will use to test your code when you submit. But be sure to test it yourself as well!

```
check50 cs50/problems/2022/python/shirtificate
```

Green smilies mean your program has passed a test! Red frownies will indicate your program output something unexpected. Visit the URL that `check50` outputs to see the input `check50` handed to your program, what output it expected, and what output your program actually gave.

How to Submit

In your terminal, execute the below to submit your work.

```
submit50 cs50/problems/2022/python/shirtificate
```

