

# Jekyll Living Documentation

# Table of Contents

<b>Summary</b>	1
<b>Features</b>	3
<b>Collections</b>	3
Scenario: Unrendered collection	3
Scenario: Rendered collection	3
Scenario: Rendered collection at a custom URL	4
Scenario: Rendered document in a layout	5
Scenario: Collections specified as an array	6
Scenario: Collections specified as an hash	7
Scenario: All the documents	8
Scenario: Documents have an output attribute, which is the converted HTML	9
Scenario: Filter documents by where	10
Scenario: Sort by title	11
Scenario: Sort by relative_path	12
<b>Configuring and using plugins</b>	13
Scenario: Add a gem-based plugin	13
Scenario: Add an empty whitelist to restrict all gems	14
Scenario: Add a whitelist to restrict some gems but allow others	15
<b>Create sites</b>	15
Scenario: Blank site	15
Scenario: Basic site	16
Scenario: Basic site with a post	16
Scenario: Basic site with layout and a page	17
Scenario: Basic site with layout and a post	17
Scenario: Basic site with layout inside a subfolder and a post	18
Scenario: Basic site with layouts, pages, posts and files	19
Scenario: Basic site with include tag	21
Scenario: Basic site with subdir include tag	21
Scenario: Basic site with nested include tag	22
Scenario: Basic site with internal post linking	23
Scenario: Basic site with whitelisted dotfile	24
Scenario: File was replaced by a directory	24
Scenario: Basic site with unpublished page	25
<b>Data</b>	26
Scenario: autoload *.yaml files in _data directory	27
Scenario: autoload *.yml files in _data directory	27
Scenario: autoload *.json files in _data directory	28
Scenario: autoload *.csv files in _data directory	29

Scenario: autoload *.yml files in _data directory with space in file name .....	30
Scenario: autoload *.yaml files in subdirectories in _data directory .....	31
Scenario: folders should have precedence over files with the same name .....	32
Scenario: should be backward compatible with site.data in _config.yml .....	33
<b>Draft Posts</b> .....	34
Scenario: Preview a draft .....	34
Scenario: Don't preview a draft. ....	35
Scenario: Don't preview a draft that is not published .....	36
Scenario: Use page.path variable .....	37
<b>Embed filters</b> .....	38
Scenario: Convert date to XML schema .....	38
Scenario: Escape text for XML .....	39
Scenario: Calculate number of words .....	40
Scenario: Convert an array into a sentence .....	41
Scenario: Markdownify a given string .....	42
Scenario: Sort by an arbitrary variable .....	43
Scenario: Sort pages by the title .....	44
Scenario: Sort pages by the title ordering pages without title last .....	45
<b>Fancy permalinks</b> .....	46
Scenario: Use none permalink schema .....	47
Scenario: Use pretty permalink schema .....	47
Scenario: Use pretty permalink schema for pages .....	48
Scenario: Use custom permalink schema with prefix .....	49
Scenario: Use custom permalink schema with category .....	50
Scenario: Use custom permalink schema with squished date .....	51
Scenario: Use per-post permalink .....	52
Scenario: Use per-post ending in .html .....	53
<b>Hooks</b> .....	54
Scenario: Run some code after site reset. ....	54
Scenario: Modify the payload before rendering the site .....	55
Scenario: Modify the site contents after reading .....	56
Scenario: Work with the site files after they've been written to disk .....	57
Scenario: Alter a page right after it is initialized .....	58
Scenario: Alter the payload for one page but not another .....	59
Scenario: Modify page contents before writing to disk .....	60
Scenario: Work with a page after writing it to disk .....	61
Scenario: Alter a post right after it is initialized .....	62
Scenario: Alter the payload for certain posts .....	62
Scenario: Modify post contents before writing to disk .....	63
Scenario: Work with a post after writing it to disk .....	64
Scenario: Register a hook on multiple owners at the same time .....	65

Scenario: Allow hooks to have a named priority .....	66
Scenario: Alter a document right after it is initialized .....	67
Scenario: Update a document after rendering it, but before writing it to disk .....	68
Scenario: Perform an action after every document is written .....	69
<b>Include tags</b> .....	70
Scenario: Include a file with parameters .....	71
Scenario: Include a file from a variable .....	73
Scenario: Include a variable file in a loop .....	74
Scenario: Include a file with variables and filters .....	75
Scenario: Include a file with partial variables.....	76
<b>Incremental rebuild</b> .....	77
Scenario: Produce correct output site .....	77
Scenario: Generate a metadata file.....	78
Scenario: Rebuild when content is changed.....	79
Scenario: Rebuild when layout is changed.....	79
Scenario: Rebuild when an include is changed.....	80
<b>Markdown</b> .....	81
Scenario: Markdown in list on index.....	82
Scenario: Markdown in pagination on index.....	82
<b>Post data</b> .....	83
Scenario: Use post.title variable .....	83
Scenario: Use post.url variable .....	84
Scenario: Use post.date variable .....	85
Scenario: Use post.id variable .....	86
Scenario: Use post.content variable .....	87
Scenario: Use post.categories variable when category is in a folder .....	88
Scenario: Use post.categories variable when category is in a folder and has category in YAML	89
Scenario: Use post.categories variable when category is in a folder and has categories in YAML	90
Scenario: Use post.categories variable when category is in a folder and duplicated category is in YAML	91
Scenario: Use post.tags variable .....	92
Scenario: Use post.categories variable when categories are in folders.....	93
Scenario: Use post.categories variable when categories are in folders with mixed case .....	94
Scenario: Use post.categories variable when category is in YAML.....	95
Scenario: Use post.categories variable when category is in YAML and is mixed-case .....	96
Scenario: Use post.categories variable when categories are in YAML.....	97
Scenario: Use post.categories variable when categories are in YAML and are duplicated ....	98
Scenario: Use post.categories variable when categories are in YAML with mixed case .....	99
Scenario Outline: Use page.path variable 🗨 .....	100
Scenario: Override page.path variable .....	101
Scenario: Disable a post from being published .....	101

Scenario: Use a custom variable .....	102
Scenario: Previous and next posts title .....	103
<b>Post excerpts</b> .....	104
Scenario: An excerpt without a layout .....	105
Scenario: An excerpt from a post with a layout .....	105
Scenario: An excerpt from a post with a layout which has context .....	107
<b>Rendering</b> .....	108
Scenario: Render Liquid and place in layout .....	108
Scenario: Don't place asset files in layout .....	108
Scenario: Render liquid in Sass .....	109
Scenario: Not render liquid in CoffeeScript without explicitly including jekyll-coffeescript ..	110
Scenario: Render liquid in CoffeeScript with jekyll-coffeescript enabled .....	110
<b>Site configuration</b> .....	110
Scenario: Change source directory .....	110
Scenario: Change destination directory .....	111
Scenario Outline: Similarly named source and destination 🐛 .....	111
Scenario: Exclude files inline .....	112
Scenario: Exclude files with YAML array .....	112
Scenario: Use RDiscount for markup .....	113
Scenario: Use Kramdown for markup .....	114
Scenario: Use Redcarpet for markup .....	114
Scenario: Highlight code with pygments .....	115
Scenario: Highlight code with rouge .....	115
Scenario: Rouge renders code block once .....	116
Scenario: Set time and no future dated posts .....	116
Scenario: Set time and future dated posts allowed .....	118
Scenario: Generate proper dates with explicitly set timezone (same as posts' time) .....	119
Scenario: Generate proper dates with explicitly set timezone (different than posts' time) ...	121
Scenario: Limit the number of posts generated by most recent date .....	122
Scenario: Copy over normally excluded files when they are explicitly included .....	123
Scenario: Using a different layouts directory .....	124
Scenario: arbitrary file reads via layouts .....	125
<b>Site data</b> .....	126
Scenario: Use page variable in a page .....	126
Scenario Outline: Use page.path variable in a page 🐛 .....	127
Scenario: Override page.path .....	127
Scenario: Use site.time variable .....	127
Scenario: Use site.posts variable for latest post .....	127
Scenario: Use site.posts variable in a loop .....	128
Scenario: Use site.categories.code variable .....	129
Scenario: Use site.tags variable .....	130

Scenario: Order Posts by name when on the same date .....	131
Scenario: Use configuration date in site payload .....	132
Scenario: Access Jekyll version via jekyll.version.....	133
<b>Site pagination</b> .....	133
Scenario Outline: Paginate with N posts per page 🐞 .....	133
Scenario Outline: Setting a custom pagination path 🐞 .....	133
Scenario Outline: Setting a custom pagination path without an index.html in it 🐞 .....	133
<b>frontmatter defaults</b> .....	134
Scenario: Use default for frontmatter variables internally .....	134
Scenario: Use default for frontmatter variables in Liquid .....	135
Scenario: Override frontmatter defaults by path .....	135
Scenario: Use frontmatter variables by relative path .....	137
Scenario: Override frontmatter defaults by type .....	138
Scenario: Actual frontmatter overrides defaults .....	139
Scenario: Define permalink default for posts .....	139
Scenario: Use frontmatter defaults in collections.....	140
Scenario: Override frontmatter defaults inside a collection .....	141
Scenario: Deep merge frontmatter defaults.....	142

# Summary

Scenarios			Steps							Features: 18	
Passed	Failed	Total	Passed	Failed	Skipped	Pending	Undefined	Missing	Total	Duration	Status
Collections											
11	0	11	75	0	0	0	0	0	75	06s 559ms	passed
Configuring and using plugins											
3	0	3	19	0	0	0	0	0	19	01s 788ms	passed
Create sites											
14	0	14	109	0	0	0	0	0	109	08s 838ms	passed
Data											
8	0	8	56	0	0	0	0	0	56	04s 524ms	passed
Draft Posts											
4	0	4	26	0	0	0	0	0	26	02s 292ms	passed
Embed filters											
8	0	8	59	0	0	0	0	0	59	04s 594ms	passed
Fancy permalinks											
8	0	8	51	0	0	0	0	0	51	05s 198ms	passed
Hooks											
17	0	17	114	0	0	0	0	0	114	09s 652ms	passed
Include tags											
5	0	5	50	0	0	0	0	0	50	03s 015ms	passed
Incremental rebuild											
5	0	5	44	0	0	0	0	0	44	07s 921ms	passed
Markdown											
2	0	2	17	0	0	0	0	0	17	01s 189ms	passed
Post data											
21	0	21	155	0	0	0	0	0	155	12s 729ms	passed
Post excerpts											

Scenarios			Steps							Features: 18	
3	0	3	32	0	0	0	0	0	32	01s 723ms	passed
Rendering											
5	0	5	27	0	0	0	0	0	27	03s 428ms	passed
Site configuration											
18	0	18	141	0	0	0	0	0	141	10s 312ms	passed
Site data											
10	0	10	51	0	0	0	0	0	51	05s 689ms	passed
Site pagination											
0	0	0	0	0	0	0	0	0	0	000ms	passed
frontmatter defaults											
10	0	10	85	0	0	0	0	0	85	05s 926ms	passed
Totals											
152	0	152	1111	0	0	0	0	0	1111	01m 35s 384ms	



# Features

## Collections

As a hacker who likes to structure content  
I want to be able to create collections of similar information  
And render them

### Scenario: Unrendered collection

*Given*

I have an "index.html" page that contains "Collections: {{ site.methods }}" 🍷 (000ms)

*And*

I have fixture collections 🍷 (000ms)

*And*

I have a configuration file with "collections" set to "['methods']" 🍷 (000ms)

*When*

I run jekyll build 🍷 (640ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

I should see "Collections: <p>Use <code>Jekyll.configuration</code> to build a full configuration for use w/Jekyll.</p>\n\n<p>Whatever: foo.bar</p>\n\n<p>Signs are nice</p>\n\n<p><code>Jekyll.sanitized\_path</code> is used to make sure your path is in your source.</p>\n\n<p>Run your generators! default</p>\n\n<p>Page without title.</p>\n\n<p>Run your generators! default</p>" in "\_site/index.html" 🍷 (000ms)

*And*

the "\_site/methods/configuration.html" file should not exist 🍷 (000ms)

### Scenario: Rendered collection

*Given*

I have an "index.html" page that contains "Collections: {{ site.collections }}" 🍷 (000ms)

*And*

I have an "collection\_metadata.html" page that contains "Methods metadata: {{ site.collections[0].foo }}" 🍷 (000ms)

*And*

I have fixture collections 🍷 (002ms)

*And*

I have a "\_config.yml" file with content: 🍷 (000ms)

```
collections:
  methods:
    output: true
    foo:    bar
```

*When*

I run jekyll build 🍷 (558ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

I should see "Collections: {"output"⇒true" in "\_site/index.html" 🍷 (000ms)

*And*

I should see "\"label\"⇒\"methods\"," in "\_site/index.html" 🍷 (000ms)

*And*

I should see "Methods metadata: bar" in "\_site/collection\_metadata.html" 🍷 (000ms)

*And*

I should see "<p>Whatever: foo.bar</p>" in "\_site/methods/configuration.html" 🍷 (000ms)

## Scenario: Rendered collection at a custom URL

*Given*

I have an "index.html" page that contains "Collections: {{ site.collections }}" 🍌 (000ms)

*And*

I have fixture collections 🍌 (001ms)

*And*

I have a "\_config.yml" file with content: 🍌 (000ms)

```
collections:
  methods:
    output: true
    permalink: /:collection/:path/
```

*When*

I run jekyll build 🍌 (547ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "<p>Whatever: foo.bar</p>" in "\_site/methods/configuration/index.html" 🍌 (000ms)

