OFFENSIVE SECURITY

Penetration Test Report for Internal Lab

marry.poppins@pltravers.writer

OSID: 271828

OFFENSIVE® security

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

1.1. MkDocs

Project documentation with Markdown.

MkDocs is a **fast**, **simple** and **downright gorgeous** static site generator that's geared towards building project documentation. Documentation source files are written in Markdown, and configured with a single YAML configuration file. Start by reading the introductory tutorial, then check the User Guide for more information.
Getting Started User Guide

1.1.1. Features

1.1.1.1. Great themes available

There's a stack of good looking themes available for MkDocs. Choose between the built in themes: mkdocs and readthedocs, select one of the third-party themes listed on the MkDocs Themes wiki page, or build your own.

1.1.1.2. Easy to customize

Get your project documentation looking just the way you want it by customizing your theme and/or installing some plugins. Modify Markdown's behavior with Markdown extensions. Many configuration options are available.

1.1.1.3. Preview your site as you work

The built-in dev-server allows you to preview your documentation as you're writing it. It will even auto-reload and refresh your browser whenever you save your changes.

1.1.1.4. Host anywhere

MkDocs builds completely static HTML sites that you can host on GitHub pages, Amazon S3, or anywhere else you choose.

1.2. Getting Started with MkDocs

An introductory tutorial!

1.2.1. Installation

To install MkDocs, run the following command from the command line:

```
pip install mkdocs
```
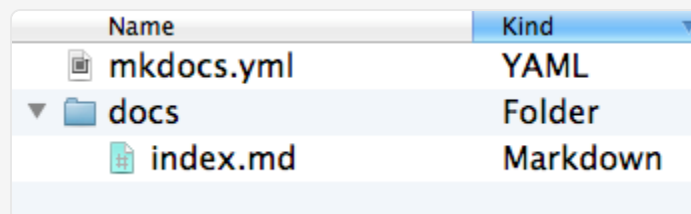
For more details, see the Installation Guide.

1.2.2. Creating a new project

Getting started is super easy. To create a new project, run the following command from the command line:

```
mkdocs new my-project
cd my-project
```

Take a moment to review the initial project that has been created for you.

Figure 1 - The initial MkDocs layout

There's a single configuration file named `mkdocs.yml` , and a folder named `docs` that will contain your documentation source files (`docs` is the default value for the docs_dir configuration setting). Right now the `docs` folder just contains a single documentation page, named `index.md` .

MkDocs comes with a built-in dev-server that lets you preview your documentation as you work on it. Make sure you're in the same directory as the `mkdocs.yml` configuration file, and then start the server by running the `mkdocs serve` command:

```
 1  $ mkdocs serve
 2  INFO    -  Building documentation...
 3  INFO    -  Cleaning site directory
 4  [I 160402 15:50:43 server:271] Serving on http://127.0.0.1:8000
 5  [I 160402 15:50:43 handlers:58] Start watching changes
 6  [I 160402 15:50:43 handlers:60] Start detecting changes
 7  [I 160402 15:50:43 handlers:58] Start watching changes
 8  [I 160402 15:50:43 handlers:60] Start detecting changes
 9  [I 160402 15:50:43 handlers:58] Start watching changes
10  [I 160402 15:50:43 handlers:60] Start detecting changes
11  [I 160402 15:50:43 handlers:58] Start watching changes
12  [I 160402 15:50:43 handlers:60] Start detecting changes
13  [I 160402 15:50:43 handlers:58] Start watching changes
14  [I 160402 15:50:43 handlers:60] Start detecting changes
```

