

17 - Selenium

Roteiro

- Sobre o Selene
- Instalação
- Browser / WebDriver
- Navegação
- Interagindo com elementos
- Validações
- Nem tudo são flores



picpay.me/dunossauro | apoia.se/livedepython

PicPay

APOIA.se



picpay.me/dunossauro | apoia.se/livedepython

```
dunossauro at babbage in ~/git/apoiase on master*
```

```
$ python apoiadores.py
```

Acássio Araújo	Ademar Peixoto	Adão Oliveira	Alan Soder	Alberto Tavares	Alessandro Lopes
Alex Lima	Alexandre Tsuno	Alexandre Villares	Alexandre Santos	Alexandre Sá	Alexandre Harano
Alysson Oliveira	Amanda Magalhães	Amaziles Carvalho	Andre Rodrigues	Antonio Ribeiro	Bernardo Fontes
Bianca Rosa	Bruno Rocha	Bruno Oliveira	Caio Nascimento	Carlos Cardoso	Carlos Santos
Cleiton Souza	Davi Ramos	Davi Lima	Diego Guimarães	Dilenon Delfino	Douglas Bastos
Edson Braga	Eduardo Nunes	Eduardo Pazzi	Elias Soares	Emerson Lara	Eugenio Mazzini
Everton Alves	Fabício Coelho	Fernando Lanfranchi	Fernando Vier	Filipe Cruz	Flavkaze
Flávia Batista	Franklin Silva	Fábio Serrão	Gabriel Simonetto	Gabriela Santiago	Gabrielly Andrade
Geandreson Costa	Gladson Menezes	Hélio Neto	Isaac Ferreira	Israel Fabiano	Ivan Vicente
Johnny Tardin	Jonatas Leon	Jonatas Oliveira	Jonatas Oliveira	Jones Leite	Jones Lourenço
Jones Leite	Jorge Plautz	José Prado	Jovan Costa	João Lugão	Juan Gutierrez
Leonardo Cruz	Leonardo Galani	Lucas Mendes	Lucas Valino	Lucas Neris	Lucas Polo
Luciano Ratamero	Luiz Lima	Luiz Bruno	Maiquel Leonel	Marcela Campos	Marcello Benigno
Marcos Jesus	Maria Clara	Mateus Braga	Melisa Campagnaro	Otavio Carneiro	Patric Lacouth
Patrick Gomes	Paulo Tadei	Paulo Gonçalves	Rafael Dias	Rafael Peixoto	Reinaldo Silva
Renne Rocha	Ricardo Schalch	Rodrigo Vaccari	Rodrigo Ferreira	Régis Tomkiel	Samanta Cicilia
Thiago Araujo	Thiago Bueno	Tiago Cordeiro	Tyrone Damasceno	Valdir Junior	Vicente Marcal
Victor Orsolan	Victor Geraldo	Vinícius Ferreira	Vinícius Bastos	Vítor Zanoni	William Oliveira
Willian Lopes	Willian Rosa	Willian Lopes	Wladimir Falcão	Falta Você!	Falta Você!



Quick Start

Simply...

```
from selene.support.shared import browser
from selene import by, be, have

browser.open('https://google.com/ncr')
browser.element(by.name('q')).should(be.blank)\
    .type('selenium').press_enter()
browser.all('.srg .g').should(have.size(10))\
    .first.should(have.text('Selenium automates browsers'))
```

<https://github.com/yashaka/selene/>



O que é o Selene?