## Scenario: Rendered document in a layout

*Given*

I have an "index.html" page that contains "Collections: {{ site.collections }}" 🍷 (000ms)

*And*

I have a default layout that contains "<div class='title'>Tom Preston-Werner</div> {{content}}" 🍷 (000ms)

*And*

I have fixture collections 🍷 (001ms)

*And*

I have a "\_config.yml" file with content: 🍷 (000ms)

```
collections:
  methods:
    output: true
    foo:   bar
```

*When*

I run jekyll build 🍷 (547ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

I should see "Collections: {"output"⇒true" in "\_site/index.html" 🍷 (000ms)

*And*

I should see "\"label\"⇒\"methods\"," in "\_site/index.html" 🍷 (000ms)

*And*

I should see "<p>Run your generators! default</p>" in "\_site/methods/site/generate.html" 🍷 (000ms)

*And*

I should see "<div class='title'>Tom Preston-Werner</div>" in "\_site/methods/site/generate.html" 🍷 (000ms)

## Scenario: Collections specified as an array

*Given*

I have an "index.html" page that contains "Collections: {% for method in site.methods %}{{ method.relative\_path }} {% endfor %}" 🍌 (000ms)

*And*

I have fixture collections 🍌 (001ms)

*And*

I have a "\_config.yml" file with content: 🍌 (000ms)

```
collections:  
- methods
```

*When*

I run jekyll build 🍌 (647ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Collections: \_methods/configuration.md \_methods/escape-\+ %20\[].md \_methods/sanitized\_path.md \_methods/site/generate.md \_methods/site/initialize.md \_methods/um\_hi.md" in "\_site/index.html" 🍌 [small right](000ms)#

## Scenario: Collections specified as an hash

*Given*

I have an "index.html" page that contains "Collections: {% for method in site.methods %}{{ method.relative\_path }} {% endfor %}" 🍌 (000ms)

*And*

I have fixture collections 🍌 (001ms)

*And*

I have a "\_config.yml" file with content: 🍌 (000ms)

```
collections:  
- methods
```

*When*

I run jekyll build 🍌 (620ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Collections: \_methods/configuration.md \_methods/escape-\+ %20\[].md \_methods/sanitized\_path.md \_methods/site/generate.md \_methods/site/initialize.md \_methods/um\_hi.md" in "\_site/index.html" 🍌 [small right](000ms)#

## Scenario: All the documents

*Given*

I have an "index.html" page that contains "All documents: {% for doc in site.documents %}{{ doc.relative\_path }} {% endfor %}" 🍷 (000ms)

*And*

I have fixture collections 🍷 (002ms)

*And*

I have a "\_config.yml" file with content: 🍷 (000ms)

```
collections:  
- methods
```

*When*

I run jekyll build 🍷 (566ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

I should see "All documents: \_methods/configuration.md \_methods/escape-\+ %20\[].md \_methods/sanitized\_path.md \_methods/site/generate.md \_methods/site/initialize.md \_methods/um\_hi.md" in "\_site/index.html" 🍷 [small right](000ms)#

**Scenario: Documents have an output attribute, which is the converted HTML**

*Given*

I have an "index.html" page that contains "First document's output: {{ site.documents.first.output }}" 🍷 (000ms)

*And*

I have fixture collections 🍷 (001ms)

*And*

I have a "\_config.yml" file with content: 🍷 (000ms)

```
collections:  
- methods
```

*When*

I run jekyll build 🍷 (609ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

I should see "First document's output: <p>Use <code>Jekyll.configuration</code> to build a full configuration for use w/Jekyll.</p>\n\n<p>Whatever: foo.bar</p>" in "\_site/index.html" 🍷 (000ms)

## Scenario: Filter documents by where



*Given*

I have an "index.html" page that contains "{% assign items = site.methods | where: 'whatever','foo.bar' %}Item count: {{ items.size }}" 🍷 (000ms)

*And*

I have fixture collections 🍷 (002ms)

*And*

I have a "\_config.yml" file with content: 🍷 (000ms)

```
collections:  
- methods
```

*When*

I run jekyll build 🍷 (599ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

I should see "Item count: 2" in "\_site/index.html" 🍷 (000ms)

## Scenario: Sort by title

*Given*

I have an "index.html" page that contains "{% assign items = site.methods | sort: 'title' %}1. of {{ items.size }}: {{ items.first.output }}" 🍷 (000ms)

*And*

I have fixture collections 🍷 (001ms)

*And*

I have a "\_config.yml" file with content: 🍷 (000ms)

```
collections:  
- methods
```

*When*

I run jekyll build 🍷 (558ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

I should see "1. of 7: <p>Page without title.</p>" in "\_site/index.html" 🍷 (000ms)

## Scenario: Sort by relative\_path

*Given*

I have an "index.html" page that contains "Collections: {% assign methods = site.methods | sort: 'relative\_path' %}{% for method in methods %}{{ method.title }}{% endfor %}" 🍌 (000ms)

*And*

I have fixture collections 🍌 (000ms)

*And*

I have a "\_config.yml" file with content: 🍌 (000ms)

```
collections:
- methods
```

*When*

I run jekyll build 🍌 (634ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Collections: Jekyll.configuration, Jekyll.escape, Jekyll.sanitized\_path, Site#generate, , Site#generate," in "\_site/index.html" 🍌 (000ms)

## Configuring and using plugins

As a hacker

I want to specify my own plugins that can modify Jekyll's behaviour

### Scenario: Add a gem-based plugin

*Given*

I have an "index.html" file that contains "Whatever" 🍌 (000ms)

*And*

I have a configuration file with "gems" set to "[jekyll\_test\_plugin]" 🍌 (000ms)

*When*

I run jekyll build 🍌 (549ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Whatever" in "\_site/index.html" 🍌 (000ms)

*And*

I should see "this is a test" in "\_site/test.txt" 🍌 (000ms)

## Scenario: Add an empty whitelist to restrict all gems

*Given*

I have an "index.html" file that contains "Whatever" 🍌 (000ms)

*And*

I have a configuration file with: 🍌 (000ms)

key	value
gems	[jekyll_test_plugin]
whitelist	[]

*When*

I run jekyll build --safe 🍌 (579ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Whatever" in "\_site/index.html" 🍌 (000ms)

*And*

the "\_site/test.txt" file should not exist 🍌 (000ms)

## Scenario: Add a whitelist to restrict some gems but allow others

### Given

I have an "index.html" file that contains "Whatever" 🍌 (000ms)

### And

I have a configuration file with: 🍌 (000ms)

key	value
gems	[jekyll_test_plugin, jekyll_test_plugin_malicious]
whitelist	[jekyll_test_plugin]

### When

I run jekyll build --safe 🍌 (658ms)

### Then

the "\_site" directory should exist 🍌 (000ms)

### And

I should see "Whatever" in "\_site/index.html" 🍌 (000ms)

### And

the "\_site/test.txt" file should exist 🍌 (000ms)

### And

I should see "this is a test" in "\_site/test.txt" 🍌 (000ms)

## Create sites

As a hacker who likes to blog

I want to be able to make a static site

In order to share my awesome ideas with the interwebs

## Scenario: Blank site

*Given*

I do not have a "test\_blank" directory 🍷 (000ms)

*When*

I run `jeekyll new test_blank --blank` 🍷 (520ms)

*Then*

the `test_blank/_layouts` directory should exist 🍷 (000ms)

*And*

the `test_blank/_posts` directory should exist 🍷 (000ms)

*And*

the `"test_blank/index.html"` file should exist 🍷 (000ms)

## Scenario: Basic site

*Given*

I have an "index.html" file that contains "Basic Site" 🍷 (000ms)

*When*

I run `jeekyll build` 🍷 (517ms)

*Then*

the `_site` directory should exist 🍷 (000ms)

*And*

I should see "Basic Site" in `"_site/index.html"` 🍷 (000ms)

## Scenario: Basic site with a post

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (002ms)

title	date	content
Hackers	2009-03-27	My First Exploit

*When*

I run `jeekyll build` 🍌 (638ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "My First Exploit" in "`_site/2009/03/27/hackers.html`" 🍌 (000ms)

## Scenario: Basic site with layout and a page

*Given*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have an "index.html" page with layout "default" that contains "Basic Site with Layout" 🍌 (000ms)

*And*

I have a default layout that contains "Page Layout: {{ content }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (623ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Page Layout: Basic Site with Layout" in "`_site/index.html`" 🍌 (000ms)

## Scenario: Basic site with layout and a post

*Given*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
Wargames	2009-03-27	default	The only winning move is not to play.

*And*

I have a default layout that contains "Post Layout: {{ content }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (555ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post Layout: `<p>The only winning move is not to play.</p>`" in "`_site/2009/03/27/wargames.html`" 🍌 (000ms)

## Scenario: Basic site with layout inside a subfolder and a post



*Given*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
Wargames	2009-03-27	post/simple	The only winning move is not to play.

*And*

I have a `post/simple` layout that contains "Post Layout: `{{ content }}`" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (599ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post Layout: `<p>The only winning move is not to play.</p>`" in "`_site/2009/03/27/wargames.html`" 🍌 (000ms)

## Scenario: Basic site with layouts, pages, posts and files

*Given*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have a page layout that contains "Page `{{ page.title }}`: `{{ content }}`" 🍌 (000ms)

*And*

I have a post layout that contains "Post `{{ page.title }}`: `{{ content }}`" 🍌 (000ms)

*And*

I have an "index.html" page with layout "page" that contains "Site contains `{{ site.pages.size }}` pages and `{{ site.posts.size }}` posts" 🍌 (000ms)

*And*

I have a `blog` directory 🍌 (000ms)

*And*

I have a "blog/index.html" page with layout "page" that contains "blog category index page" 🍌 (000ms)

*And*

I have an "about.html" file that contains "No replacement {{ site.posts.size }}" 🍌 (000ms)

*And*

I have an "another\_file" file that contains "" 🍌 (000ms)

*And*

I have a \_posts directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
entry1	2009-03-27	post	content for entry1.
entry2	2009-04-27	post	content for entry2.

*And*

I have a category/\_posts directory 🍌 (000ms)

*And*

I have the following posts in "category": 🍌 (000ms)

title	date	layout	content
entry3	2009-05-27	post	content for entry3.
entry4	2009-06-27	post	content for entry4.

*When*

I run jekyll build 🍌 (658ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Page : Site contains 2 pages and 4 posts" in "\_site/index.html" 🍌 (000ms)

*And*

I should see "No replacement {{ site.posts.size }}" in "\_site/about.html" 🍌 (000ms)

*And*

I should see "" in "\_site/another\_file" 🍌 (000ms)

*And*

I should see "Page : blog category index page" in "\_site/blog/index.html" 🍌 (000ms)

*And*

I should see "Post entry1: <p>content for entry1.</p>" in "\_site/2009/03/27/entry1.html" 🍷 (000ms)

*And*

I should see "Post entry2: <p>content for entry2.</p>" in "\_site/2009/04/27/entry2.html" 🍷 (000ms)

*And*

I should see "Post entry3: <p>content for entry3.</p>" in "\_site/category/2009/05/27/entry3.html" 🍷 (000ms)

*And*

I should see "Post entry4: <p>content for entry4.</p>" in "\_site/category/2009/06/27/entry4.html" 🍷 (000ms)

## Scenario: Basic site with include tag

*Given*

I have a \_includes directory 🍷 (000ms)

*And*

I have an "index.html" page that contains "Basic Site with include tag: {% include about.textile %}" 🍷 (000ms)

*And*

I have an "\_includes/about.textile" file that contains "Generated by Jekyll" 🍷 (000ms)

*When*

I run jekyll build 🍷 (518ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

I should see "Basic Site with include tag: Generated by Jekyll" in "\_site/index.html" 🍷 (000ms)

## Scenario: Basic site with subdir include tag

*Given*

I have a `_includes` directory 🍌 (000ms)

*And*

I have an `"_includes/about.textile"` file that contains "Generated by Jekyll" 🍌 (000ms)

*And*

I have an `info` directory 🍌 (000ms)

*And*

I have an `"info/index.html"` page that contains "Basic Site with subdir include tag: {% include about.textile %}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (487ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Basic Site with subdir include tag: Generated by Jekyll" in `"_site/info/index.html"` 🍌 (000ms)

## Scenario: Basic site with nested include tag

*Given*

I have a `_includes` directory 🍌 (000ms)

*And*

I have an `"_includes/about.textile"` file that contains `"Generated by {% include jekyll.textile %}"` 🍌 (000ms)

*And*

I have an `"_includes/jekyll.textile"` file that contains `"Jekyll"` 🍌 (000ms)

*And*

I have an `"index.html"` page that contains `"Basic Site with include tag: {% include about.textile %}"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (534ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see `"Basic Site with include tag: Generated by Jekyll"` in `"_site/index.html"` 🍌 (000ms)

## Scenario: Basic site with internal post linking

*Given*

I have an "index.html" page that contains "URL: {% post\_url 2020-01-31-entry2 %}" 🍌 (000ms)

*And*

I have a configuration file with "permalink" set to "pretty" 🍌 (000ms)

*And*

I have a \_posts directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
entry1	2007-12-31	post	content for entry1.
entry2	2020-01-31	post	content for entry2.

