

### Navigation & Windows

```
visit('url')

page.driver.browser.switch_to.window(window_handle)

my_original_win_handle = page.driver.browser.window_handles.first
my_last_win_handle = page.driver.browser.window_handles.last

page.driver.browser.switch_to.alert.accept

page.driver.go_back
```

### Clicking Links and Buttons

```
click_button('text, id, title, or value')

click_link('link id, link title or *link text')

click_on('link, button value, or button text')

page.driver.browser.action.move_to(page.all('p', text:
'Text').first.native).click.perform

*also see Querying
```

### Forms, Data and Typing

```
fill_in('name, id or text', with: 'value')

find('css_selector').set('value')

find(:xpath, 'xpath_selector').set('value')

select('text', from: 'select name, id or text ')

check('name, id or text of checkbox')

uncheck('name, id or text of checkbox')

choose('radio button name, id or text')

listbox-ex.native.sendkeys(:arrow_down)

attach_file('name, id or text, 'file/name.png')
```

### Scoping

```
within('css_locator')

within(:xpath, 'xpath_locator')

within_frame('name or id')

within_table('name or id')

within_fieldset('name or id')
```

### Querying

```
page.has_xpath?('//table/tr')

page.has_css?('table tr.foo')

page.has_content?('Text')

page.should have_xpath('//table/tr')

page.should have_css('table tr.foo')

page.should have_content('Text')

page.should have_no_content('Text')

find_field('Text').value

find_link('Text').visible?

find_button('Send').click

find('//table/tr').click

find_by_id(id)

locate("//*[@id='overlay']").find("//h1").click

all('a').each { |a| a[:href] }

expect(page or location).to have_content('Text')

expect(page).not_to have_content('Text')

expect(page).to have_css('a', text: 'Text', visible: true)
```

### Movement, Scrolling and Window Size

```
page.execute_script('window.scrollTo(0,-200)')

page.driver.browser.action.move_to(find_button(text).native).perform

page.driver.browser.manage.window.maximize

page.driver.browser.manage.window.resize_to(x,y)
```

### Page Source

```
expect page.html.downcase.include? 'text'

page.status_code.should be 200 (not supported by Selenium driver)

page.driver.browser.manage.all_cookies.each {|cookie|
```

### Scripting

```
result = page.evaluate_script('4 + 4');
```

### Debugging

```
save_and_open_page

also, use binding.pry
```