Table 1 - Code with lines numbers and lines highlighted

Open up `http://127.0.0.1:8000/` in your browser, and you'll see the default home page being displayed:
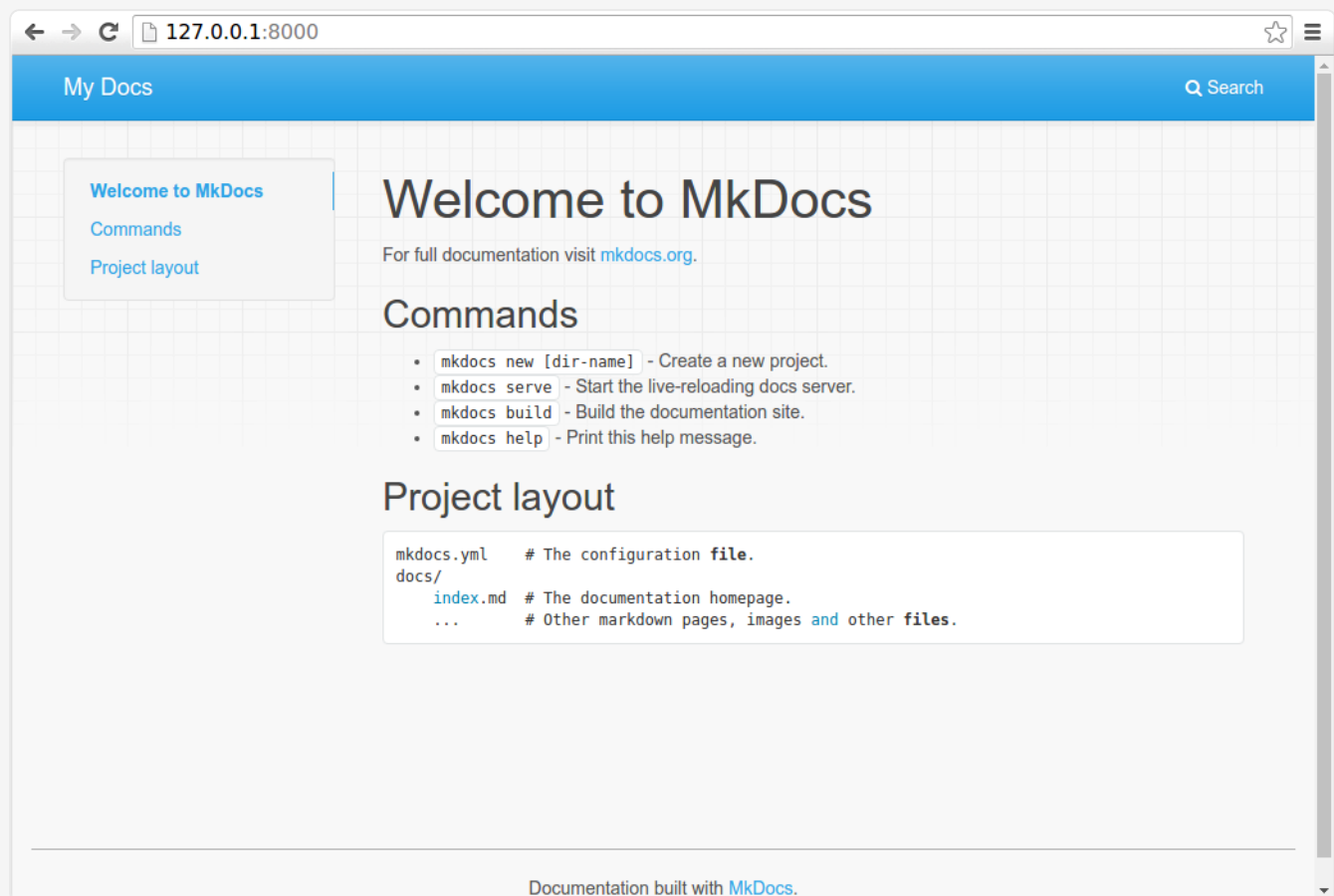
Figure 2 - The MkDocs live server

The dev-server also supports auto-reloading, and will rebuild your documentation whenever anything in the configuration file, documentation directory, or theme directory changes.

```
 1  $ mkdocs serve
 2  INFO     -  Building documentation...
 3  INFO     -  Cleaning site directory
 4  [I 160402 15:50:43 server:271] Serving on http://127.0.0.1:8000
 5  [I 160402 15:50:43 handlers:58] Start watching changes
 6  [I 160402 15:50:43 handlers:60] Start detecting changes
 7  [I 160402 15:50:43 handlers:58] Start watching changes
 8  [I 160402 15:50:43 handlers:60] Start detecting changes
 9  [I 160402 15:50:43 handlers:58] Start watching changes
10  [I 160402 15:50:43 handlers:60] Start detecting changes
11  [I 160402 15:50:43 handlers:58] Start watching changes
12  [I 160402 15:50:43 handlers:60] Start detecting changes
13  [I 160402 15:50:43 handlers:58] Start watching changes
14  [I 160402 15:50:43 handlers:60] Start detecting changes
```

Open the `docs/index.md` document in your text editor of choice, change the initial heading to `MkLorum`, and save your changes. Your browser will auto-reload and you should see your updated documentation immediately.