*When*

I run jekyll build 🍌 (539ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "URL: /2020/01/31/entry2/" in "\_site/index.html" 🍌 (000ms)

## Scenario: Basic site with whitelisted dotfile

*Given*

I have an ".htaccess" file that contains "SomeDirective" 🍌 (000ms)

*When*

I run jekyll build 🍌 (541ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "SomeDirective" in "\_site/.htaccess" 🍌 (000ms)

## Scenario: File was replaced by a directory

*Given*

I have a "test" file that contains "some stuff" 🍷 (000ms)

*When*

I run jekyll build 🍷 (520ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*When*

I delete the file "test" 🍷 (000ms)

*Given*

I have a test directory 🍷 (000ms)

*And*

I have a "test/index.html" file that contains "some other stuff" 🍷 (000ms)

*When*

I run jekyll build 🍷 (482ms)

*Then*

the \_site/test directory should exist 🍷 (000ms)

*And*

I should see "some other stuff" in "\_site/test/index.html" 🍷 (000ms)

## Scenario: Basic site with unpublished page

*Given*

I have an "index.html" page with title "index" that contains "Published page" 🍌 (000ms)

*And*

I have a "public.html" page with published "true" that contains "Explicitly published page" 🍌 (000ms)

*And*

I have a "secret.html" page with published "false" that contains "Unpublished page" 🍌 (000ms)

*When*

I run jekyll build 🍌 (489ms)

*Then*

the "\_site" directory should exist 🍌 (000ms)

*And*

the "\_site/index.html" file should exist 🍌 (000ms)

*And*

the "\_site/public.html" file should exist 🍌 (000ms)

*But*

the "\_site/secret.html" file should not exist 🍌 (000ms)

*When*

I run jekyll build --unpublished 🍌 (589ms)

*Then*

the "\_site" directory should exist 🍌 (000ms)

*And*

the "\_site/index.html" file should exist 🍌 (000ms)

*And*

the "\_site/public.html" file should exist 🍌 (000ms)

*And*

the "\_site/secret.html" file should exist 🍌 (000ms)

## Data



In order to use well-formatted data in my blog  
As a blog's user  
I want to use `_data` directory in my site

## Scenario: autoload \*.yaml files in `_data` directory

*Given*

I have a `_data` directory 🍌 (000ms)

*And*

I have a "`_data/products.yaml`" file with content: 🍌 (000ms)

```
- name: sugar  
  price: 5.3  
- name: salt  
  price: 2.5
```

*And*

I have an "`index.html`" page that contains "{% for product in site.data.products %}{{product.name}}{% endfor %}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (588ms)

*Then*

the "`_site/index.html`" file should exist 🍌 (000ms)

*And*

I should see "sugar" in "`_site/index.html`" 🍌 (000ms)

*And*

I should see "salt" in "`_site/index.html`" 🍌 (000ms)

## Scenario: autoload \*.yml files in `_data` directory

*Given*

I have a `_data` directory 🍌 (000ms)

*And*

I have a `"_data/members.yml"` file with content: 🍌 (000ms)

```
- name: Jack  
  age: 28  
- name: Leon  
  age: 34
```

*And*

I have an `"index.html"` page that contains `"{% for member in site.data.members %}{{member.name}}{% endfor %}"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (552ms)

*Then*

the `"_site/index.html"` file should exist 🍌 (000ms)

*And*

I should see `"Jack"` in `"_site/index.html"` 🍌 (000ms)

*And*

I should see `"Leon"` in `"_site/index.html"` 🍌 (000ms)

## Scenario: autoload \*.json files in `_data` directory

*Given*

I have a `_data` directory 🍌 (000ms)

*And*

I have a `"_data/members.json"` file with content: 🍌 (000ms)

```
[{"name": "Jack", "age": 28}, {"name": "Leon", "age": 34}]
```

*And*

I have an `"index.html"` page that contains `"{% for member in site.data.members %}{{member.name}}{% endfor %}"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (510ms)

*Then*

the `"_site/index.html"` file should exist 🍌 (000ms)

*And*

I should see `"Jack"` in `"_site/index.html"` 🍌 (000ms)

*And*

I should see `"Leon"` in `"_site/index.html"` 🍌 (000ms)

## Scenario: autoload \*.csv files in `_data` directory

*Given*

I have a `_data` directory 🍌 (000ms)

*And*

I have a `"_data/members.csv"` file with content: 🍌 (000ms)

```
name,age
Jack,28
Leon,34
```

*And*

I have an `"index.html"` page that contains `"{% for member in site.data.members %}{{member.name}}{% endfor %}"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (503ms)

*Then*

the `"_site/index.html"` file should exist 🍌 (000ms)

*And*

I should see `"Jack"` in `"_site/index.html"` 🍌 (000ms)

*And*

I should see `"Leon"` in `"_site/index.html"` 🍌 (000ms)

**Scenario: autoload \*.yml files in `_data` directory with space in file name**

*Given*

I have a `_data` directory 🍌 (000ms)

*And*

I have a `"_data/team members.yml"` file with content: 🍌 (000ms)

```
- name: Jack  
  age: 28  
- name: Leon  
  age: 34
```

*And*

I have an `"index.html"` page that contains `"{% for member in site.data.team_members %}{{member.name}}{% endfor %}"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (555ms)

*Then*

the `"_site/index.html"` file should exist 🍌 (000ms)

*And*

I should see `"Jack"` in `"_site/index.html"` 🍌 (000ms)

*And*

I should see `"Leon"` in `"_site/index.html"` 🍌 (000ms)

**Scenario: autoload \*.yaml files in subdirectories in `_data` directory**

*Given*

I have a `_data` directory 🍌 (000ms)

*And*

I have a `_data/categories` directory 🍌 (000ms)

*And*

I have a `"_data/categories/dairy.yaml"` file with content: 🍌 (000ms)

```
name: Dairy Products
```

*And*

I have an `"index.html"` page that contains `"{{ site.data.categories.dairy.name }}"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (605ms)

*Then*

the `"_site/index.html"` file should exist 🍌 (000ms)

*And*

I should see `"Dairy Products"` in `"_site/index.html"` 🍌 (000ms)

**Scenario: folders should have precedence over files with the same name**

*Given*

I have a `_data` directory 🍌 (000ms)

*And*

I have a `_data/categories` directory 🍌 (000ms)

*And*

I have a `"_data/categories/dairy.yaml"` file with content: 🍌 (000ms)

```
name: Dairy Products
```

*And*

I have a `"_data/categories.yaml"` file with content: 🍌 (000ms)

```
dairy:  
  name: Should not display this
```

*And*

I have an `"index.html"` page that contains `"{{ site.data.categories.dairy.name }}"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (592ms)

*Then*

the `"_site/index.html"` file should exist 🍌 (000ms)

*And*

I should see `"Dairy Products"` in `"_site/index.html"` 🍌 (000ms)

**Scenario: should be backward compatible with site.data in `_config.yml`**

*Given*

I have a "\_config.yml" file with content: 🍌 (000ms)

```
data:
  - name: Jack
    age: 28
  - name: Leon
    age: 34
```

*And*

I have an "index.html" page that contains "{% for member in site.data %}{{member.name}}{% endfor %}" 🍌 (000ms)

*When*

I run jekyll build 🍌 (608ms)

*Then*

the "\_site/index.html" file should exist 🍌 (000ms)

*And*

I should see "Jack" in "\_site/index.html" 🍌 (000ms)

*And*

I should see "Leon" in "\_site/index.html" 🍌 (000ms)

## Draft Posts

As a hacker who likes to blog  
I want to be able to preview drafts locally  
In order to see if they look alright before publishing

### Scenario: Preview a draft



*Given*

I have a configuration file with "permalink" set to "none" 🍌 (000ms)

*And*

I have a \_drafts directory 🍌 (000ms)

*And*

I have the following draft: 🍌 (000ms)

title	date	layout	content
Recipe	2009-03-27	default	Not baked yet.

*When*

I run jekyll build --drafts 🍌 (547ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Not baked yet." in "\_site/recipe.html" 🍌 (000ms)

## Scenario: Don't preview a draft

*Given*

I have a configuration file with "permalink" set to "none" 🍌 (000ms)

*And*

I have an "index.html" page that contains "Totally index" 🍌 (000ms)

*And*

I have a \_drafts directory 🍌 (000ms)

*And*

I have the following draft: 🍌 (000ms)

title	date	layout	content
Recipe	2009-03-27	default	Not baked yet.

*When*

I run jekyll build 🍌 (509ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

the "\_site/recipe.html" file should not exist 🍌 (000ms)

## Scenario: Don't preview a draft that is not published

*Given*

I have a configuration file with "permalink" set to "none" 🍌 (000ms)

*And*

I have an "index.html" page that contains "Totally index" 🍌 (000ms)

*And*

I have a \_drafts directory 🍌 (000ms)

*And*

I have the following draft: 🍌 (000ms)

title	date	layout	published	content
Recipe	2009-03-27	default	false	Not baked yet.

*When*

I run jekyll build --drafts 🍌 (577ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

the "\_site/recipe.html" file should not exist 🍌 (000ms)

## Scenario: Use page.path variable

*Given*

I have a configuration file with "permalink" set to "none" 🍌 (000ms)

*And*

I have a \_drafts directory 🍌 (000ms)

*And*

I have the following draft: 🍌 (000ms)

title	date	layout	content
Recipe	2009-03-27	simple	Post path: {{ page.path }}

*When*

I run jekyll build --drafts 🍌 (652ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Post path: \_drafts/recipe.markdown" in "\_site/recipe.html" 🍌 (000ms)

## Embed filters

As a hacker who likes to blog

I want to be able to transform text inside a post or page

In order to perform cool stuff in my posts

## Scenario: Convert date to XML schema

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	content
Star Wars	2009-03-27	default	These aren't the droids you're looking for.

*And*

I have a default layout that contains "{{ site.time | date\_to\_xmlschema }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (578ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see today's date in "`_site/2009/03/27/star-wars.html`" 🍌 (000ms)

## Scenario: Escape text for XML

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	content
Star & Wars	2009-03-27	default	These aren't the droids you're looking for.

*And*

I have a default layout that contains "{{ page.title | xml\_escape }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (587ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Star & Wars" in "`_site/2009/03/27/star-wars.html`" 🍌 (000ms)

## Scenario: Calculate number of words

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	content
Star Wars	2009-03-27	default	These aren't the droids you're looking for.

*And*

I have a default layout that contains "{{ content | number\_of\_words }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (583ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "7" in "`_site/2009/03/27/star-wars.html`" 🍌 (000ms)

## Scenario: Convert an array into a sentence

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	tags	content
Star Wars	2009-03-27	default	[scifi, movies, force]	These aren't the droids you're looking for.

*And*

I have a default layout that contains "{{ page.tags | array\_to\_sentence\_string }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (555ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "scifi, movies, and force" in "`_site/2009/03/27/star-wars.html`" 🍌 (000ms)

## Scenario: Markdownify a given string



*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	content
Star Wars	2009-03-27	default	These aren't the droids you're looking for.

*And*

I have a default layout that contains "By {{ 'Obi-wan' | markdownify }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (603ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "By <p><em>Obi-wan</em></p>" in "`_site/2009/03/27/star-wars.html`" 🍌 (000ms)

## Scenario: Sort by an arbitrary variable

*Given*

I have a \_layouts directory 🍌 (000ms)

*And*

I have the following page: 🍌 (000ms)

title	layout	value	content
Page-1	default	8	Something

*And*

I have the following page: 🍌 (000ms)

title	layout	value	content
Page-2	default	6	Something

*And*

I have a default layout that contains "{{ site.pages | sort:'value' | map:'title' | join:', ' }}" 🍌 (000ms)

*When*

I run jekyll build 🍌 (568ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see exactly "Page-2, Page-1" in "\_site/page-1.html" 🍌 (000ms)

*And*

I should see exactly "Page-2, Page-1" in "\_site/page-2.html" 🍌 (000ms)

## Scenario: Sort pages by the title

*Given*

I have a \_layouts directory 🍌 (000ms)

*And*

I have the following page: 🍌 (000ms)

title	layout	content
Dog	default	Run

*And*

I have the following page: 🍌 (000ms)

title	layout	content
Bird	default	Fly

*And*

I have the following page: 🍌 (000ms)

layout	content
default	Jump

*And*

I have a default layout that contains "{% assign sorted\_pages = site.pages | sort: 'title' %}The rule of {{ sorted\_pages.size }}: {% for p in sorted\_pages %}{{ p.content | strip\_html | strip\_newlines }}{% endfor %}" 🍌 (000ms)

*When*

I run jekyll build 🍌 (561ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see exactly "The rule of 3: Jump, Fly, Run," in "\_site/bird.html" 🍌 (000ms)

## Scenario: Sort pages by the title ordering pages without title last

*Given*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following page: 🍌 (000ms)

title	layout	content
Dog	default	Run

*And*

I have the following page: 🍌 (000ms)

title	layout	content
Bird	default	Fly

*And*

I have the following page: 🍌 (000ms)

layout	content
default	Jump

*And*

I have a default layout that contains "{% assign sorted\_pages = site.pages | sort: 'title', 'last' %}The rule of {{ sorted\_pages.size }}: {% for p in sorted\_pages %}{{ p.content | strip\_html | strip\_newlines }}{% endfor %}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (543ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see exactly "The rule of 3: Fly, Run, Jump," in "`_site/bird.html`" 🍌 (000ms)

## Fancy permalinks

As a hacker who likes to blog

I want to be able to set permalinks

In order to make my blog URLs awesome

## Scenario: Use none permalink schema

*Given*

I have a \_posts directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	content
None Permalink Schema	2009-03-27	Totally nothing.

*And*

I have a configuration file with "permalink" set to "none" 🍌 (000ms)

*When*

I run jekyll build 🍌 (635ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Totally nothing." in "\_site/none-permalink-schema.html" 🍌 (000ms)

## Scenario: Use pretty permalink schema

*Given*

I have a \_posts directory 🍷 (000ms)

*And*

I have the following post: 🍷 (000ms)

title	date	content
Pretty Permalink Schema	2009-03-27	Totally wordpress.

