



# File Extensions

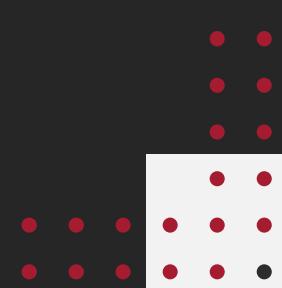
Problem set



با اینکه ویندوز و macOS بعضاً اوقات پسوند فایل‌ها یا file extensions را مخفی می‌کنند، ولی اکثر فایل‌ها extension در واقع یک پسوند است که بعد از ( . ) در پایان اسم فایل نمایان می‌شود.

برای مثال اسم فایل‌هایی که از نوع GIF هستند با gif. به اتمام می‌رسد و اسم فایل‌هایی که از نوع JPEG هستند با jpg. یا jpeg. به اتمام می‌رسد و نمایش داده می‌شود.

زمانی که شما روی یک فایل دوبار کلیک می‌کنید، کامپیوتر شما از این پسوندها (extensions) استفاده می‌کند تا نوع فایل را تشخیص دهد و از برنامه مناسب برای اجرای آن استفاده کند.



از طرفی، مرورگر های وب برای تعیین چگونگی نمایش فایل هایی که در وب قرار دارند، به media type ها، که به MIME هم معروف هستند، نیاز دارند. زمانی که یک فایل را از یک سرور وب دانلود می کنید، آن سرور وب همراه با فایل، یک HTTP header (اطلاعاتی که همراه یک درخواست وب ارسال می شود) که در آن media type فایل نشان داده شده را هم ارسال می کند. به عنوان مثال، media type برای یک فایل GIF به صورت image است و برای یک JPEG به شکل image/jpeg است. برای تعیین media type فایل، یک سرور وب معمولاً به پسوند فایل نگاه می کند و آن را با یکی از media type های موجود مطابقت می دهد.

برای مشاهده انواع رایج MIME ها، می توانید به لینک زیر مراجعه کنید:

[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)

در فایلی به نام extension.py، برنامه ای بنویسید که نام فایل را از کاربر گرفته و در صورتی که نام فایل با یکی از پسوند های زیر مطابقت داشت، media type در خروجی نمایش داده شود. دقیق شود که این مقایسه باید به صورت case-insensitive باشد (کوچک و بزرگ بودن حروف تفاوتی در نتیجه ندارد).

.gif

.jpg

.png

.txt

.pdf

.zip

.jpeg

اگر نام فایل وارد شده پسوند نداشت یا پسوند آن به غیر از موارد زیر بود، در خروجی application/octet-stream چاپ شود.

#### نکته ▼

- به یاد داشته باشید که string توابع مختلفی دارد. توضیحات بیشتر را در اینجا می‌توانید ببینید:

[docs.python.org/3/library/stdtypes.html#string-methods](https://docs.python.org/3/library/stdtypes.html#string-methods)

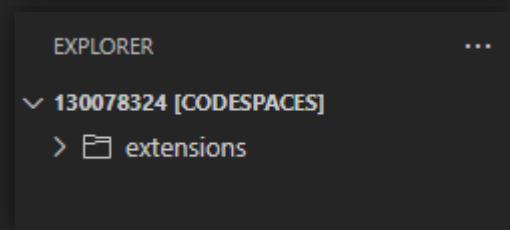


وارد [code.cs50.io](https://code.cs50.io) شوید، سپس بر روی پنجرهی Terminal کلیک کنید. توجه داشته باشید که دستور پنجرهی Terminal شما باید به صورت زیر باشد:

```
$
```

سپس کد زیر را اجرا کنید تا یک پوشه به نام extensions در codespace ایجاد شود:

```
$ mkdir extensions
```

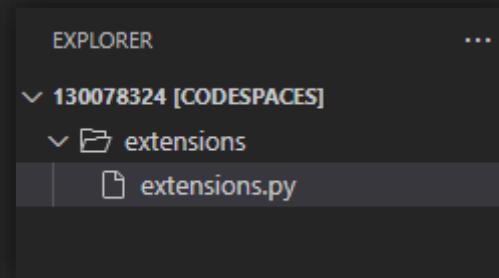


سپس `cd extensions` را اجرا کنید تا به پوشه‌ی `extensions` منتقل شوید. اکنون شما باید آدرس `$/extensions` را در Terminal مشاهده کنید:

```
$ cd extensions  
extensions/ $
```

حالا می‌توانید دستور `code extensions.py` را اجرا کنید تا فایلی با نام `extension.py` ایجاد شود:

```
extensions/ $ code extensions.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

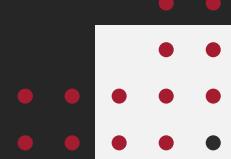
اکنون می‌خواهیم برنامه‌ی نوشته شده‌ی خود را تست کنیم:

- دستور `python extensions.py happy.jpg` را اجرا کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: `image/jpeg`
- دستور `python extensions.py document.pdf` را اجرا کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: `application/pdf`

- ❖ شما می‌توانید از آدرس زیر برای بررسی کردن کد خود استفاده کنید. CS50 از این برنامه برای آزمایش کد شما استفاده می‌کند. از این دستور استفاده کنید تا کدهایتان را امتحان کنید.

```
$ check50 cs50/problems/2022/python/extensions
```

لبخند های سبز به این معنی هستند که برنامه‌ی شما در تست قبول شده و اخم های قرمز یعنی برنامه‌ی شما دارای ایراد هست. با مراجعه به URL check50 می‌توانید خروجی مورد انتظار و خروجی برنامه‌ی خود را بررسی کنید.



با اجرا کردن صورت زیر در Terminal پاسخ خود را ارسال کنید.

```
$ submit50 cs50/problems/2022/python/extensions
```

# CS50x Iran

Harvard's Computer Science 50x Iran