Now try editing the configuration file: `mkdocs.yml`. Change the `site_name` setting to `MkLorum` and save the file.

```
site_name: MkLorum
site_url: https://example.com/
```

Your browser should immediately reload, and you'll see your new site name take effect.
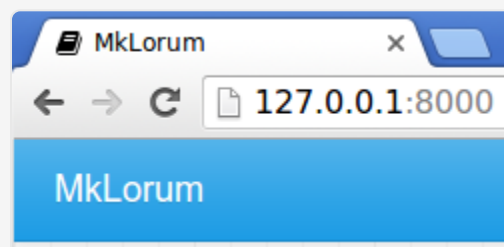
Figure 3 - The `site_name` setting

> **Note**
>
> The `site_name` and `site_url` configuration options are the only two required options in your configuration file. When you create a new project, the `site_url` option is assigned the placeholder value: `https://example.com`. If the final location is known, you can change the setting now to point to it. Or you may choose to leave it alone for now. Just be sure to edit it before you deploy your site to a production server.

1.2.3. Adding pages

Now add a second page to your documentation:

```
curl 'https://jaspervdj.be/lorem-markdownum/markdown.txt' > docs/about.md
```

As our documentation site will include some navigation headers, you may want to edit the configuration file and add some information about the order, title, and nesting of each page in the navigation header by adding a `nav` setting:

```
site_name: MkLorum
site_url: https://example.com/
nav:
    - Home: index.md
    - About: about.md
```

Save your changes and you'll now see a navigation bar with `Home` and `About` items on the left as well as `Search`, `Previous`, and `Next` items on the right.
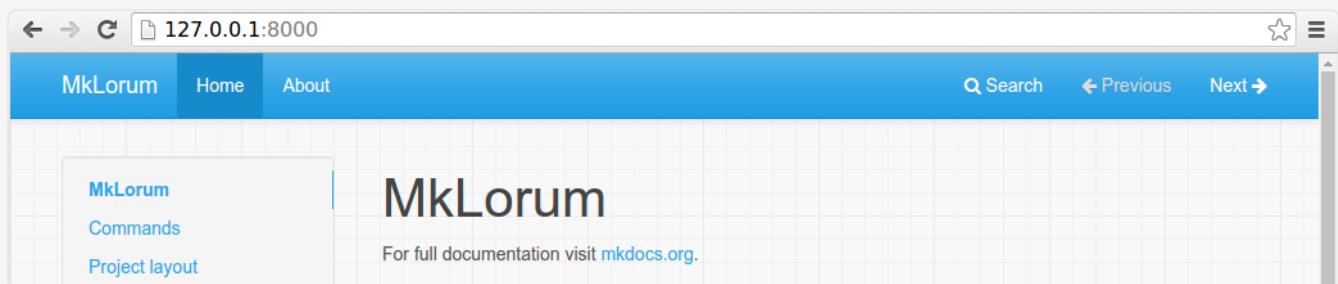
Figure 4 - Screenshot

