



Watch on YouTube

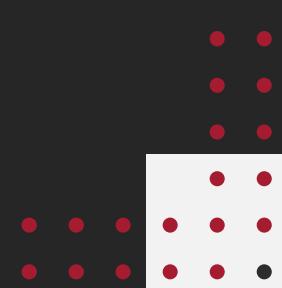
Problem set



اغلب ویدئوهای YouTube را می‌توان در وبسایت‌های دیگر جاسازی کرد، درست مانند [این ویدئو](#). برای مثال، اگر روی لپتاپ یا دسکتاپ تان این لینک (<https://youtu.be/xvFZjo5PgG0>) را نگاه کنید، و بعد روی گزینه share کلیک کنید و سپس روی Embed کلیک کنید، و کد HTML (زبانی که صفحات وب با آن نوشته می‌شوند) که به شما می‌دهد را در کدهای وبسایت خود کپی کنید.

در این تکه کد، element یا عنصر HTML است، و `src` یکی از چندین attributes یا صفات HTML است که مقدار آن بین علامت‌های نقل‌وقول "<https://www.youtube.com/embed/xvFZjo5PgG0>" قرار گرفته است.

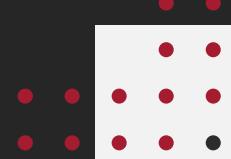
```
<iframe width="560" height="315"  
src="https://www.youtube.com/embed/xvFZjo5PgG0" title="YouTube video  
player" frameborder="0" allow="accelerometer; autoplay; clipboard-write;  
encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
```



از آنجایی که برخی از attributes یا صفات HTML اختیاری هستند، به جای آن می‌توانید نسخه کوتاه‌شده زیر را در کد خود وارد کنید:

```
<iframe src="https://www.youtube.com/embed/xvFZjo5PgG0"></iframe>
```

فرض کنید می‌خواهید آدرس‌های اینترنتی ویدیوهای YouTube را که در صفحات جاسازی شده‌اند را استخراج کنید، به عنوان مثال `https://www.youtube.com/embed/xvFZjo5PgG0`، و آنها را به آدرس‌های اینترنتی کوتاه‌تر و قابل اشتراک‌گذاری مشابه `https://youtu.be/xvFZjo5PgG0` تبدیل کنید، در واقع، به شکلی که می‌توان آنها را در YouTube تماشا کرد.



در فایلی به نام `watch.py`, تابعی به نام `parse` پیاده‌سازی کنید که یک `string` از HTML را به عنوان ورودی می‌گیرد، هر لینک یوتیوبی را که به صفت `src` در `element` یا عنصر `iframe` داده شده است را استخراج می‌کند و معادل کوتاه‌تر و قابل‌اشتراك آن را با دامنه `youtu.be` و به عنوان یک `string` برگرداند. انتظار می‌رود که هر URL در یکی از قالب‌های زیر باشد:

- <http://youtube.com/embed/xvFZjo5PgG0>
- <https://youtube.com/embed/xvFZjo5PgG0>
- <https://www.youtube.com/embed/xvFZjo5PgG0>

فرض کنید که مقدار `src` توسط دو علامت double quotes (" ") احاطه شده است و فرض کنید که ورودی حاوی بیش از یک URL نخواهد بود. اگر ورودی اصلًاً حاوی چنین URL‌ای نبود، تابع `None` را به عنوان خروجی بر می‌گرداند.



ساختار watch.py به شرح زیر است، که در آن شما می‌توانید هر طور که می‌خواهید تابع‌های دیگر را در `main` پیاده سازی کنید. اما دقت کنید که نمی‌توانید هیچ کتابخانه دیگری را `import` کنید. هر چند می‌توانید ولی نیازی نیست که از `sys` و `re` استفاده کنید.

```
import re
import sys

def main():
    print(parse(input("HTML: ")))

def parse(s):

    if __name__ == "__main__":
        main()
```

