

Component: Application Wrappers

Pro Tip: Use iPython for Selenium Exploration

```
In [1]: from selenium import webdriver
```

```
In [2]: d = webdriver.Chrome()
```

```
In [3]: d.get('https://www.youtube.com')
```

```
In [4]: vid = '1JM90JmrBfU'
```

```
In [5]: sel_search = 'form input#search'
```

```
In [6]: search = d.find_element_by_css_selector(sel_search)
```

```
In [7]: search.click()
```

```
In [8]: search.send_keys(vid)
```

```
In [9]: search.submit()
```

```
In [10]: video = d.find_element_by_css_selector('a[href="/watch?v=%s"]' % vid)
```

```
In [11]: video.get_attribute('href')
```

```
Out[11]: 'https://www.youtube.com/watch?v=1JM90JmrBfU'
```


Component: Application Wrappers

The standard application wrapper is the Page Object Model, where pages are described in a class structure. The conceit is that these classes “know about themselves and their affordances”.

The goal is to abstract out the specifics of the interactions with these pages from the test cases to supporting code, making it easier to maintain the interface to the application under test.