Try the menu items and navigate back and forth between pages. Then click on `Search`. A search dialog will appear, allowing you to search for any text on any page. Notice that the search results include every occurrence of the search term on the site and links directly to the section of the page in which the search term appears. You get all of that with no effort or configuration on your part!
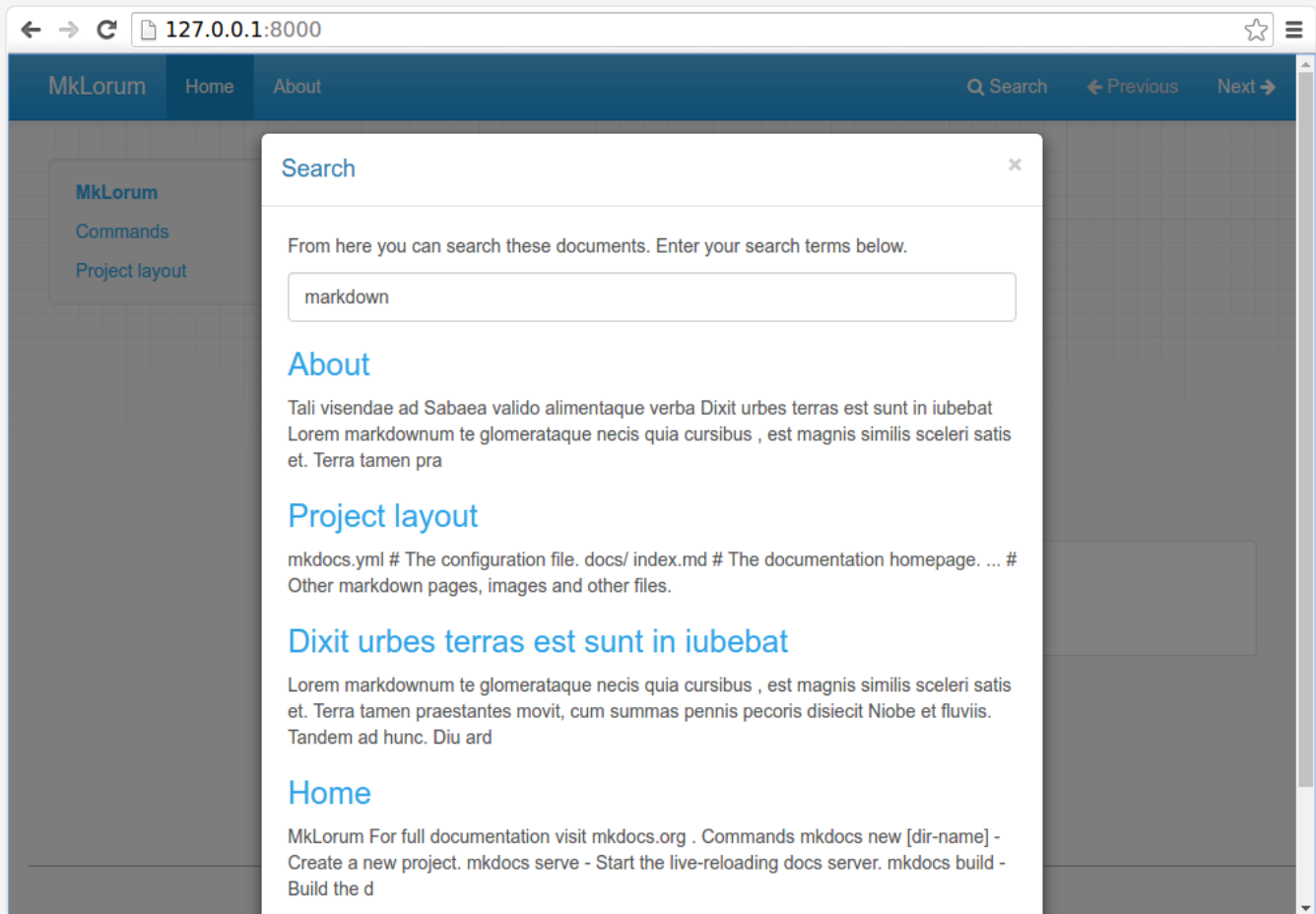
Figure 5 - Screenshot

1.2.4. Theming our documentation

Now change the configuration file to alter how the documentation is displayed by changing the theme. Edit the `mkdocs.yml` file and add a `theme` setting:

```
site_name: MkLorum
site_url: https://example.com/
nav:
    - Home: index.md
    - About: about.md
theme: readthedocs
```