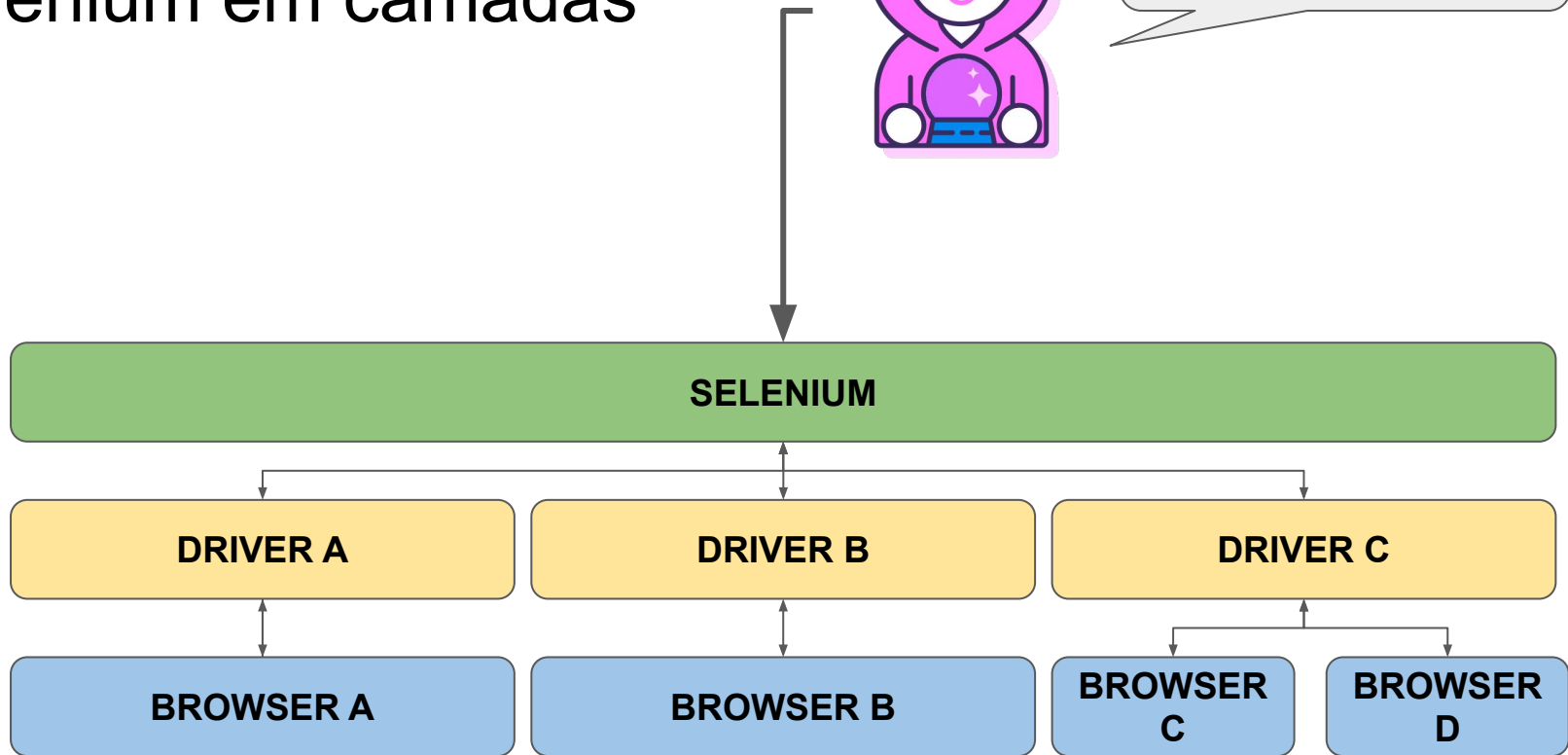
<https://github.com/yashaka/selene/>



Selenium em camadas



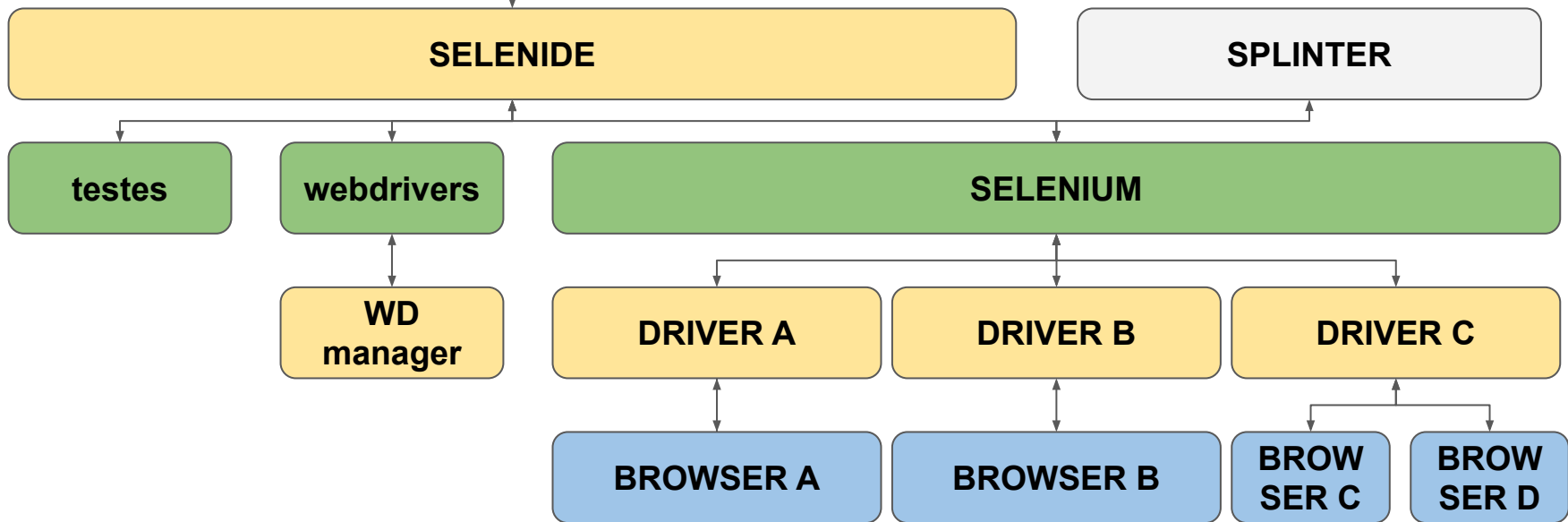
`browser.open('url')`



Selenium em camadas



`browser.open('url')`



Selene

O selene é baseado no Selenide, um wrapper do selenium feito do mundo java. Que não só automatiza tarefas, mas é um framework completo para testes.

Diferente do que vimos até agora com selenium original e o splinter. O Selene se propõem a resolver tanto as assertivas, quando a manipulação do browser e gerenciamento do webdriver.



Instalação



```
pip install selene
```

```
pip install selene --pre*
```



Browser / WebDriver



Manipulando o webdriver

A API do Selenium é incrível e não manipula somente o browser. Mas se preocupa com a instalação do webdriver também.

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

```
In [2]: browser.open('http://ddg.gg')
```

Manipulando o webdriver

A API do Selenium é incrível e não manipula somente o browser. Mas se preocupa com a instalação do webdriver também.

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

In [2]: browser.open('http://ddg.gg')
/bin/sh: google-chrome: comando não encontrado

[WDM] - Current google-chrome version is 84.0.4147
[WDM] - Get LATEST driver version for 84.0.4147
[WDM] - Driver [/home/dunossauro/.wdm/drivers/chromedriver/linux64/84.0.4147.30/chromedriver] found in cache
Out[2]: <selenium.support.shared.browser.SharedBrowser at 0x7f54974c4490>
```



Webdriver config



browser.config

No selene as configurações são compartilhadas e simples de serem feitas usando um objeto único.

