

Selenium Básico

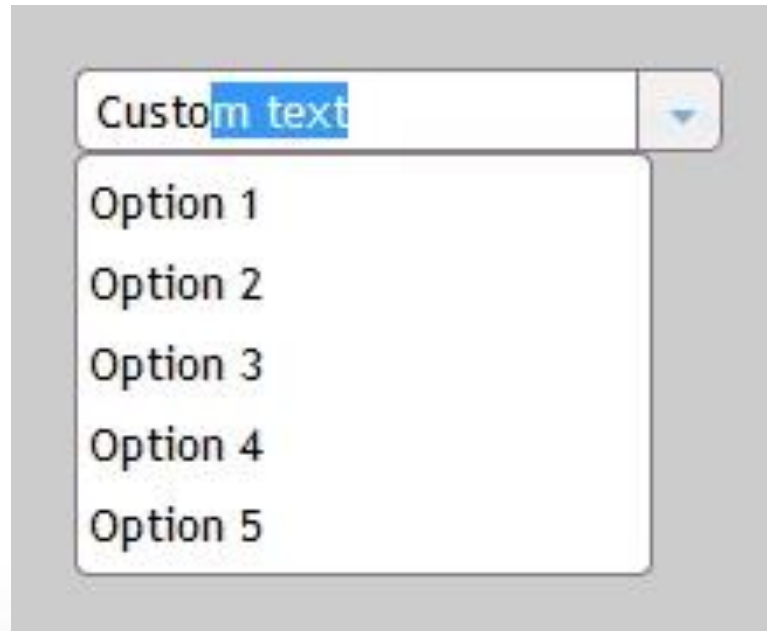
Select.

```
"""  
Example:  
from selenium.webdriver.support.ui import Select  
  
Select(driver.find_element_by_tag_name("select")).select  
by_index(2)  
"""
```

```
def select_by_index(self, index):  
    """Select the option at the given index.  
    This is done by examining the "index" attribute  
    of an  
        element, and not merely by counting.  
  
    :Args:  
        - index - The option at this index  
        will be selected  
  
        throws NoSuchElementException If there  
        is no option with specised index in SELECT  
    """  
    match = str(index)  
    for opt in self.options:  
        if opt.get_attribute("index") ==  
match:  
            self._setSelected(opt)  
  
        raise NoSuchElementException("Could not  
locate element with index %d" % index)
```

Autor: Reinaldo M. R. Junior

HTML combo box:



HTML Conteúdo combo box:

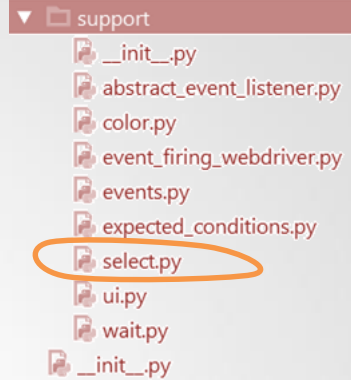
```
<!DOCTYPE html>
<html>
<body>

<select>
  <option value="1">Volvo</option>
  <option value="2">Saab</option>
  <option value="3">Opel</option>
  <option value="4">Audi</option>
</select>

</body>
</html>
```

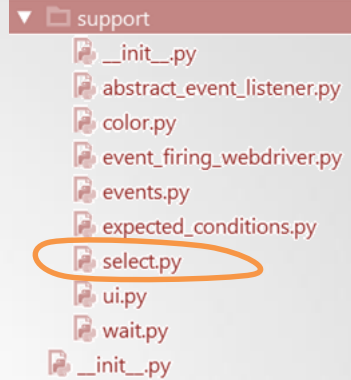
Nome Visível para o usuário.

Usando a classe Select.