*And*

I have a configuration file with "permalink" set to "pretty" 🍷 (000ms)

*When*

I run jekyll build 🍷 (684ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

I should see "Totally wordpress." in "\_site/2009/03/27/pretty-permalink-schema/index.html" 🍷 (000ms)

## Scenario: Use pretty permalink schema for pages

*Given*

I have an "index.html" page that contains "Totally index" 📄👍 (000ms)

*And*

I have an "awesome.html" page that contains "Totally awesome" 📄👍 (000ms)

*And*

I have an "sitemap.xml" page that contains "Totally uhm, sitemap" 📄👍 (000ms)

*And*

I have a configuration file with "permalink" set to "pretty" 📄👍 (000ms)

*When*

I run jekyll build 📄👍 (565ms)

*Then*

the \_site directory should exist 📄👍 (000ms)

*And*

I should see "Totally index" in "\_site/index.html" 📄👍 (000ms)

*And*

I should see "Totally awesome" in "\_site/awesome/index.html" 📄👍 (000ms)

*And*

I should see "Totally uhm, sitemap" in "\_site/sitemap.xml" 📄👍 (000ms)

## **Scenario: Use custom permalink schema with prefix**

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	category	date	content
Custom Permalink Schema	stuff	2009-03-27	Totally custom.

*And*

I have a configuration file with "permalink" set to `"/blog/:year/:month/:day/:title/"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (664ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Totally custom." in  `"_site/blog/2009/03/27/custom-permalink-schema/index.html"` 🍌 (000ms)

## Scenario: Use custom permalink schema with category



*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	category	date	content
Custom Permalink Schema	stuff	2009-03-27	Totally custom.

*And*

I have a configuration file with "permalink" set to `"/:categories/:title.html"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (673ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Totally custom." in `"_site/stuff/custom-permalink-schema.html"` 🍌 (000ms)

## Scenario: Use custom permalink schema with squished date

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	category	date	content
Custom Permalink Schema	stuff	2009-03-27	Totally custom.

*And*

I have a configuration file with "permalink" set to `"/:month-:day-:year/:title.html"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (637ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Totally custom." in `"_site/03-27-2009/custom-permalink-schema.html"` 🍌 (000ms)

## Scenario: Use per-post permalink

*Given*

I have a \_posts directory 🍷 (000ms)

*And*

I have the following post: 🍷 (000ms)

title	date	permalink	content
Some post	2013-04-14	/custom/posts/1/	bla bla

*When*

I run jekyll build 🍷 (656ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

the \_site/custom/posts/1 directory should exist 🍷 (000ms)

*And*

I should see "bla bla" in "\_site/custom/posts/1/index.html" 🍷 (000ms)

## Scenario: Use per-post ending in .html

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	permalink	content
Some post	2013-04-14	/custom/posts/some.html	bla bla

*When*

I run `jeekyll build` 🍌 (674ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

the `_site/custom/posts` directory should exist 🍌 (000ms)

*And*

I should see "bla bla" in "`_site/custom/posts/some.html`" 🍌 (000ms)

## Hooks

As a plugin author

I want to be able to run code during various stages of the build process

### Scenario: Run some code after site reset

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
Jekyll::Hooks.register :site, :after_reset do |site|
  pageklass = Class.new(Jekyll::Page) do
    def initialize(site, base)
      @site = site
      @base = base
      @data = {}
      @dir = '/'
      @name = 'foo.html'
      @content = 'mytinypage'

      self.process(@name)
    end
  end

  site.pages << pageklass.new(site, site.source)
end
```

*When*

I run `jeekyll build` 🍌 (513ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see `"mytinypage"` in `"_site/foo.html"` 🍌 (000ms)

## Scenario: Modify the payload before rendering the site

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `index.html` page that contains `"{{ site.injected }}"` 🍌 (000ms)

*And*

I have a `_plugins/ext.rb` file with content: 🍌 (000ms)

```
Jekyll::Hooks.register :site, :pre_render do |site, payload|
  payload['site']['injected'] = 'myparam'
end
```

*When*

I run `jeekyll build` 🍌 (558ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see `"myparam!"` in `_site/index.html` 🍌 (000ms)

## Scenario: Modify the site contents after reading

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a "page1.html" page that contains "page1" 🍌 (000ms)

*And*

I have a "page2.html" page that contains "page2" 🍌 (000ms)

*And*

I have a "`_plugins/ext.rb`" file with content: 🍌 (000ms)

```
Jekyll::Hooks.register :site, :post_read do |site|
  site.pages.delete_if { |p| p.name == 'page1.html' }
end
```

*When*

I run `jeekyll build` 🍌 (639ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

the "`_site/page1.html`" file should not exist 🍌 (000ms)

*And*

I should see "page2" in "`_site/page2.html`" 🍌 (000ms)

**Scenario: Work with the site files after they've been written to disk**

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
Jekyll::Hooks.register :site, :post_write do |site|
  firstpage = site.pages.first
  content = File.read firstpage.destination(site.dest)
  File.write(File.join(site.dest, 'firstpage.html'), content)
end
```

*And*

I have a `"page1.html"` page that contains `"page1"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (522ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see `"page1"` in `"_site/firstpage.html"` 🍌 (000ms)

## Scenario: Alter a page right after it is initialized



*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
Jekyll::Hooks.register :page, :post_init do |page|  
  page.name = 'renamed.html'  
  page.process(page.name)  
end
```

*And*

I have a `"page1.html"` page that contains `"page1"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (590ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see `"page1"` in `"_site/renamed.html"` 🍌 (000ms)

## Scenario: Alter the payload for one page but not another

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
Jekyll::Hooks.register :page, :pre_render do |page, payload|  
  payload['myparam'] = 'special' if page.name == 'page1.html'  
end
```

*And*

I have a `"page1.html"` page that contains `"{{ myparam }}"` 🍌 (000ms)

*And*

I have a `"page2.html"` page that contains `"{{ myparam }}"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (503ms)

*Then*

I should see `"special"` in `"_site/page1.html"` 🍌 (000ms)

*And*

I should not see `"special"` in `"_site/page2.html"` 🍌 (000ms)

## Scenario: Modify page contents before writing to disk

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a "index.html" page that contains "WRAP ME" 🍌 (000ms)

*And*

I have a "`_plugins/ext.rb`" file with content: 🍌 (000ms)

```
Jekyll::Hooks.register :page, :post_render do |page|
  page.output = "{{{{ {{page.output.chomp}} }}"
end
```

*When*

I run `jeekyll build` 🍌 (569ms)

*Then*

I should see "{{{{ WRAP ME }}" in "`_site/index.html`" 🍌 (000ms)

## Scenario: Work with a page after writing it to disk

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a "index.html" page that contains "HELLO FROM A PAGE" 🍌 (000ms)

*And*

I have a "`_plugins/ext.rb`" file with content: 🍌 (000ms)

```
Jekyll::Hooks.register :page, :post_write do |page|
  require 'fileutils'
  filename = page.destination(page.site.dest)
  FileUtils.mv(filename, "#{filename}.moved")
end
```

*When*

I run `jeekyll build` 🍌 (560ms)

*Then*

I should see "HELLO FROM A PAGE" in "`_site/index.html.moved`" 🍌 (000ms)

## Scenario: Alter a post right after it is initialized

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
# rot13 translate
Jekyll::Hooks.register :post, :post_init do |post|
  post.content.tr!('abcdefghijklmnopqrstuvwxyz',
    'nopqrstuvwxyzabcdefghijklmnopqrstuvwxyz')
end
```

*And*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
entry1	2015-03-14	nil	content for entry1.

*When*

I run `jeekyll build` 🍌 (531ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see `"pbagrag sbe ragel1."` in `"_site/2015/03/14/entry1.html"` 🍌 (000ms)

## Scenario: Alter the payload for certain posts

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
# Add myvar = 'old' to posts before 2015-03-15, and myvar = 'new' for
# others
Jekyll::Hooks.register :post, :pre_render do |post, payload|
  if post.date < Time.new(2015, 3, 15)
    payload['myvar'] = 'old'
  else
    payload['myvar'] = 'new'
  end
end
```

*And*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
entry1	2015-03-14	nil	{{ myvar }} post
entry2	2015-03-15	nil	{{ myvar }} post

*When*

I run `jeekyll build` 🍌 (534ms)

*Then*

I should see "old post" in `"_site/2015/03/14/entry1.html"` 🍌 (000ms)

*And*

I should see "new post" in `"_site/2015/03/15/entry2.html"` 🍌 (000ms)

## Scenario: Modify post contents before writing to disk

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
# Replace content after rendering
Jekyll::Hooks.register :post, :post_render do |post|
  post.output.gsub! /42/, 'the answer to life, the universe and everything'
end
```

*And*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (001ms)

title	date	layout	content
entry1	2015-03-14	nil	{{ 6
times: 7 }}	entry2	2015-03-15	nil

*When*

I run `jeekyll build` 🍌 (619ms)

*Then*

I should see "the answer to life, the universe and everything" in `"_site/2015/03/14/entry1.html"` 🍌 (000ms)

*And*

I should see "48" in `"_site/2015/03/15/entry2.html"` 🍌 (000ms)

## Scenario: Work with a post after writing it to disk

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
# Log all post filesystem writes
Jekyll::Hooks.register :post, :post_write do |post|
  filename = post.destination(post.site.dest)
  open('_site/post-build.log', 'a') do |f|
    f.puts "Wrote #{filename} at #{Time.now}"
  end
end
```

*And*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
entry1	2015-03-14	nil	entry one
entry2	2015-03-15	nil	entry two

*When*

I run `jeekyll build` 🍌 (549ms)

*Then*

I should see `"_site/2015/03/14/entry1.html"` at in `"_site/post-build.log"` 🍌 (000ms)

*Then*

I should see `"_site/2015/03/15/entry2.html"` at in `"_site/post-build.log"` 🍌 (000ms)

## Scenario: Register a hook on multiple owners at the same time

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
Jekyll::Hooks.register [:page, :post], :post_render do |owner|
  owner.output = "{{$ {{owner.output.chomp}} }}"
end
```

*And*

I have a `"index.html"` page that contains `"WRAP ME"` 🍌 (000ms)

*And*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
entry1	2015-03-14	nil	entry one

*When*

I run `jeekyll build` 🍌 (643ms)

*Then*

I should see `"{{$ {{ WRAP ME }} }}"` in `"_site/index.html"` 🍌 (000ms)

*And*

I should see `"{{$ {{ <p>entry one</p> }} }}"` in `"_site/2015/03/14/entry1.html"` 🍌 (000ms)

## Scenario: Allow hooks to have a named priority



*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
Jekyll::Hooks.register :page, :post_render, priority: :normal do |owner|
  # first normal runs second
  owner.output = "1 #{owner.output.chomp}"
end
Jekyll::Hooks.register :page, :post_render, priority: :high do |owner|
  # high runs last
  owner.output = "2 #{owner.output.chomp}"
end
Jekyll::Hooks.register :page, :post_render do |owner|
  # second normal runs third (normal is default)
  owner.output = "3 #{owner.output.chomp}"
end
Jekyll::Hooks.register :page, :post_render, priority: :low do |owner|
  # low runs first
  owner.output = "4 #{owner.output.chomp}"
end
```

*And*

I have a `"index.html"` page that contains `"WRAP ME"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (486ms)

*Then*

I should see `"2 3 1 4 WRAP ME"` in `"_site/index.html"` 🍌 (000ms)

## Scenario: Alter a document right after it is initialized

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
Jekyll::Hooks.register :document, :pre_render do |doc, payload|
  doc.data['text'] = doc.data['text'] << ' are belong to us'
end
```

*And*

I have a `"_config.yml"` file that contains `"collections: [ memes ]"` 🍌 (000ms)

*And*

I have a `_memes` directory 🍌 (000ms)

*And*

I have a `"_memes/doc1.md"` file with content: 🍌 (000ms)

```
---
text: all your base
---
```

*And*

I have an `"index.md"` file with content: 🍌 (000ms)

```
---
---
{{ site.memes.first.text }}
```

*When*

I run `jeekyll build` 🍌 (566ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see `"all your base are belong to us"` in `"_site/index.html"` 🍌 (000ms)

**Scenario: Update a document after rendering it, but before writing it to disk**

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
Jekyll::Hooks.register :document, :post_render do |doc|
  doc.output.gsub! /<p>/, '<p class="meme">'
end
```

*And*

I have a `"_config.yml"` file with content: 🍌 (000ms)

```
collections:
  memes:
    output: true
```

*And*

I have a `_memes` directory 🍌 (000ms)

*And*

I have a `"_memes/doc1.md"` file with content: 🍌 (000ms)

```
---
text: all your base are belong to us
---
{{ page.text }}
```

*When*

I run `jeekyll build` 🍌 (619ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see `"<p class=\"meme\">all your base are belong to us"` in `"_site/memes/doc1.html"` 🍌 (000ms)

**Scenario: Perform an action after every document is written**

*Given*

I have a `_plugins` directory 🍌 (000ms)

*And*

I have a `"_plugins/ext.rb"` file with content: 🍌 (000ms)

```
Jekyll::Hooks.register :document, :post_write do |doc|
  open('_site/document-build.log', 'a') do |f|
    f.puts "Wrote document #{doc.collection.docs.index doc} at #{Time.now}"
  end
end
```

*And*

I have a `"_config.yml"` file with content: 🍌 (000ms)

```
collections:
  memes:
    output: true
```