```
from selene.support.shared import browser

url = 'http://selenium.dunossauro.live'

browser.config.browser_name = 'chrome'
browser.config.base_url = url

browser.open('/caixinha')
```



Driver Core API



Maneira “tradicional”

Quando for preciso especificar um browser não local, também podemos usar as formas tradicionais do selenium python. E ainda obter vantagens das configurações globais

```
from selene import Browser, Config
from selenium.webdriver import Firefox

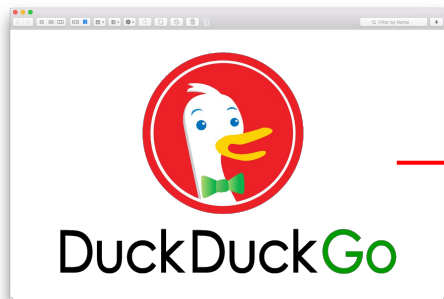
browser = Browser(
    Config(
        driver=Firefox(),
        base_url='https://google.com',
    )
)
```



Navegação



Navegação



Histórico	
pos	url
2	http://ddg.gg
1	http://redit.com
0	https://www.fsf.org/

```
>>> browser.open('http://fsf.org')
```

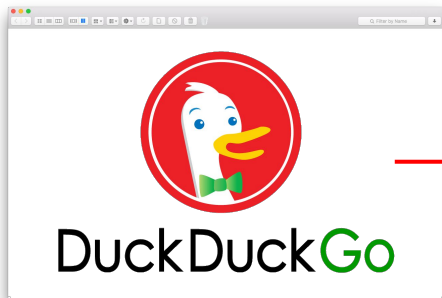
Notas sobre a próxima parte

O selene ainda não tem uma API completa do selenium.

Nesse caso não temos como ir para frente e para trás usando o wrapper.



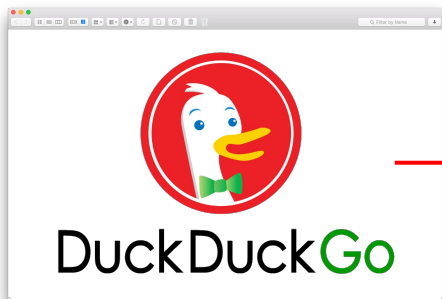
Navegação



Histórico	
pos	url
2	http://ddg.gg
1	http://redit.com
0	https://www.fsf.org/

```
>>> browser.open('http://fsf.org')  
>>> browser.driver.back()
```

Navegação



Histórico	
pos	url
2	http://ddg.gg
1	http://reddit.com
0	https://www.fsf.org/

```
>>> browser.open('http://fsf.org')  
>>> browser.driver.back()  
>>> browser.driver.forward()
```

Navegando com Selene

- `get()` -> `open()`
- `back()` -> `driver.back()`
- `forward()` -> `driver.forward()`
- `current_url` -> `driver.current_url`
- `title` -> `driver.title`
- `page_source` -> `driver.page_source`



Recursos legais de navegação

Embora os recursos primordiais estejam no driver e não no wrapper. Ainda podemos acessar a `config.base_url` que nos dá diversos confortos para trabalhar com várias páginas.

```
browser.open('')    #Vai para home  
browser.open('/caixinha')    #Vai para caixinha  
browser.open('/keyboard')    #Vai para teclado
```

Interagindo com elementos



Buscando elementos

Para buscar elementos com **selenium** podemos os encontrar usando 3 formas diferentes:

- Driver
- JQuery style
- Conectivos By



Buscando elementos

A forma mais simples, é usar o próprio wrapper. Ele nos permite usar css e xpath.

```
browser.element( ' [name="q"] ' )  
browser.element( ' //*[@name="q"] ' )
```

Buscando elementos

Podemos achar mais de um elemento usando o driver.

```
browser.element(' [name="q"] ' )
```

```
browser.all(' [name="q"] ' )
```

```
browser.element(' //*[@name="q"] ' )
```

```
browser.all(' //*[@name="q"] ' )
```

Buscando elementos - JQuery style

No estilo jquery podemos escrever expressões menores que também são compatíveis com css e xpath

```
browser.s( ' //*[@name="q"] ' )  
browser.ss( ' [name="q"] ' )
```

Buscando elementos

Ou usando o `by`