- Instancia a classe Select passando o WebElement:
 - `select = Select(driver.find_element_by_css_selector("#identityMenu"))`
- Retorna uma lista de todas as opções que pertencem a tag select:
 - `select_options = select.options`
- A classe Select faz parte da biblioteca de support do Selenium -> *from selenium.webdriver.support.select import Select*

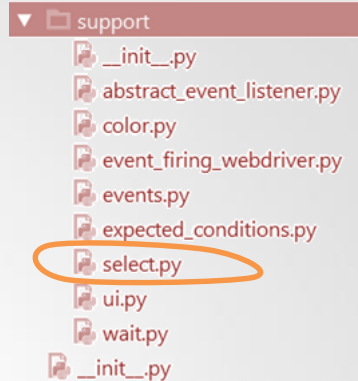
Usando a classe Select.



Opções da classe Select:

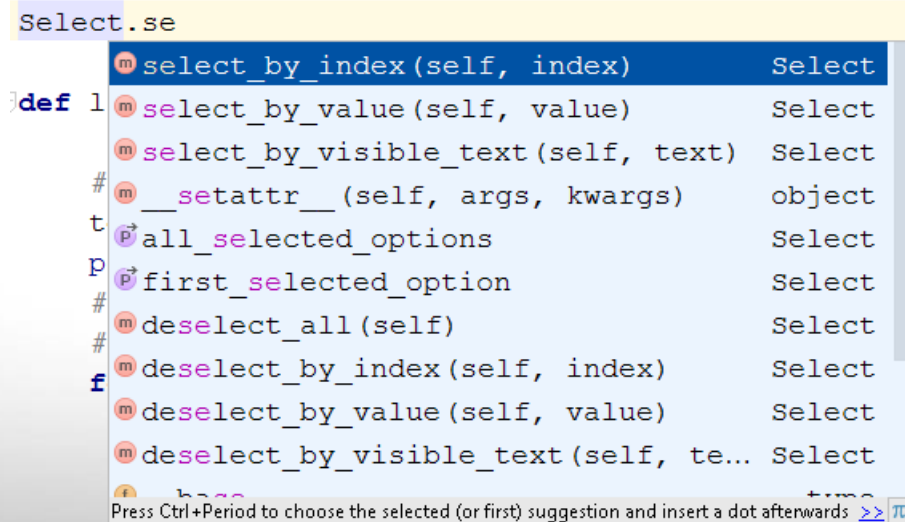
- **Options** - *Returns a list of all options belonging to this select tag.*
- **all_selected_options** - *Returns a list of all selected options belonging to this select tag.*
- **first_selected_option** - *The first selected option in this select tag.*
- **select_by_value** - *Select all options that have a value matching the argument.*
- **select_by_index** - *Select the option at the given index.*
- **select_by_visible_text** - *Select all options that display text matching the argument.*
- **deselect_all** - *Clear all selected entries.*
- **deselect_by_value** - *Deselect all options that have a value matching the argument.*
- **deselect_by_index** - *Deselect the option at the given index.*
- **deselect_by_visible_text** - *Deselect all options that display text matching the argument.*

Usando a classe Select.



Opções da classe Select via Pycharm:

- *from selenium.webdriver.support.select import Select*
- *Select.se*



Selecionando uma opção do combo box via índice.

- Tenho 4 elementos que começa pelo índice 0:
 - `select.select_by_index(0)`
 - `select.select_by_index(1)`
 - `select.select_by_index(2)`
 - `select.select_by_index(3)`
- Mostra o total de itens:
 - **`print (len(select_options))`**

Selecionando uma opção do combo box pelo nome.

- Função `select_by_visible_text`:
 - `select.select_by_visible_text("Volvo")`
 - `select.select_by_visible_text("Saab")`
 - `select.select_by_visible_text("Opel")`
 - `select.select_by_visible_text("Audi")`
 - Mostra todos os nomes:
 - `for x in range(len(select_options)):`
 `print (select_options[x].text)`
- Ou melhor:
- `for x, item in enumerate(select_options):`
 `print (x, item.get_attribute('text'))`

Selecionando uma opção do combo box pelo value.

