



Navigation

> visit

visit navigates to a particular path. Pass a string or use one of Rails' path helpers.

```
visit "/blog"  
visit blogs_path
```

> click_link

click_link will click an anchor tag. Pass a string containing the anchor text.

```
click_link "Sign in"
```

Page Interaction and Scoping

> within

within will scope interaction to within a particular selector. Useful if you're looking for content in a particular area.

```
within("footer") { page.should have_content("Copyright") }
```

> has_content?

has_content? returns a boolean value reporting whether specific content is present on the page.

```
page.has_content?("Sign in")
```

> wait_until

wait_until executes a block until it returns true or raises a Timeout. This is the standard way to wait for Javascript interaction to complete. Works with Javascript drivers.

```
wait_until { page.has_content?("Data loaded!") }
```

Page Assertions

Note: All page assertions can be nested within `within` any number of times.

> have_content

have_content asserts that certain text is present on the page.

```
page.should have_content("What are you looking for?")
```

> have_css

have_css asserts that a certain selector is present on the page. have_css accepts CSS3 and is incredibly powerful.

```
page.should have_css("header")  
page.should have_css("table#records + .pagination a[rel='next']")
```

Node Interactions

> click

click triggers a click on a Capybara::Element. Works with Javascript drivers.

```
find("article a.title").click
```

> trigger

trigger allows triggering of custom events. Works with Javascript drivers.

```
find("input[name='post[title]']").trigger("focus")
```

> visible?

visible? returns a boolean value reporting if the Capybara::Element is visible. Works with Javascript drivers.

```
wait_until { find(".navigation").visible? }
```

Form Interactions

> fill_in

fill_in fills in fields for you. Pass the label text or the name of the input.

```
fill_in "Title", :with => "I love Cucumber!"  
fill_in "post[title]", :with => "I love Cucumber!"
```

> check

check checks a checkbox. Pass the label text.

```
check "I accept the terms of the site"  
check "I am thirteen years of age or older"
```

> uncheck

uncheck unchecks a checkbox. Pass the label text.

```
uncheck "Admin access?"
```

> select

select selects an option from a select tag.

```
select "Moderate", :from => "Political Party"  
select "MA", :from => "State"
```

> click_button

click_button will press a button or input[type='submit']

```
click_button "Create My Account"  
click_button "Save Record"
```

Debugging

> save_and_open_page

save_and_open_page will save the current page (typically to Rails.root/tmp) and attempt to open the html the default web browser.