Save your changes, and you'll see the ReadTheDocs theme being used.
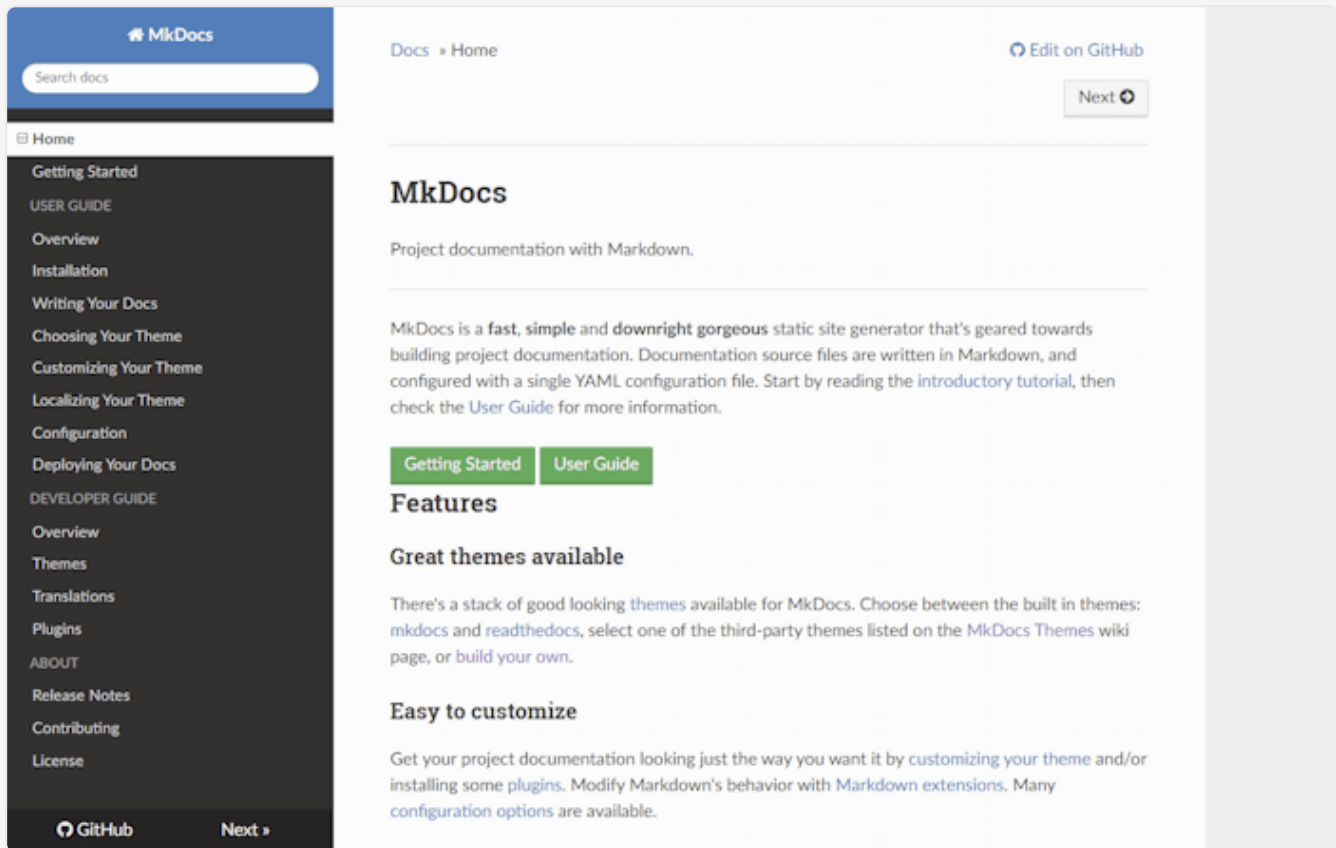
Figure 6 - Screenshot

1.2.5. Changing the Favicon Icon

By default, MkDocs uses the MkDocs favicon icon. To use a different icon, create an `img` subdirectory in the `docs` directory and copy your custom `favicon.ico` file to that directory. MkDocs will automatically detect and use that file as your favicon icon.

1.2.6. Building the site

That's looking good. You're ready to deploy the first pass of your `MkLorum` documentation. First build the documentation:

```
mkdocs build
```

This will create a new directory, named `site`. Take a look inside the directory:

```
$ ls site
about   fonts   index.html   license   search.html
css     img     js           mkdocs    sitemap.xml
```

Notice that your source documentation has been output as two HTML files named `index.html` and `about/index.html` . You also have various other media that's been copied into the `site` directory as part of the documentation theme. You even have a `sitemap.xml` file and `mkdocs/search_index.json` .

If you're using source code control such as `git` you probably don't want to check your documentation builds into the repository. Add a line containing `site/` to your `.gitignore` file.

```
echo "site/" >> .gitignore
```

If you're using another source code control tool you'll want to check its documentation on how to ignore specific directories.

1.2.7. Other Commands and Options

There are various other commands and options available. For a complete list of commands, use the `--help` flag:

```
mkdocs --help
```

To view a list of options available on a given command, use the `--help` flag with that command. For example, to get a list of all options available for the `build` command run the following:

```
mkdocs build --help
```

1.2.8. Deploying

The documentation site that you just built only uses static files so you'll be able to host it from pretty much anywhere. Simply upload the contents of the entire `site` directory to wherever you're hosting your website from and you're done. For specific instructions on a number of common hosts, see the Deploying your Docs page.

1.2.9. Getting help

See the User Guide for more complete documentation of all of MkDocs' features.

To get help with MkDocs, please use the GitHub discussions or GitHub issues.

2. User Guide

2.1. User Guide

Building Documentation with MkDocs

The MkDocs Developer Guide provides documentation for users of MkDocs. See Getting Started for an introductory tutorial. You can jump directly to a page listed below, or use the next and previous buttons in the navigation bar at the top of the page to move through the documentation in order.