نکته ▼

- یادتان باشد که ماثول `re` توابعی مانند `search` دارد که می‌توانید در این لینک اطلاعات بیشتری در مورد آن کسب کنید.
- می‌توانید در Regular Expressions (یا regex) از کاراکتر های ویژه ای استفاده کنید. این لینک را مطالعه کنید.
- چون کاراکتر `\` در regex می‌تواند با escape sequences (برای مثال `\1` که اگر در متن استفاده شود، حرف بعد از آن به عنوان یک فرمان در نظر گرفته می‌شود. برای مثال `\n` که حرف `n` بعد از `\` به معنای new line بوده و هر جای متن استفاده شود، به معنای رفتن به خط بعد و شکستن متن است) اشتباه گرفته شوند، بهتر است از نماد رشته خام پایتون برای الگوهای regex استفاده کنید. مانند Format strings که با پیشوند `f` مشخص می‌شوند، رشته‌های خام نیز یک پیشوند `r` دارند. مثلاً به جای "جامعة Harvard" از `"harvard\.edu"` استفاده کنید.
- توجه داشته باشید که اگر `re.search` الگویی شامل "grouping capturing" (گروه های قابل استخراج) که با علائمی مثل پرانتز مشخص می‌شوند به عنوان ورودی دریافت کند، طبق (match object) عنصر مطابق با الگو را برمی‌گرداند، که در آن تطابق یافته ها (matches)، ایندکس ۱ هستند. که می‌توانید به آنها به صورت جداگانه با `group` و یا به صورت جمعی با `groups` دسترسی داشته باشید.
- توجه داشته باشید که `*` و `+` به طور "افراطی" عمل می‌کنند؛ به عبارت دیگر، بیشترین تعداد ممکن از متن سازگار با شرط را باز می‌گردانند، اضافه کردن علامت `?` بلافاصله پس از هر کدام از `*` یا `+`، مانند `?*` و `?+`، باعث می‌شود که تطبیق در حالت غیرافراطی یا حداقلی انجام شود و تنها تعدادی از کاراکترها با شرط سازگار بازگردانده می‌شود.

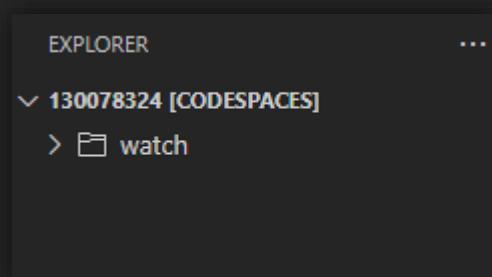


وارد code.cs50.io شوید، سپس بر روی پنجره Terminal کلیک کنید. توجه داشته باشید که دستور پنجره Terminal شما باید به صورت زیر باشد:

```
$
```

سپس کد زیر را اجرا کنید تا یک پوشه به نام watch ایجاد شود :

```
$ mkdir watch
```

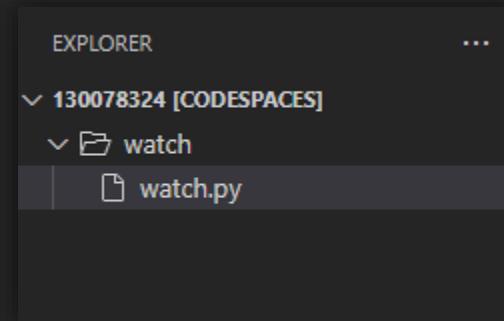


سپس `watch/` را اجرا کنید تا به پوشهی `watch` منتقل شوید. اکنون شما باید آدرس `$` را در Terminal مشاهده کنید:

```
$ cd watch  
watch/ $
```

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

```
watch/ $ code watch.py
```



اکنون میتوانید در این فایل برنامه‌ی خود را بنویسید.

- برنامه خود را با `python watch.py` اجرا کنید و اطمینان حاصل کنید که برنامه شما ورودی HTML، از شما درخواست کرده است، سپس متن زیر را کپی و درج کنید:

```
<iframe src="http://www.youtube.com/embed/xvFZjo5PgG0"></iframe>
```

کلید Enter را فشار دهید، سپس برنامه باید لینک <https://youtu.be/xvFZjo5PgG0> را به عنوان خروجی برگرداند. توجه کنید که با وجود اینکه ویژگی `src` با <http://www.youtube.com/embed/> شروع شده است، لینک حاصل با <https://youtu.be/> شروع می‌شود.

- برنامه خود را با `python watch.py` اجرا کنید و اطمینان حاصل کنید که برنامه شما ورودی HTML، از شما درخواست کرده است، سپس متن زیر را کپی و درج کنید. سپس کلید Enter را فشار دهید و خروجی برنامه شما برابر می‌شود با <https://youtu.be/xvFZjo5PgG0>.

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/xvFZjo5PgG0" title="YouTube video player"
frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-
media; gyroscope; picture-in-picture" allowfullscreen></iframe>
```

- برنامه خود را با `python watch.py` اجرا کنید و اطمینان حاصل کنید که برنامه شما برای ورودی HTML، از شما درخواست کرده است، سپس متن زیر را کپی و درج کنید:

```
<iframe width="560" height="315" src="https://cs50.harvard.edu/python"></iframe>
```

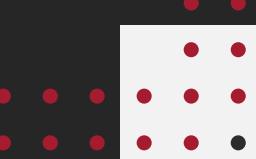
کلید Enter را فشار دهید، سپس برنامه باید None را به عنوان خروجی برگرداند. توجه کنید که attribute `src` به یک لینک یوتیوب اشاره نمی‌کند.



- ❖ 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/watch
```

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/watch
```

CS50x Iran

Harvard's Computer Science 50x Iran