*And*

I have a `_memes` directory 🍌 (000ms)

*And*

I have a `"_memes/doc1.md"` file with content: 🍌 (000ms)

```
---
text: all your base are belong to us
---
{{ page.text }}
```

*When*

I run `jeekyll build` 🍌 (623ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Wrote document 0" in `"_site/document-build.log"` 🍌 (000ms)

## Include tags

In order to share their content across several pages  
As a hacker who likes to blog  
I want to be able to include files in my blog posts

## Scenario: Include a file with parameters

*Given*

I have an `_includes` directory 🍌 (000ms)

*And*

I have an `"_includes/header.html"` file that contains `"<header>My awesome blog header: {{include.param}}</header>"` 🍌 (000ms)

*And*

I have an `"_includes/params.html"` file that contains `"Parameters:<ul>{% for param in include %}<li>{{param[0]}} = {{param[1]}}</li>{% endfor %}</ul>"` 🍌 (000ms)

*And*

I have an `"_includes/ignore.html"` file that contains `"<footer>My blog footer</footer>"` 🍌 (000ms)

*And*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (002ms)

title	date	type	content
Include Files	2013-03-21	html	<code>{% include header.html param="myparam" %}</code>
Ignore params if unused	2013-03-21	html	<code>{% include ignore.html date="today" %}</code>
List multiple parameters	2013-03-21	html	<code>{% include params.html date="today" start="tomorrow" %}</code>
Dont keep parameters	2013-03-21	html	<code>{% include ignore.html param="test" %} {% include header.html %}</code>

title	date	type	content
Allow params with spaces and quotes	2013-04-07	html	{% include params.html cool="param with spaces" super="\quoted\"" single='has "quotes"' escaped='\single\' quotes' %}
Parameter syntax	2013-04-12	html	{% include params.html param1_or_2="value" %}
Pass a variable	2013-06-22	html	{% assign var = 'some text' %}{% include params.html local=var title=page.title %}

*When*

I run jekyll build 🍌 (606ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "<header>My awesome blog header: myparam</header>" in "\_site/2013/03/21/include-files.html" 🍌 (000ms)

*And*

I should not see "myparam" in "\_site/2013/03/21/ignore-params-if-unused.html" 🍌 (000ms)

*And*

I should see "<li>date = today</li>" in "\_site/2013/03/21/list-multiple-parameters.html" 🍌 (000ms)

*And*

I should see "<li>start = tomorrow</li>" in "\_site/2013/03/21/list-multiple-parameters.html" 🍌 (000ms)

*And*

I should not see "<header>My awesome blog header: myparam</header>" in "\_site/2013/03/21/dont-keep-parameters.html" 🍌 (000ms)

*But*

I should see "<header>My awesome blog header: </header>" in "\_site/2013/03/21/dont-keep-parameters.html" 🍌 (000ms)

*And*

I should see "<li>cool = param with spaces</li>" in "\_site/2013/04/07/allow-params-with-spaces-and-quotes.html" 🍌 (000ms)

*And*

I should see "<li>super = \"quoted\"</li>" in "\_site/2013/04/07/allow-params-with-spaces-and-quotes.html" 🍌 (000ms)

*And*

I should see "<li>single = has \"quotes\"</li>" in "\_site/2013/04/07/allow-params-with-spaces-and-quotes.html" 🍌 (000ms)

*And*

I should see "<li>escaped = 'single' quotes</li>" in "\_site/2013/04/07/allow-params-with-spaces-and-quotes.html" 🍌 (000ms)

*And*

I should see "<li>param1\_or\_2 = value</li>" in "\_site/2013/04/12/parameter-syntax.html" 🍌 (000ms)

*And*

I should see "<li>local = some text</li>" in "\_site/2013/06/22/pass-a-variable.html" 🍌 (000ms)

*And*

I should see "<li>title = Pass a variable</li>" in "\_site/2013/06/22/pass-a-variable.html" 🍌 (000ms)

## **Scenario: Include a file from a variable**

*Given*

I have an `_includes` directory 🍌 (000ms)

*And*

I have an `"_includes/snippet.html"` file that contains "a snippet" 🍌 (000ms)

*And*

I have an `"_includes/parametrized.html"` file that contains "works with `{{include.what}}`" 🍌 (000ms)

*And*

I have a configuration file with: 🍌 (000ms)

key	value
include_file1	snippet.html
include_file2	parametrized.html

*And*

I have an `"index.html"` page that contains `"{% include {{site.include_file1}} %} that {% include {{site.include_file2}} what='parameters' %}"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (702ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "a snippet that works with parameters" in `"_site/index.html"` 🍌 (000ms)

## Scenario: Include a variable file in a loop



*Given*

I have an `_includes` directory 🍌 (000ms)

*And*

I have an `_includes/one.html` file that contains "one" 🍌 (000ms)

*And*

I have an `_includes/two.html` file that contains "two" 🍌 (000ms)

*And*

I have an `index.html` page with files `[one.html, two.html]` that contains `"{% for file in page.files %}{% include {{file}} %} {% endfor %}"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (552ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "one two" in `_site/index.html` 🍌 (000ms)

## Scenario: Include a file with variables and filters

*Given*

I have an `_includes` directory 🍌 (000ms)

*And*

I have an `_includes/one.html` file that contains "one included" 🍌 (000ms)

*And*

I have a configuration file with: 🍌 (000ms)

key	value
include_file	one

*And*

I have an `index.html` page that contains "{% include {{ site.include\_file | append: '.html' }} %}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (542ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "one included" in `_site/index.html` 🍌 (000ms)

## Scenario: Include a file with partial variables

*Given*

I have an `_includes` directory 🍌 (000ms)

*And*

I have an `"_includes/one.html"` file that contains `"one included"` 🍌 (000ms)

*And*

I have a configuration file with: 🍌 (000ms)

key	value
include_file	one

*And*

I have an `"index.html"` page that contains `"{% include {{ site.include_file }}.html %}"` 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (600ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see `"one included"` in `"_site/index.html"` 🍌 (000ms)

## Incremental rebuild

As an impatient hacker who likes to blog  
I want to be able to make a static site  
Without waiting too long for it to build

### Scenario: Produce correct output site

*Given*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
Wargames	2009-03-27	default	The only winning move is not to play.

*And*

I have a default layout that contains "Post Layout: {{ content }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (622ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post Layout: `<p>The only winning move is not to play.</p>`" in "`_site/2009/03/27/wargames.html`" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (512ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post Layout: `<p>The only winning move is not to play.</p>`" in "`_site/2009/03/27/wargames.html`" 🍌 (000ms)

## Scenario: Generate a metadata file

*Given*

I have an "index.html" file that contains "Basic Site" 🍌 (000ms)

*When*

I run jekyll build 🍌 (499ms)

*Then*

the ".jekyll-metadata" file should exist 🍌 (000ms)

## Scenario: Rebuild when content is changed

*Given*

I have an "index.html" file that contains "Basic Site" 🍌 (000ms)

*When*

I run jekyll build 🍌 (481ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Basic Site" in "\_site/index.html" 🍌 (000ms)

*When*

I wait 1 second 🍌 (01s)

*Then*

I have an "index.html" file that contains "Bacon Site" 🍌 (000ms)

*When*

I run jekyll build 🍌 (576ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Bacon Site" in "\_site/index.html" 🍌 (000ms)

## Scenario: Rebuild when layout is changed

*Given*

I have a `_layouts` directory 🍷 (000ms)

*And*

I have an "index.html" page with layout "default" that contains "Basic Site with Layout" 🍷 (000ms)

*And*

I have a default layout that contains "Page Layout: {{ content }}" 🍷 (000ms)

*When*

I run `jeekyll build` 🍷 (487ms)

*Then*

the `_site` directory should exist 🍷 (000ms)

*And*

I should see "Page Layout: Basic Site with Layout" in "`_site/index.html`" 🍷 (000ms)

*When*

I wait 1 second 🍷 (01s)

*Then*

I have a default layout that contains "Page Layout Changed: {{ content }}" 🍷 (000ms)

*When*

I run `jeekyll build --full-rebuild` 🍷 (580ms)

*Then*

the `_site` directory should exist 🍷 (000ms)

*And*

I should see "Page Layout Changed: Basic Site with Layout" in "`_site/index.html`" 🍷 (000ms)

## Scenario: Rebuild when an include is changed

*Given*

I have a `_includes` directory 🍌 (000ms)

*And*

I have an "index.html" page that contains "Basic Site with include tag: {% include about.textile %}" 🍌 (000ms)

*And*

I have an "\_includes/about.textile" file that contains "Generated by Jekyll" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (593ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Basic Site with include tag: Generated by Jekyll" in "`_site/index.html`" 🍌 (000ms)

*When*

I wait 1 second 🍌 (01s)

*Then*

I have an "\_includes/about.textile" file that contains "Regenerated by Jekyll" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (560ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Basic Site with include tag: Regenerated by Jekyll" in "`_site/index.html`" 🍌 (000ms)

## Markdown

As a hacker who likes to blog

I want to be able to make a static site

In order to share my awesome ideas with the interwebs

## Scenario: Markdown in list on index

*Given*

I have a configuration file with "paginate" set to "5" 🍌 (000ms)

*And*

I have an "index.html" page that contains "Index - {% for post in site.posts %} {{ post.content }} {% endfor %}" 🍌 (000ms)

*And*

I have a \_posts directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	content	type
Hackers	2009-03-27	# My Title	markdown

*When*

I run jekyll build 🍌 (576ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Index" in "\_site/index.html" 🍌 (000ms)

*And*

I should see "<h1 id=\"my-title\">My Title</h1>" in "\_site/2009/03/27/hackers.html" 🍌 (000ms)

*And*

I should see "<h1 id=\"my-title\">My Title</h1>" in "\_site/index.html" 🍌 (000ms)

## Scenario: Markdown in pagination on index



*Given*

I have a configuration file with: 🍌 (000ms)

key	value
paginate	5
gems	[jekyll-paginate]

*And*

I have an "index.html" page that contains "Index - {% for post in paginator.posts %} {{ post.content }} {% endfor %}" 🍌 (000ms)

*And*

I have a \_posts directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	content	type
Hackers	2009-03-27	# My Title	markdown

*When*

I run jekyll build 🍌 (610ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Index" in "\_site/index.html" 🍌 (000ms)

*And*

I should see "<h1 id=\"my-title\">My Title</h1>" in "\_site/index.html" 🍌 (000ms)

## Post data

As a hacker who likes to blog

I want to be able to embed data into my posts

In order to make the posts slightly dynamic

### Scenario: Use post.title variable

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	content
Star Wars	2009-03-27	simple	Luke, I am your father.

*And*

I have a simple layout that contains "Post title: {{ page.title }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (664ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post title: Star Wars" in "`_site/2009/03/27/star-wars.html`" 🍌 (000ms)

## Scenario: Use `post.url` variable

*Given*

I have a `_posts` directory 🍇 (000ms)

*And*

I have a `_layouts` directory 🍇 (000ms)

*And*

I have the following post: 🍇 (000ms)

title	date	layout	content
Star Wars	2009-03-27	simple	Luke, I am your father.

*And*

I have a simple layout that contains "Post url: {{ page.url }}" 🍇 (000ms)

*When*

I run `jeekyll build` 🍇 (728ms)

*Then*

the `_site` directory should exist 🍇 (000ms)

*And*

I should see "Post url: `/2009/03/27/star-wars.html`" in "`_site/2009/03/27/star-wars.html`" 🍇 (000ms)

## Scenario: Use `post.date` variable

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	content
Star Wars	2009-03-27	simple	Luke, I am your father.

*And*

I have a simple layout that contains "Post date: {{ page.date | date\_to\_string }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (610ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post date: 27 Mar 2009" in "`_site/2009/03/27/star-wars.html`" 🍌 (000ms)

## Scenario: Use `post.id` variable

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	content
Star Wars	2009-03-27	simple	Luke, I am your father.

*And*

I have a simple layout that contains "Post id: {{ page.id }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (632ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post id: /2009/03/27/star-wars" in "`_site/2009/03/27/star-wars.html`" 🍌 (000ms)

## Scenario: Use `post.content` variable

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	content
Star Wars	2009-03-27	simple	Luke, I am your father.

*And*

I have a simple layout that contains "Post content: `{{ content }}`" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (605ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post content: `<p>Luke, I am your father.</p>`" in "`_site/2009/03/27/star-wars.html`" 🍌 (000ms)

**Scenario: Use `post.categories` variable when category is in a folder**

*Given*

I have a movies directory 🍌 (000ms)

*And*

I have a movies/\_posts directory 🍌 (000ms)

*And*

I have a \_layouts directory 🍌 (000ms)

*And*

I have the following post in "movies": 🍌 (000ms)

title	date	layout	content
Star Wars	2009-03-27	simple	Luke, I am your father.

*And*

I have a simple layout that contains "Post category: {{ page.categories }}" 🍌 (000ms)

*When*

I run jekyll build 🍌 (622ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Post category: movies" in "\_site/movies/2009/03/27/star-wars.html" 🍌 (000ms)

**Scenario: Use post.categories variable when category is in a folder and has category in YAML**

*Given*

I have a movies directory 🍌 (000ms)

*And*

I have a movies/\_posts directory 🍌 (000ms)

*And*

I have a \_layouts directory 🍌 (000ms)

*And*

I have the following post in "movies": 🍌 (000ms)

title	date	layout	category	content
Star Wars	2009-03-27	simple	film	Luke, I am your father.