- Installation
- Writing Your Docs
- Choosing Your Theme
- Customizing Your Theme
- Configuration
- Deploying Your Docs

2.2. MkDocs Installation

A detailed guide.

2.2.1. Requirements

MkDocs requires a recent version of Python and the Python package manager, pip, to be installed on your system.

You can check if you already have these installed from the command line:

```
$ python --version
Python 3.8.2
$ pip --version
pip 20.0.2 from /usr/local/lib/python3.8/site-packages/pip (python 3.8)
```

If you already have those packages installed, you may skip down to Installing MkDocs.

2.2.1.1. Installing Python

Install Python using your package manager of choice, or by downloading an installer appropriate for your system from python.org and running it.

> **Note**
>
> If you are installing Python on Windows, be sure to check the box to have Python added to your PATH if the installer offers such an option (it's normally off by default).
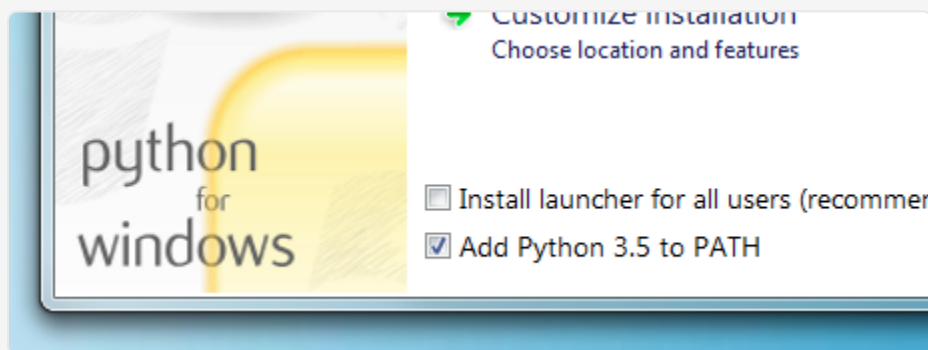
Figure 7 - Add Python to PATH

2.2.1.2. Installing pip

If you're using a recent version of Python, the Python package manager, pip, is most likely installed by default. However, you may need to upgrade pip to the lasted version:

```
pip install --upgrade pip
```

If you need to install pip for the first time, download get-pip.py. Then run the following command to install it:

```
python get-pip.py
```

2.2.2. Installing MkDocs

Install the `mkdocs` package using pip:

```
pip install mkdocs
```

You should now have the `mkdocs` command installed on your system. Run `mkdocs --version` to check that everything worked okay.

```
$ mkdocs --version
mkdocs, version 1.2.0 from /usr/local/lib/python3.8/site-packages/mkdocs (Python 3.8)
```

> **Note**
>
> If you would like manpages installed for MkDocs, the click-man tool can generate and install them for you. Simply run the following two commands:
>
> ```
> pip install click-man
> click-man --target path/to/man/pages mkdocs
> ```
>
> See the click-man documentation for an explanation of why manpages are not automatically generated and installed by pip.

Note

If you are using Windows, some of the above commands may not work out-of-the-box.

A quick solution may be to preface every Python command with `python -m` like this:

```
python -m pip install mkdocs
python -m mkdocs
```

For a more permanent solution, you may need to edit your `PATH` environment variable to include the `Scripts` directory of your Python installation. Recent versions of Python include a script to do this for you. Navigate to your Python installation directory (for example `C:\Python38\`), open the `Tools`, then `Scripts` folder, and run the `win_add2path.py` file by double clicking on it. Alternatively, you can download the script and run it (`python win_add2path.py`).

3. Developer Guide

3.1. Developer Guide

Extending MkDocs

The MkDocs Developer Guide provides documentation for developers of third party themes and plugins. Please see the Contributing Guide for information on contributing to MkDocs itself. You can jump directly to a page listed below, or use the next and previous buttons in the navigation bar at the top of the page to move through the documentation in order.

- Themes
- Plugins

3.2. Developing Themes

A guide to creating and distributing custom themes.

> **Note**
>
> If you are looking for existing third party themes, they are listed in the MkDocs community wiki. If you want to share a theme you create, you should list it on the Wiki.

When creating a new theme, you can either follow the steps in this guide to create one from scratch or you can download the `mkdocs-basic-theme` as a basic, yet complete, theme with all the boilerplate required. **You can find this base theme on GitHub**. It contains detailed comments in the code to describe the different features and their usage.