```
from selene import by
```

```
browser.element(' [name="q"] ')
```

```
browser.element(by.css(' [name="q"] '))
```

```
browser.element(' //*[@name="q"] ')
```

```
browser.element(by.xpath(' //*[@name="q"] '))
```



Buscando elementos

Ou usando o `by`

```
from selene import by
```

```
browser.element(' [name="q"] ')
```

```
browser.element(by.css(' [name="q"] '))
```

```
browser.element(' //*[@name="q"] ')
```

```
browser.element(by.xpath(' //*[@name="q"] '))
```



Procurando elementos

Selenium	Splinter	Selene
find_element_by_id()	find_by_id()	by.id()
find_elements_by_css_selector()	find_by_css()	by(), by.css()
find_elements_by_tag_name()	find_by_tag()	...
find_elements_by_xpath()	find_by_xpath()	by(), by.xpath()
find_elements_by_link_text()	...	by.link_text()
find_elements_by_partial_link_text()	...	by_partial_link_text()
find_elements_by_name()	find_by_name()	by.name()
...	find_by_text()	by.text()
...	...	by.partial_text()
...	find_by_value()	...



Validadores



Validadores

Temos alguns validadores incríveis no selene

- be
- have
- not



As chamadas

Antes disso, precisamos aprender a chamar os validadores

```
browser.open( '/aula_07' )  
  
browser.s( by.name( 'nome' ) ).click()  
  
browser.s( '[for="nome"]' ).should(  
    have.text( 'Não' )  
)
```

be	
be.visible	Verifica se elemento está visível
be.hidden	Verifica se elemento está oculto
be.selected	Verifica se elemento está selecionado
be.present	Verifica se elemento está presente (be.in_dom, be.existing)
be.absent	Verifica se elemento está não está presente
be.enabled	Verifica se elemento está habilitado
be.disabled	Verifica se elemento está desabilitado
be.clickable	Verifica se elemento está é clicável (botões)
be.blank	Verifica se elemento está em branco (inputs)

not	
not.visible	Verifica se elemento não está visível
not.hidden	Verifica se elemento não está oculto
not.selected	Verifica se elemento não está selecionado
not.present	Verifica se elemento não está presente (be.in_dom, be.existing)
not.absent	Verifica se elemento não está não está presente
not.enabled	Verifica se elemento não está habilitado
not.disabled	Verifica se elemento não está desabilitado
not.clickable	Verifica se elemento não está é clicável (botões)
not.blank	Verifica se elemento não está em branco (inputs)

be / not	
.visible	Verifica se elemento está, ou não, visível
.hidden	Verifica se elemento está, ou não, oculto
.selected	Verifica se elemento está, ou não, selecionado
.present	Verifica se elemento está, ou não, presente (be.in_dom, be.existing)
.absent	Verifica se elemento está, ou não, presente
.enabled	Verifica se elemento está, ou não, habilitado
.disabled	Verifica se elemento está, ou não, desabilitado
.clickable	Verifica se elemento está, ou não, clicável (botões)
.blank	Verifica se elemento está, ou não, branco (inputs)

have



have

Have, são condições especiais em que a validação vai além do selenium.



Propriedades



Propriedades do HTML / CSS

- `have.css_property(foo).value(bar)`
- `have.attribute(foo).value(bar)`
- `value`
- `value_containing`
- `css_class`
- `tag`
- `tag_containing`



Tamanhos



Tamanhos

- size
- size_less_than
- size_less_than_or_equal
- size_greater_than
- size_greater_than_or_equal



Tabs

Tabs

- `tabs_number`
- `tabs_number_less_than`
- `tabs_number_less_than_or_equal`
- `tabs_number_greater_than`
- `tabs_number_greater_than_or_equal`

Variados



Outras opções legais

- title
- title_containing
- url
- url_containing
- text
- texts



Nem tudo são flores



Nem tudo são flores

Sim, tá longe de ser perfeito

- Não tem documentação
- API em mudança constante
- Faz duas funções
- Lazy*



picpay.me/dunossauro | apoia.se/livedepython

PicPay

APOIA.se