- Função *select_by_value* :
 - `select.select_by_value("volvo")`
 - `select.select_by_value("saab")`
 - `select.select_by_value("opel")`
 - `select.select_by_value("audio")`
 - Mostra todos os valores, usando a função `get_attribute('value')`:
 - **`for x in range(len(select_options)):`**
`print (select_options[x].get_attribute('value'))`
- Ou melhor:
- **`for x, item in enumerate(select_options):`**
`print (x, item.get_attribute('value'))`

Após a Seleção de um combo box você pode deselegionar o mesmo.

- `deselect_all` – *Limpa todas as seleções.*
- `deselect_by_value` – *Limpa a seleção através do valor passado.*
- `deselect_by_index` - *Limpa a seleção através do índice passado.*
- `deselect_by_visible_text` - *Limpa a seleção através do visível texto passado.*
- Pycharm auto-complemento.

```
Select.dese  
def 1 m deselect_all(self) Select  
    m deselect_by_index(self, index) Select  
    m deselect_by_value(self, value) Select  
    # m deselect_by_visible_text(self, te... Select  
t Press Ctrl+Period to choose the selected (or first) suggestion and insert a dot afterwards >> π
```

Validando a Seleção.

Podemos verificar se o combo box foi selecionado com a função “**is_selected**”, ou fazer via JavaScript trazendo o texto selecionado.

```
from selenium import webdriver
from selenium.webdriver.support.select import Select

driver = webdriver.Firefox()
driver.get("http://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_select_options_selindex"
)

driver.switch_to.frame("iframeResult")
test = Select(driver.find_element_by_tag_name("select")).select_by_visible_text("Banana")
# Verificacao via javascript da opcao selecionada do combo box.
# Retorna o texto selecionado.
opcao_selecionado = driver.execute_script("var x = document.getElementById(\"mySelect\"); \
    var i = x.selectedIndex;\
    return x.options[i].text;")

print(opcao_selecionado)
# Forma comum de verificar a selecao pelo Selenium usando a funcao is_selected.
# No entanto ela retorna True ou False.
resultado = driver.find_element_by_css_selector("#mySelect > option:nth-child(4)").is_selected()
assert resultado
```

Podemos procurar o elemento e selecionar na mesma função.

- Nossa classe CheckBox em Helpers:

```
class CheckBox(object):

    def __set__(self, driver, attribute, value):

        test = ""

        if attribute == "index":
            test = Select(driver.find_element_by_tag_name("select")).select_by_index(value)
        if attribute == "value":
            test = Select(driver.find_element_by_tag_name("select")).select_by_value(value)
        if attribute == "text":
            test = Select(driver.find_element_by_tag_name("select")).select_by_visible_text(value)

        return test
```



Dificuldades

- Antes de mudar a opção do combobox-box tiver que desativar a seleção, a função `deselect_all` e `deselect_by_index`, não funcionou como esperado, foi mais fácil selecionar a primeira opção, que também nos enganou pois ela fica desativada após selecionar outra opção.
- `<option value="NONE" disabled="">Selecionar perfil...</option>`



Dificuldades

- A segunda dificuldade, em algumas opções do combo-box aparece um pop-up, que trava a tela, a opção de fechar pra alguns opções são diferentes, o que engana bastante.

```
▼ <div id="commentsHolder">  
  ▼ <div id="identityHolder" style="display: none;">  
    ▶ <div id="openIdIdentityHolder" class="dialog"></div>  
    ▶ <div id="genericOpenIdIdentityHolder" class="dialog"></div>  
    ▶ <div id="nameUrlIdentityHolder" class="dialog"></div>  
  </div>
```

Referências:

- [0] <http://selenium-python.readthedocs.io/api.html>
- [1] <https://github.com/SeleniumHQ/selenium/blob/master/py/selenium/webdriver/support/select.py>
- [2] http://www.w3schools.com/tags/tag_select.asp