3.2.1. Creating a custom theme

The bare minimum required for a custom theme is a `main.html` Jinja2 template file which is placed in a directory that is not a child of the docs_dir. Within `mkdocs.yml`, set the `theme.custom_dir` option to the path of the directory containing `main.html`. The path should be relative to the configuration file. For example, given this example project layout:

```
mkdocs.yml
docs/
    index.md
    about.md
custom_theme/
    main.html
    ...
```

... you would include the following settings in `mkdocs.yml` to use the custom theme directory:

```
theme:
    name: null
    custom_dir: 'custom_theme/'
```

> **Note**
>
> Generally, when building your own custom theme, the theme.name configuration setting would be set to `null`. However, if the theme.custom_dir configuration value is used in combination with an existing theme, the theme.custom_dir can be used to replace only specific parts of a built-in theme. For example, with the above layout and if you set `name: "mkdocs"` then the `main.html` file in the theme.custom_dir would replace the file of the same name in the `mkdocs` theme but otherwise the `mkdocs` theme would remain unchanged. This is useful if you want to make small adjustments to an existing theme.
>
> For more specific information, see Customizing Your Theme.

> **Warning**
>
> A theme's configuration defined in a `mkdocs_theme.yml` file is not loaded from `theme.custom_dir`. When an entire theme exists in `theme.custom_dir` and `theme.name` is set to `null`, then the entire theme configuration must be defined in the theme configuration option in the `mkdocs.yml` file.
>
> However, when a theme is packaged up for distribution, and loaded using the `theme.name` configuration option, then a `mkdocs_theme.yml` file is required for the theme.

3.2.2. Basic theme

The simplest `main.html` file is the following:

```html
<!DOCTYPE html>
<html>
  <head>
    <title>{% if page.title %}{{ page.title }} - {% endif %}{{ config.site_name }}</tit
  </head>
  <body>
    {{ page.content }}
  </body>
</html>
```

The body content from each page specified in `mkdocs.yml` is inserted using the `{{ page.content }}` tag. Style-sheets and scripts can be brought into this theme as with a normal HTML file. Navbars and tables of contents can also be generated and included automatically, through the `nav` and `toc` objects, respectively. If you wish to write your own theme, it is recommended to start with one of the built-in themes and modify it accordingly.

> **Note**
>
> As MkDocs uses Jinja as its template engine, you have access to all the power of Jinja, including template inheritance. You may notice that the themes included with MkDocs make extensive use of template inheritance and blocks, allowing users to easily override small bits and pieces of the templates from the theme `custom_dir`. Therefore, the built-in themes are implemented in a `base.html` file, which `main.html` extends. Although not required, third party template authors are encouraged to follow a similar pattern and may want to define the same blocks as are used in the built-in themes for consistency.

3.2.3. Theme Files

There are various files which a theme treats special in some way. Any other files are simply copied from the theme directory to the same path in the `site_dir` when the site it built. For example image and CSS files have no special significance and are copied as-is. Note, however, that if the user provides a file with the same path in their `docs_dir`, then the user's file will replace the theme file.

3.2.3.1. Template Files

Any files with the `.html` extension are considered to be template files and are not copied from the theme directory or any subdirectories. Also, any files listed in static_templates are treated as templates regardless of their file extension.

3.2.3.2. Theme Meta Files

The various files required for packaging a theme are also ignored. Specifically, the `mkdocs_theme.yml` configuration file and any Python files.

3.2.3.3. Dot Files

Theme authors can explicitly force MkDocs to ignore files by starting a file or directory name with a dot. Any of the following files would be ignored:

```
.ignored.txt
.ignored/file.txt
foo/.ignored.txt
foo/.ignored/file.txt
```

3.2.3.4. Documentation Files

All documentation files are ignored. Specifically, any Markdown files (using any of the file extensions supported by MKDocs). Additionally, any README files which may exist in the theme directories are ignored.

3.2.4. Template Variables

Each template in a theme is built with a template context. These are the variables that are available to themes. The context varies depending on the template that is being built. At the moment templates are either built with the global context or with a page specific context. The global context is used for HTML pages that don't represent an individual Markdown document, for example a 404.html page or search.html.