*And*

I have a simple layout that contains "Post category: {{ page.categories }}" 🍌 (000ms)

*When*

I run jekyll build 🍌 (590ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Post category: movies" in "\_site/movies/film/2009/03/27/star-wars.html" 🍌 (000ms)

**Scenario: Use post.categories variable when category is in a folder and has categories in YAML**



*Given*

I have a movies directory 🍌 (000ms)

*And*

I have a movies/\_posts directory 🍌 (000ms)

*And*

I have a \_layouts directory 🍌 (000ms)

*And*

I have the following post in "movies": 🍌 (000ms)

title	date	layout	categories	content
Star Wars	2009-03-27	simple	[film, scifi]	Luke, I am your father.

*And*

I have a simple layout that contains "Post category: {{ page.categories }}" 🍌 (000ms)

*When*

I run jekyll build 🍌 (570ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Post category: movies" in "\_site/movies/film/scifi/2009/03/27/star-wars.html" 🍌 (000ms)

**Scenario: Use post.categories variable when category is in a folder and duplicated category is in YAML**

*Given*

I have a movies directory 🍌 (000ms)

*And*

I have a movies/\_posts directory 🍌 (000ms)

*And*

I have a \_layouts directory 🍌 (000ms)

*And*

I have the following post in "movies": 🍌 (000ms)

title	date	layout	category	content
Star Wars	2009-03-27	simple	movies	Luke, I am your father.

*And*

I have a simple layout that contains "Post category: {{ page.categories }}" 🍌 (000ms)

*When*

I run jekyll build 🍌 (627ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Post category: movies" in "\_site/movies/2009/03/27/star-wars.html" 🍌 (000ms)

## Scenario: Use post.tags variable

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	tag	content
Star Wars	2009-05-18	simple	twist	Luke, I am your father.

*And*

I have a simple layout that contains "Post tags: {{ page.tags }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (568ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post tags: twist" in "`_site/2009/05/18/star-wars.html`" 🍌 (000ms)

**Scenario: Use `post.categories` variable when categories are in folders**

*Given*

I have a scifi directory 🍌 (000ms)

*And*

I have a scifi/movies directory 🍌 (000ms)

*And*

I have a scifi/movies/\_posts directory 🍌 (000ms)

*And*

I have a \_layouts directory 🍌 (000ms)

*And*

I have the following post in "scifi/movies": 🍌 (000ms)

title	date	layout	content
Star Wars	2009-03-27	simple	Luke, I am your father.

*And*

I have a simple layout that contains "Post categories: {{ page.categories | array\_to\_sentence\_string }}" 🍌 (000ms)

*When*

I run jekyll build 🍌 (641ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Post categories: scifi and movies" in "\_site/scifi/movies/2009/03/27/star-wars.html" 🍌 (000ms)

**Scenario: Use post.categories variable when categories are in folders with mixed case**

*Given*

I have a scifi directory 🍌 (000ms)

*And*

I have a scifi/Movies directory 🍌 (000ms)

*And*

I have a scifi/Movies/\_posts directory 🍌 (000ms)

*And*

I have a \_layouts directory 🍌 (000ms)

*And*

I have the following post in "scifi/Movies": 🍌 (000ms)

title	date	layout	content
Star Wars	2009-03-27	simple	Luke, I am your father.

*And*

I have a simple layout that contains "Post categories: {{ page.categories | array\_to\_sentence\_string }}" 🍌 (000ms)

*When*

I run jekyll build 🍌 (609ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Post categories: scifi and Movies" in "\_site/scifi/movies/2009/03/27/star-wars.html" 🍌 (000ms)

**Scenario: Use post.categories variable when category is in YAML**

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	category	content
Star Wars	2009-03-27	simple	movies	Luke, I am your father.

*And*

I have a simple layout that contains "Post category: {{ page.categories }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (560ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post category: movies" in "`_site/movies/2009/03/27/star-wars.html`" 🍌 (000ms)

**Scenario: Use `post.categories` variable when category is in YAML and is mixed-case**

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	category	content
Star Wars	2009-03-27	simple	Movies	Luke, I am your father.

*And*

I have a simple layout that contains "Post category: {{ page.categories }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (583ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post category: Movies" in "`_site/movies/2009/03/27/star-wars.html`" 🍌 (000ms)

**Scenario: Use `post.categories` variable when categories are in YAML**

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	categories	content
Star Wars	2009-03-27	simple	['scifi', 'movies']	Luke, I am your father.

*And*

I have a simple layout that contains "Post categories: {{ page.categories | array\_to\_sentence\_string }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (589ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post categories: scifi and movies" in "`_site/scifi/movies/2009/03/27/star-wars.html`" 🍌 (000ms)

**Scenario: Use `post.categories` variable when categories are in YAML and are duplicated**



*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	categories	content
Star Wars	2009-03-27	simple	['movies', 'movies']	Luke, I am your father.

*And*

I have a simple layout that contains "Post category: {{ page.categories }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (641ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post category: movies" in "`_site/movies/2009/03/27/star-wars.html`" 🍌 (000ms)

**Scenario: Use `post.categories` variable when categories are in YAML with mixed case**

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (001ms)

title	date	layout	categories	content
Star Wars	2009-03-27	simple	['scifi', 'Movies']	Luke, I am your father.
Star Trek	2013-03-17	simple	['SciFi', 'movies']	Jean Luc, I am your father.

*And*

I have a simple layout that contains "Post categories: {{ page.categories | array\_to\_sentence\_string }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (550ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post categories: scifi and Movies" in "`_site/scifi/movies/2009/03/27/star-wars.html`" 🍌 (000ms)

*And*

I should see "Post categories: SciFi and movies" in "`_site/scifi/movies/2013/03/17/star-trek.html`" 🍌 (000ms)

## Scenario Outline: Use `page.path` variable 🍌

Table 1. Example

dir	path_prefix
.	
dir	dir/
dir/nested	dir/nested/

## Scenario: Override page.path variable

### Given

I have a \_posts directory 🍌 (000ms)

### And

I have the following post: 🍌 (000ms)

title	date	path	content
override	2013-04-12	override-path.html	Custom path: {{ page.path }}

### When

I run jekyll build 🍌 (577ms)

### Then

the \_site directory should exist 🍌 (000ms)

### And

I should see "Custom path: override-path.html" in "\_site/2013/04/12/override.html" 🍌 (000ms)

## Scenario: Disable a post from being published

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have an `"index.html"` file that contains `"Published!"` 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	published	content
Star Wars	2009-03-27	simple	false	Luke, I am your father.

*When*

I run `jeekyll build` 🍌 (607ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

the `"_site/2009/03/27/star-wars.html"` file should not exist 🍌 (000ms)

*And*

I should see `"Published!"` in `"_site/index.html"` 🍌 (000ms)

## Scenario: Use a custom variable

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	layout	author	content
Star Wars	2009-03-27	simple	Darth Vader	Luke, I am your father.

*And*

I have a simple layout that contains "Post author: {{ page.author }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (549ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Post author: Darth Vader" in "`_site/2009/03/27/star-wars.html`" 🍌 (000ms)

## Scenario: Previous and next posts title

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (001ms)

title	date	layout	author	content
Star Wars	2009-03-27	ordered	Darth Vader	Luke, I am your father.
Some like it hot	2009-04-27	ordered	Osgood	Nobody is perfect.
Terminator	2009-05-27	ordered	Arnold	Sayonara, baby

*And*

I have a ordered layout that contains "Previous post: {{ page.previous.title }} and next post: {{ page.next.title }}" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (570ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "next post: Some like it hot" in "`_site/2009/03/27/star-wars.html`" 🍌 (000ms)

*And*

I should see "Previous post: Some like it hot" in "`_site/2009/05/27/terminator.html`" 🍌 (000ms)

## Post excerpts

As a hacker who likes to blog

I want to be able to make a static site

In order to share my awesome ideas with the interwebs

But some people can only focus for a few moments

So just give them a taste

## Scenario: An excerpt without a layout

### Given

I have an "index.html" page that contains "{% for post in site.posts %}{{ post.excerpt }}{% endfor %}" 🍷 (000ms)

### And

I have a \_posts directory 🍷 (000ms)

### And

I have the following posts: 🍷 (000ms)

title	date	layout	content
entry1	2007-12-31	post	content for entry1.

### When

I run jekyll build 🍷 (627ms)

### Then

the \_site directory should exist 🍷 (000ms)

### And

I should see exactly "<p>content for entry1.</p>" in "\_site/index.html" 🍷 (000ms)

## Scenario: An excerpt from a post with a layout

*Given*

I have an "index.html" page that contains "{% for post in site.posts %}{{ post.excerpt }}{% endfor %}" 🍌 (000ms)

*And*

I have a \_posts directory 🍌 (000ms)

*And*

I have a \_layouts directory 🍌 (000ms)

*And*

I have a post layout that contains "{{ page.excerpt }}" 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
entry1	2007-12-31	post	content for entry1.

*When*

I run jekyll build 🍌 (552ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

the \_site/2007 directory should exist 🍌 (000ms)

*And*

the \_site/2007/12 directory should exist 🍌 (000ms)

*And*

the \_site/2007/12/31 directory should exist 🍌 (000ms)

*And*

the "\_site/2007/12/31/entry1.html" file should exist 🍌 (000ms)

*And*

I should see exactly "<p>content for entry1.</p>" in "\_site/2007/12/31/entry1.html" 🍌 (000ms)

*And*

I should see exactly "<p>content for entry1.</p>" in "\_site/index.html" 🍌 (000ms)



## Scenario: An excerpt from a post with a layout which has context

*Given*

I have an "index.html" page that contains "{% for post in site.posts %}{{ post.excerpt }}{% endfor %}" 🍌 (000ms)

*And*

I have a \_posts directory 🍌 (000ms)

*And*

I have a \_layouts directory 🍌 (000ms)

*And*

I have a post layout that contains "<html><head></head><body>{{ page.excerpt }}</body></html>" 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
entry1	2007-12-31	post	content for entry1.

*When*

I run jekyll build 🍌 (538ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

the \_site/2007 directory should exist 🍌 (000ms)

*And*

the \_site/2007/12 directory should exist 🍌 (000ms)

*And*

the \_site/2007/12/31 directory should exist 🍌 (000ms)

*And*

the "\_site/2007/12/31/entry1.html" file should exist 🍌 (000ms)

*And*

I should see "<p>content for entry1.</p>" in "\_site/index.html" 🍌 (000ms)

*And*

I should see "<html><head></head><body><p>content for entry1.</p>\n\n</body></html>" in "\_site/2007/12/31/entry1.html" 🍌 (000ms)

# Rendering

As a hacker who likes to blog  
I want to be able to make a static site  
In order to share my awesome ideas with the interwebs  
But I want to make it as simply as possible  
So render with Liquid and place in Layouts

## Scenario: Render Liquid and place in layout

*Given*

I have a "index.html" page with layout "simple" that contains "Hi there, Jekyll {{ jekyll.environment }}" 🍷 (000ms)

*And*

I have a simple layout that contains "{{ content }}" Ahoy, indeed!" 🍷 (000ms)

*When*

I run jekyll build 🍷 (569ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

I should see "Hi there, Jekyll development!\nAhoy, indeed" in "\_site/index.html" 🍷 (000ms)

## Scenario: Don't place asset files in layout

*Given*

I have an "index.scss" page with layout "simple" that contains ".foo-bar { color:black; }" 🍌 (000ms)

*And*

I have an "index.coffee" page with layout "simple" that contains "whatever()" 🍌 (000ms)

*And*

I have a configuration file with "gems" set to "[jekyll-coffeescript]" 🍌 (000ms)

*And*

I have a simple layout that contains "{{ content }}Ahoy, indeed!" 🍌 (000ms)

*When*

I run jekyll build 🍌 (882ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should not see "Ahoy, indeed!" in "\_site/index.css" 🍌 (000ms)

*And*

I should not see "Ahoy, indeed!" in "\_site/index.js" 🍌 (000ms)

## Scenario: Render liquid in Sass

*Given*

I have an "index.scss" page that contains ".foo-bar { color:{{site.color}}; }" 🍌 (000ms)

*And*

I have a configuration file with "color" set to "red" 🍌 (000ms)

*When*

I run jekyll build 🍌 (590ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see ".foo-bar {\n color: red; }" in "\_site/index.css" 🍌 (000ms)

## Scenario: Not render liquid in CoffeeScript without explicitly including jekyll-coffeescript

### Given

I have an "index.coffee" page with animal "cicada" that contains "hey='for {{page.animal}}'" 🍌 (000ms)

### When

I run jekyll build 🍌 (586ms)

### Then

the \_site directory should exist 🍌 (000ms)

### And

the "\_site/index.js" file should not exist 🍌 (000ms)

## Scenario: Render liquid in CoffeeScript with jekyll-coffeescript enabled

### Given

I have an "index.coffee" page with animal "cicada" that contains "hey='for {{page.animal}}'" 🍌 (000ms)

### And

I have a configuration file with "gems" set to "[jekyll-coffeescript]" 🍌 (000ms)

### When

I run jekyll build 🍌 (796ms)

### Then

the \_site directory should exist 🍌 (000ms)

### And

I should see "hey = 'for cicada';" in "\_site/index.js" 🍌 (000ms)

## Site configuration

As a hacker who likes to blog  
I want to be able to configure jekyll  
In order to make setting up a site easier

## Scenario: Change source directory

*Given*

I have a blank site in "\_sourcedir" 🍌 (000ms)

*And*

I have an "\_sourcedir/index.html" file that contains "Changing source directory" 🍌 (000ms)

*And*

I have a configuration file with "source" set to "\_sourcedir" 🍌 (000ms)

