

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>)

File Extensions

Even though Windows and macOS sometimes hide them, most files have file extensions (https://en.wikipedia.org/wiki/Filename_extension), a suffix that starts with a period (`.`) at the end of their name. For instance, file names for GIFs (<https://en.wikipedia.org/wiki/GIF>) end with `.gif`, and file names for JPEGs (<https://en.wikipedia.org/wiki/JPEG>) end with `.jpg` or `.jpeg`. When you double-click on a file to open it, your computer uses its file extension to determine which program to launch.

Web browsers, by contrast, rely on media types (https://en.wikipedia.org/wiki/Media_type), formerly known as MIME types, to determine how to display files that live on the web. When you download a file from a web server, that server sends an HTTP header (https://en.wikipedia.org/wiki/List_of_HTTP_header_fields), along with the file itself, indicating the file's media type. For instance, the media type for a GIF is `image/gif`, and the media type for a JPEG is `image/jpeg`. To determine the media type for a file, a web server typically looks at the file's extension, mapping one to the other.

See developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types (https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types) for common types.

In a file called `extensions.py`, implement a program that prompts the user for the name of a file and then outputs that file's media type if the file's name ends, case-insensitively, in any of these suffixes:

- `.gif`
- `.jpg`
- `.jpeg`
- `.png`
- `.pdf`
- `.txt`
- `.zip`

If the file's name ends with some other suffix or has no suffix at all, output `application/octet-stream` instead, which is a common default.

► Hints

Demo



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 extensions
```

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

Then execute

```
cd extensions
```

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

```
code extensions.py
```

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

How to Test

Here's how to test your code manually:

- Run your program with `python extensions.py`. Type `happy.jpg` and press Enter. Your program should output:

```
image/jpeg
```

- Run your program with `python extensions.py`. Type `document.pdf` and press Enter. Your program should output:

```
application/pdf
```

Be sure to test each of the other file formats, vary the casing of your input, and “accidentally” add spaces on either side of your input before pressing enter. Your program should behave as expected, case- and space-insensitively.

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/extensions
```

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/extensions
```