*When*

I run jekyll build 🍌 (574ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Changing source directory" in "\_site/index.html" 🍌 (000ms)

## Scenario: Change destination directory

*Given*

I have an "index.html" file that contains "Changing destination directory" 🍌 (000ms)

*And*

I have a configuration file with "destination" set to "\_mysite" 🍌 (000ms)

*When*

I run jekyll build 🍌 (578ms)

*Then*

the \_mysite directory should exist 🍌 (000ms)

*And*

I should see "Changing destination directory" in "\_mysite/index.html" 🍌 (000ms)

## Scenario Outline: Similarly named source and destination 🗨️

Table 2. Example

source	dest	file_exist
mysite_source	mysite	
mysite	mysite_dest	
mysite/	mysite	not

source	dest	file_exist
mysite	./mysite	not
mysite/source	mysite	not
mysite	mysite/dest	

## Scenario: Exclude files inline

*Given*

I have an "Rakefile" file that contains "I want to be excluded" 🍌 (000ms)

*And*

I have an "README" file that contains "I want to be excluded" 🍌 (000ms)

*And*

I have an "index.html" file that contains "I want to be included" 🍌 (000ms)

*And*

I have a configuration file with "exclude" set to "['Rakefile', 'README']" 🍌 (000ms)

*When*

I run jekyll build 🍌 (558ms)

*Then*

I should see "I want to be included" in "\_site/index.html" 🍌 (000ms)

*And*

the "\_site/Rakefile" file should not exist 🍌 (000ms)

*And*

the "\_site/README" file should not exist 🍌 (000ms)

## Scenario: Exclude files with YAML array

*Given*

I have an "Rakefile" file that contains "I want to be excluded" 🍷 (000ms)

*And*

I have an "README" file that contains "I want to be excluded" 🍷 (000ms)

*And*

I have an "index.html" file that contains "I want to be included" 🍷 (000ms)

*And*

I have a configuration file with "exclude" set to: 🍷 (000ms)

value
README
Rakefile

*When*

I run jekyll build 🍷 (498ms)

*Then*

I should see "I want to be included" in "\_site/index.html" 🍷 (000ms)

*And*

the "\_site/Rakefile" file should not exist 🍷 (000ms)

*And*

the "\_site/README" file should not exist 🍷 (000ms)

## Scenario: Use RDiscount for markup

*Given*

I have an "index.markdown" page that contains "[Google](<a href='\"http://google.com)\"'\" class='\"bare\">http://google.com)\"</a> <font name='\"fa\"> </font> <span class='\"small right\">(000ms)</span>

*And*

I have a configuration file with "markdown" set to "rdiscount" 🍌 (000ms)

*When*

I run jekyll build 🍌 (498ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "<a href='\"http://google.com\">Google</a>" in "\_site/index.html" 🍌 (000ms)

## Scenario: Use Kramdown for markup

*Given*

I have an "index.markdown" page that contains "[Google](<a href='\"http://google.com)\"'\" class='\"bare\">http://google.com)\"</a> <font name='\"fa\"> </font> <span class='\"small right\">(000ms)</span>

*And*

I have a configuration file with "markdown" set to "kramdown" 🍌 (000ms)

*When*

I run jekyll build 🍌 (552ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "<a href='\"http://google.com\">Google</a>" in "\_site/index.html" 🍌 (000ms)

## Scenario: Use Redcarpet for markup



*Given*

I have an "index.markdown" page that contains "[Google](<a href='\"http://google.com')\"'\" class='\"bare'>http://google.com)\"</a> <font name='\"fa'> </font> <span class='\"small right'>(000ms)</span>

*And*

I have a configuration file with "markdown" set to "redcarpet" 🍌 (000ms)

*When*

I run jekyll build 🍌 (595ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "<a href='\"http://google.com'>Google</a>" in "\_site/index.html" 🍌 (000ms)

## Scenario: Highlight code with pygments

*Given*

I have an "index.html" page that contains "{% highlight ruby %} puts 'Hello world!' {% endhighlight %}" 🍌 (000ms)

*When*

I run jekyll build 🍌 (593ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Hello world!" in "\_site/index.html" 🍌 (000ms)

*And*

I should see "class='\"highlight'\"" in "\_site/index.html" 🍌 (000ms)

## Scenario: Highlight code with rouge

*Given*

I have an "index.html" page that contains "{% highlight ruby %} puts 'Hello world!' {% endhighlight %}" 🍷 (000ms)

*And*

I have a configuration file with "highlighter" set to "rouge" 🍷 (000ms)

*When*

I run jekyll build 🍷 (630ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

I should see "Hello world!" in "\_site/index.html" 🍷 (000ms)

*And*

I should see "class=\"highlight\"" in "\_site/index.html" 🍷 (000ms)

## Scenario: Rouge renders code block once

*Given*

I have a configuration file with "highlighter" set to "rouge" 🍷 (000ms)

*And*

I have a \_posts directory 🍷 (000ms)

*And*

I have the following post: 🍷 (001ms)

title	date	layout	content
foo	2014-04-27 11:34	default	{% highlight text %} test {% endhighlight %}

*When*

I run jekyll build 🍷 (733ms)

*Then*

I should not see "highlight(.\*?)highlight" in "\_site/2014/04/27/foo.html" 🍷 (000ms)

## Scenario: Set time and no future dated posts

*Given*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have a page layout that contains "Page Layout: {{ site.posts.size }} on {{ site.time | date: \"%Y-%m-%d\" }}" 🍌 (000ms)

*And*

I have a post layout that contains "Post Layout: {{ content }}" 🍌 (000ms)

*And*

I have an "index.html" page with layout "page" that contains "site index page" 🍌 (000ms)

*And*

I have a configuration file with: 🍌 (000ms)

key	value
time	2010-01-01
future	false

*And*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
entry1	2007-12-31	post	content for entry1.
entry2	2020-01-31	post	content for entry2.

*When*

I run `jeekyll build` 🍌 (575ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Page Layout: 1 on 2010-01-01" in "`_site/index.html`" 🍌 (000ms)

*And*

I should see "Post Layout: <p>content for entry1.</p>" in "`_site/2007/12/31/entry1.html`" 🍌 (000ms)

*And*

the "`_site/2020/01/31/entry2.html`" file should not exist 🍌 (000ms)

## Scenario: Set time and future dated posts allowed

*Given*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have a page layout that contains "Page Layout: {{ site.posts.size }} on {{ site.time | date: \"%Y-%m-%d\" }}" 🍌 (000ms)

*And*

I have a post layout that contains "Post Layout: {{ content }}" 🍌 (000ms)

*And*

I have an "index.html" page with layout "page" that contains "site index page" 🍌 (000ms)

*And*

I have a configuration file with: 🍌 (000ms)

key	value
time	2010-01-01
future	true

*And*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
entry1	2007-12-31	post	content for entry1.
entry2	2020-01-31	post	content for entry2.

*When*

I run `jeekyll build` 🍌 (540ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Page Layout: 2 on 2010-01-01" in "`_site/index.html`" 🍌 (000ms)

*And*

I should see "Post Layout: `<p>content for entry1.</p>`" in "`_site/2007/12/31/entry1.html`" 🍌 (000ms)

*And*

I should see "Post Layout: <p>content for entry2.</p>" in "\_site/2020/01/31/entry2.html" 🍻  
(000ms)

**Scenario: Generate proper dates with explicitly set timezone (same as posts' time)**

*Given*

I have a `_layouts` directory 🍷 (000ms)

*And*

I have a page layout that contains "Page Layout: `{{ site.posts.size }}`" 🍷 (000ms)

*And*

I have a post layout that contains "Post Layout: `{{ content }}` built at `{{ page.date | date_to_xmlschema }}`" 🍷 (000ms)

*And*

I have an "index.html" page with layout "page" that contains "site index page" 🍷 (000ms)

*And*

I have a configuration file with: 🍷 (000ms)

key	value
timezone	America/New_York

*And*

I have a `_posts` directory 🍷 (000ms)

*And*

I have the following posts: 🍷 (001ms)

title	date	layout	content
entry1	2013-04-09 23:22 -0400	post	content for entry1.
entry2	2013-04-10 03:14 -0400	post	content for entry2.

*When*

I run `jeekyll build` 🍷 (544ms)

*Then*

the `_site` directory should exist 🍷 (000ms)

*And*

I should see "Page Layout: 2" in "`_site/index.html`" 🍷 (000ms)

*And*

I should see "Post Layout: `<p>content for entry1.</p>\n` built at 2013-04-09T23:22:00-04:00" in "`_site/2013/04/09/entry1.html`" 🍷 (000ms)

*And*

I should see "Post Layout: `<p>content for entry2.</p>\n` built at 2013-04-10T03:14:00-04:00" in "`_site/2013/04/10/entry2.html`" 🍷 (000ms)

## Scenario: Generate proper dates with explicitly set timezone (different than posts' time)

*Given*

I have a \_layouts directory 🍌 (000ms)

*And*

I have a page layout that contains "Page Layout: {{ site.posts.size }}" 🍌 (000ms)

*And*

I have a post layout that contains "Post Layout: {{ content }} built at {{ page.date | date\_to\_xmlschema }}" 🍌 (000ms)

*And*

I have an "index.html" page with layout "page" that contains "site index page" 🍌 (000ms)

*And*

I have a configuration file with: 🍌 (000ms)

key	value
timezone	Pacific/Honolulu

*And*

I have a \_posts directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (001ms)

title	date	layout	content
entry1	2013-04-09 23:22 +0400	post	content for entry1.
entry2	2013-04-10 03:14 +0400	post	content for entry2.

*When*

I run jekyll build 🍌 (639ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Page Layout: 2" in "\_site/index.html" 🍌 (000ms)

*And*

the "\_site/2013/04/09/entry1.html" file should exist 🍌 (000ms)

*And*

the "\_site/2013/04/09/entry2.html" file should exist 🍌 (000ms)

*And*

I should see "Post Layout: <p>content for entry1.</p>\n built at 2013-04-09T09:22:00-10:00" in "\_site/2013/04/09/entry1.html" 🍌 (000ms)

*And*

I should see "Post Layout: <p>content for entry2.</p>\n built at 2013-04-09T13:14:00-10:00" in "\_site/2013/04/09/entry2.html" 🍌 (000ms)

## **Scenario: Limit the number of posts generated by most recent date**



*Given*

I have a \_posts directory 🍌 (000ms)

*And*

I have a configuration file with: 🍌 (000ms)

key	value
limit_posts	2

*And*

I have the following posts: 🍌 (001ms)

title	date	content
Apples	2009-03-27	An article about apples
Oranges	2009-04-01	An article about oranges
Bananas	2009-04-05	An article about bananas

*When*

I run jekyll build 🍌 (581ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

the "\_site/2009/04/05/bananas.html" file should exist 🍌 (000ms)

*And*

the "\_site/2009/04/01/oranges.html" file should exist 🍌 (000ms)

*And*

the "\_site/2009/03/27/apples.html" file should not exist 🍌 (000ms)

**Scenario: Copy over normally excluded files when they are explicitly included**

*Given*

I have a ".gitignore" file that contains ".DS\_Store" 🍌 (000ms)

*And*

I have an ".htaccess" file that contains "SomeDirective" 🍌 (000ms)

*And*

I have a configuration file with "include" set to: 🍌 (000ms)

value
.gitignore
.foo

*When*

I run jekyll build 🍌 (517ms)

*Then*

the "\_site" directory should exist 🍌 (000ms)

*And*

I should see ".DS\_Store" in "\_site/.gitignore" 🍌 (000ms)

*And*

the "\_site/.htaccess" file should not exist 🍌 (000ms)

## Scenario: Using a different layouts directory

*Given*

I have a "\_theme" directory 🍌 (000ms)

*And*

I have a page theme that contains "Page Layout: {{ site.posts.size }} on {{ site.time | date: \"%Y-%m-%d\" }}" 🍌 (000ms)

*And*

I have a post theme that contains "Post Layout: {{ content }}" 🍌 (000ms)

*And*

I have an "index.html" page with layout "page" that contains "site index page" 🍌 (000ms)

*And*

I have a configuration file with: 🍌 (000ms)

key	value
time	2010-01-01
future	true
layouts	_theme

*And*

I have a \_posts directory 🍌 (000ms)

*And*

I have the following posts: 🍌 (000ms)

title	date	layout	content
entry1	2007-12-31	post	content for entry1.
entry2	2020-01-31	post	content for entry2.

*When*

I run jekyll build 🍌 (577ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Page Layout: 2 on 2010-01-01" in "\_site/index.html" 🍌 (000ms)

*And*

I should see "Post Layout: <p>content for entry1.</p>" in "\_site/2007/12/31/entry1.html" 🍌 (000ms)

*And*

I should see "Post Layout: <p>content for entry2.</p>" in "\_site/2020/01/31/entry2.html" 🍌 (000ms)

## Scenario: arbitrary file reads via layouts

*Given*

I have an "index.html" page with layout "page" that contains "FOO" 🍌 (000ms)

*And*

I have a "\_config.yml" file that contains "layouts: '../..../usr/include'" 🍌 (000ms)

*When*

I run jekyll build 🍌 (493ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "FOO" in "\_site/index.html" 🍌 (000ms)

*And*

I should not see " " in "\_site/index.html" 🍌 (000ms)

## Site data

As a hacker who likes to blog

I want to be able to embed data into my site

In order to make the site slightly dynamic

### Scenario: Use page variable in a page

*Given*

I have an "contact.html" page with title "Contact" that contains "{{ page.title }}: [email@example.com](mailto:email@example.com)" 🍌 (000ms)

*When*

I run jekyll build 🍌 (540ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "Contact: [email@example.com](mailto:email@example.com)" in "\_site/contact.html" 🍌 (000ms)

## Scenario Outline: Use page.path variable in a page 🗨️

Table 3. Example

dir	path
.	index.html
dir	dir/about.html
dir/nested	dir/nested/page.html

## Scenario: Override page.path

### Given

I have an "override.html" page with path "custom-override.html" that contains "Custom path: {{ page.path }}" 🗨️ (000ms)

### When

I run jekyll build 🗨️ (493ms)

### Then

the \_site directory should exist 🗨️ (000ms)

### And

I should see "Custom path: custom-override.html" in "\_site/override.html" 🗨️ (000ms)

## Scenario: Use site.time variable

### Given

I have an "index.html" page that contains "{{ site.time }}" 🗨️ (000ms)

### When

I run jekyll build 🗨️ (529ms)

### Then

the \_site directory should exist 🗨️ (000ms)

### And

I should see today's time in "\_site/index.html" 🗨️ (000ms)

## Scenario: Use site.posts variable for latest post

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have an "index.html" page that contains "{{ site.posts.first.title }}: {{ site.posts.first.url }}" 🍌 (000ms)

*And*

I have the following posts: 🍌 (001ms)

title	date	content
First Post	2009-03-25	My First Post
Second Post	2009-03-26	My Second Post
Third Post	2009-03-27	My Third Post

*When*

I run `jeekyll build` 🍌 (545ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Third Post: /2009/03/27/third-post.html" in "`_site/index.html`" 🍌 (000ms)

## Scenario: Use `site.posts` variable in a loop

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have an "index.html" page that contains "{% for post in site.posts %} {{ post.title }} {% endfor %}" 🍌 (000ms)

*And*

I have the following posts: 🍌 (001ms)

title	date	content
First Post	2009-03-25	My First Post
Second Post	2009-03-26	My Second Post
Third Post	2009-03-27	My Third Post

*When*

I run `jeekyll build` 🍌 (579ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Third Post Second Post First Post" in "`_site/index.html`" 🍌 (000ms)

## Scenario: Use `site.categories.code` variable

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have an "index.html" page that contains "{% for post in site.categories.code %} {{ post.title }} {% endfor %}" 🍌 (000ms)

*And*

I have the following posts: 🍌 (001ms)

title	date	category	content
Awesome Hack	2009-03-26	code	puts 'Hello World'
Delicious Beer	2009-03-26	food	1) Yuengling

*When*

I run `jeekyll build` 🍌 (581ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "Awesome Hack" in "`_site/index.html`" 🍌 (000ms)

## Scenario: Use `site.tags` variable



*Given*

I have a \_posts directory 🍷 (000ms)

*And*

I have an "index.html" page that contains "{% for post in site.tags.beer %} {{ post.content }}  
{% endfor %}" 🍷 (000ms)

*And*

I have the following posts: 🍷 (000ms)

title	date	tag	content
Delicious Beer	2009-03-26	beer	1) Yuengling

*When*

I run jekyll build 🍷 (586ms)

*Then*

the \_site directory should exist 🍷 (000ms)

*And*

I should see "Yuengling" in "\_site/index.html" 🍷 (000ms)

## Scenario: Order Posts by name when on the same date

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have an "index.html" page that contains "{% for post in site.posts %}{{ post.title }}:{{ post.previous.title}},{{ post.next.title}} {% endfor %}" 🍌 (000ms)

*And*

I have the following posts: 🍌 (001ms)

title	date	content
first	2009-02-26	first
A	2009-03-26	A
B	2009-03-26	B
C	2009-03-26	C
last	2009-04-26	last

*When*

I run `jeekyll build` 🍌 (594ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "last:C, C:B,last B:A,C A:first,B first:,A" in "`_site/index.html`" 🍌 (000ms)

## Scenario: Use configuration date in site payload

*Given*

I have an "index.html" page that contains "{{ site.url }}" 🍌 (000ms)

*And*

I have a configuration file with "url" set to "http://example.com" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (616ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "http://example.com" in "`_site/index.html`" 🍌 (000ms)

## Scenario: Access Jekyll version via jekyll.version

### Given

I have an "index.html" page that contains "{{ jekyll.version }}" 🍌 (000ms)

### When

I run jekyll build 🍌 (607ms)

### Then

the \_site directory should exist 🍌 (000ms)

### And

I should see "\d+\\.\\d+\\.\\d+" in "\_site/index.html" 🍌 (000ms)

## Site pagination

In order to paginate my blog

As a blog's user

I want divide the posts in several pages

## Scenario Outline: Paginate with N posts per page 🍌

Table 4. Example

num	exist	posts	not_exist
1	4	1	5
2	2	2	3
3	2	1	3

## Scenario Outline: Setting a custom pagination path 🍌

Table 5. Example

exist	posts	not_exist
2	1	5
3	1	6
4	1	7

## Scenario Outline: Setting a custom pagination path without an index.html in it 🍌

Table 6. Example

exist	posts	not_exist
2	1	5
3	1	6
4	1	7

## frontmatter defaults

### Scenario: Use default for frontmatter variables internally

*Given*

I have a \_layouts directory 🍌 (000ms)

*And*

I have a pretty layout that contains "THIS IS THE LAYOUT: {{content}}" 🍌 (000ms)

*And*

I have a \_posts directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	content
default layout	2013-09-11	just some post

*And*

I have an "index.html" page with title "some title" that contains "just some page" 🍌 (000ms)

*And*

I have a configuration file with "defaults" set to "[{scope: {path: ""}, values: {layout: "pretty"}}]" 🍌 (000ms)

*When*

I run jekyll build 🍌 (638ms)

*Then*

the \_site directory should exist 🍌 (000ms)

*And*

I should see "THIS IS THE LAYOUT: <p>just some post</p>" in "\_site/2013/09/11/default-layout.html" 🍌 (000ms)

*And*

I should see "THIS IS THE LAYOUT: just some page" in "\_site/index.html" 🍌 (000ms)

## Scenario: Use default for frontmatter variables in Liquid

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	content
default data	2013-09-11	<p>{{page.custom}}</p><div>{{page.author}}</div>

*And*

I have an "index.html" page that contains "just {{page.custom}} by {{page.author}}" 🍌 (000ms)

*And*

I have a configuration file with "defaults" set to "[{scope: {path: ""}, values: {custom: "some special data", author: "Ben"}}]" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (626ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "<p>some special data</p>\n<div>Ben</div>" in "`_site/2013/09/11/default-data.html`" 🍌 (000ms)

*And*

I should see "just some special data by Ben" in "`_site/index.html`" 🍌 (000ms)

## Scenario: Override frontmatter defaults by path

*Given*

I have a `_layouts` directory 🍌 (000ms)

*And*

I have a root layout that contains "root: {{ content }}" 🍌 (000ms)

*And*

I have a subfolder layout that contains "subfolder: {{ content }}" 🍌 (000ms)

*And*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	content
about	2013-10-14	info on {{page.description}}

*And*

I have a special/`_posts` directory 🍌 (000ms)

*And*

I have the following post in "special": 🍌 (000ms)

title	date	path	content
about	2013-10-14	local	info on {{page.description}}

*And*

I have an "index.html" page with title "overview" that contains "Overview for {{page.description}}" 🍌 (000ms)

*And*

I have an "special/index.html" page with title "section overview" that contains "Overview for {{page.description}}" 🍌 (000ms)

*And*

I have a configuration file with "defaults" set to "[{scope: {path: "special"}, values: {layout: "subfolder", description: "the special section"}}, {scope: {path: ""}, values: {layout: "root", description: "the webpage"}}]" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (641ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "root: <p>info on the webpage</p>" in "`_site/2013/10/14/about.html`" 🍌 (000ms)

*And*

I should see "subfolder: <p>info on the special section</p>" in "`_site/special/2013/10/14/about.html`" 🍌 (000ms)

*And*

I should see "root: Overview for the webpage" in "`_site/index.html`" 🍌 (000ms)

And

I should see "subfolder: Overview for the special section" in "\_site/special/index.html" 🍌 (000ms)

## Scenario: Use frontmatter variables by relative path

Given

I have a \_layouts directory 🍌 (000ms)

And

I have a main layout that contains "main: {{ content }}" 🍌 (000ms)

And

I have a \_posts directory 🍌 (000ms)

And

I have the following post: 🍌 (000ms)

title	date	content
about	2013-10-14	content of site/2013/10/14/about.html

And

I have a special/\_posts directory 🍌 (000ms)

And

I have the following post in "special": 🍌 (000ms)

title	date	path	content
about1	2013-10-14	local	content of site/special/2013/10/14/ /about1.html
about2	2013-10-14	local	content of site/special/2013/10/14/ /about2.html

And

I have a configuration file with "defaults" set to "[{scope: {path: "special"}, values: {layout: "main"}}, {scope: {path: "special/\_posts"}, values: {layout: "main"}}, {scope: {path: "\_posts"}, values: {layout: "main"}}]" 🍌 (000ms)

When

I run jekyll build 🍌 (643ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see "main: `<p>content of site/2013/10/14/about.html</p>`" in "`_site/2013/10/14/about.html`" 🍌 (000ms)

*And*

I should see "main: `<p>content of site/special/2013/10/14/about1.html</p>`" in "`_site/special/2013/10/14/about1.html`" 🍌 (000ms)

*And*

I should see "main: `<p>content of site/special/2013/10/14/about2.html</p>`" in "`_site/special/2013/10/14/about2.html`" 🍌 (000ms)

## Scenario: Override frontmatter defaults by type

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	content
this is a post	2013-10-14	blabla

*And*

I have an "index.html" page that contains "interesting stuff" 🍌 (000ms)

*And*

I have a configuration file with "defaults" set to "[{scope: {path: '', type: 'post'}, values: {permalink: '/post.html'}}, {scope: {path: '', type: 'page'}, values: {permalink: '/page.html'}}, {scope: {path: ''}, values: {permalink: '/perma.html'}}]" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (543ms)

*Then*

I should see "blabla" in "`_site/post.html`" 🍌 (000ms)

*And*

I should see "interesting stuff" in "`_site/page.html`" 🍌 (000ms)

*But*

the "`_site/perma.html`" file should not exist 🍌 (000ms)



## Scenario: Actual frontmatter overrides defaults

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	permalink	author	content
override	2013-10-14	/frontmatter.html	some guy	a blog by {{page.author}}

*And*

I have an "index.html" page with permalink "override.html" that contains "nothing" 🍌 (000ms)

*And*

I have a configuration file with "defaults" set to "[{scope: {path: ""}, values: {permalink: "/perma.html", author: "Chris"}}]" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (556ms)

*Then*

I should see "a blog by some guy" in "`_site/frontmatter.html`" 🍌 (000ms)

*And*

I should see "nothing" in "`_site/override.html`" 🍌 (000ms)

*But*

the "`_site/perma.html`" file should not exist 🍌 (000ms)

## Scenario: Define permalink default for posts

*Given*

I have a `_posts` directory 🍌 (000ms)

*And*

I have the following post: 🍌 (000ms)

title	date	category	content
testpost	2013-10-14	blog	blabla

*And*

I have a configuration file with "defaults" set to "[{scope: {path: "", type: "posts"}, values: {permalink: "/:categories/:title/"}}]" 🍌 (000ms)

*When*

I run `jeekyll build` 🍌 (584ms)

*Then*

I should see "blabla" in "`_site/blog/testpost/index.html`" 🍌 (000ms)

## Scenario: Use frontmatter defaults in collections

*Given*

I have a `_slides` directory 🍌 (000ms)

*And*

I have a `"index.html"` file that contains `"nothing"` 🍌 (000ms)

*And*

I have a `"_slides/slide1.html"` file with content: 🍌 (000ms)

```
---  
---  
Value: {{ page.myval }}
```

*And*

I have a `"_config.yml"` file with content: 🍌 (000ms)

```
collections:  
  slides:  
    output: true  
defaults:  
  -  
    scope:  
      path: ""  
      type: slides  
    values:  
      myval: "Test"
```

*When*

I run `jeekyll build` 🍌 (607ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see `"Value: Test"` in `"_site/slides/slide1.html"` 🍌 (000ms)

## Scenario: Override frontmatter defaults inside a collection

*Given*

I have a `_slides` directory 🍌 (000ms)

*And*

I have a `"index.html"` file that contains `"nothing"` 🍌 (000ms)

*And*

I have a `"_slides/slide2.html"` file with content: 🍌 (000ms)

```
---  
myval: Override  
---  
Value: {{ page.myval }}
```

*And*

I have a `"_config.yml"` file with content: 🍌 (000ms)

```
collections:  
  slides:  
    output: true  
defaults:  
  -  
    scope:  
      path: ""  
      type: slides  
    values:  
      myval: "Test"
```

*When*

I run `jeekyll build` 🍌 (509ms)

*Then*

the `_site` directory should exist 🍌 (000ms)

*And*

I should see `"Value: Override"` in `"_site/slides/slide2.html"` 🍌 (000ms)

## Scenario: Deep merge frontmatter defaults

*Given*

I have an "index.html" page with fruit "{orange: 1}" that contains "Fruits: {{ page.fruit.orange | plus: page.fruit.apple }}" 🍌 (000ms)

*And*

I have a configuration file with "defaults" set to "[{scope: {path: ""}, values: {fruit: {apple: 2}}}]"

*When*

I run jekyll build 🍌 (560ms)

*Then*

I should see "Fruits: 3" in "\_site/index.html" 🍌 (000ms)