3.2.4.1. Global Context

The following variables are available globally on any template.

config

The `config` variable is an instance of MkDocs' config object generated from the `mkdocs.yml` config file. While you can use any config option, some commonly used options include:

- config.site_name
- config.site_url
- config.site_author
- config.site_description
- config.theme.locale (See also Theme Configuration below)
- config.extra_javascript
- config.extra_css
- config.repo_url
- config.repo_name
- config.copyright
- config.google_analytics

nav

The `nav` variable is used to create the navigation for the documentation. The `nav` object is an iterable of navigation objects as defined by the nav configuration setting.

In addition to the iterable of navigation objects, the `nav` object contains the following attributes:

nav.homepage

The page object for the homepage of the site.

nav.pages

A flat list of all page objects contained in the navigation. This list is not necessarily a complete list of all site pages as it does not contain pages which are not included in the navigation. This list does match the list and order of pages used for all "next page" and "previous page" links. For a list of all pages, use the pages template variable.

Nav Example

Following is a basic usage example which outputs the first and second level navigation as a nested list.

```
{% if nav|length>1 %}
    <ul>
    {% for nav_item in nav %}
        {% if nav_item.children %}
            <li>{{ nav_item.title }}
                <ul>
                {% for nav_item in nav_item.children %}
                    <li class="{% if nav_item.active%}current{% endif %}">
                        <a href="{{ nav_item.url|url }}">{{ nav_item.title }}</a>
                    </li>
                {% endfor %}
                </ul>
            </li>
        {% else %}
            <li class="{% if nav_item.active%}current{% endif %}">
                <a href="{{ nav_item.url|url }}">{{ nav_item.title }}</a>
            </li>
        {% endif %}
    {% endfor %}
    </ul>
{% endif %}
```

base_url

The `base_url` provides a relative path to the root of the MkDocs project. While this can be used directly by prepending it to a local relative URL, it is best to use the url template filter, which is smarter about how it applies `base_url`.

mkdocs_version

Contains the current MkDocs version.

build_date_utc

A Python datetime object that represents the date and time the documentation was built in UTC. This is useful for showing how recently the documentation was updated.

pages

A flat list of `File` objects for all pages in the project. This list can contain pages not included in the global navigation and may not match the order of pages within that navigation. The page object for each `File` can be accessed from `file.page`.

page

In templates which are not rendered from a Markdown source file, the `page` variable is `None`. In templates which are rendered from a Markdown source file, the `page` variable contains a `page` object. The same `page` objects are used as `page` navigation objects in the global navigation and in the pages template variable.

All `page` objects contain the following attributes:

page.title

Contains the Title for the current page.

page.content

The rendered Markdown as HTML, this is the contents of the documentation.

page.toc

An iterable object representing the Table of contents for a page. Each item in the `toc` is an `AnchorLink` which contains the following attributes:

- `AnchorLink.title` : The text of the item.
- `AnchorLink.url` : The hash fragment of a URL pointing to the item.
- `AnchorLink.level` : The zero-based level of the item.
- `AnchorLink.children` : An iterable of any child items.

The following example would display the top two levels of the Table of Contents for a page.

```
<ul>
{% for toc_item in page.toc %}
    <li><a href="{{ toc_item.url }}">{{ toc_item.title }}</a></li>
    {% for toc_item in toc_item.children %}
        <li><a href="{{ toc_item.url }}">{{ toc_item.title }}</a></li>
    {% endfor %}
{% endfor %}
</ul>
```

page.meta

A mapping of the metadata included at the top of the markdown page. In this example we define a `source` property above the page title.

```
source: generics.py
        mixins.py

# Page title

Content...
```

A template can access this metadata for the page with the `meta.source` variable. This could then be used to link to source files related to the documentation page.

```
{% for filename in page.meta.source %}
  <a class="github" href="https://github.com/.../{{ filename }}">
    <span class="label label-info">{{ filename }}</span>
  </a>
{% endfor %}
```

page.url

The URL of the page relative to the MkDocs `site_dir`. It is expected that this be used with the url filter to ensure the URL is relative to the current page.

```
<a href="{{ page.url|url }}">{{ page.title }}</a>
```

page.abs_url

The absolute URL of the page from the server root as determined by the value assigned to the site_url configuration setting. The value includes any subdirectory included in the `site_url`, but not the domain. base_url should not be used with this variable.

For example, if `site_url: https://example.com/` , then the value of `page.abs_url` for the page `foo.md` would be `/foo/` . However, if `site_url: https://example.com/bar/` , then the value of `page.abs_url` for the page `foo.md` would be `/bar/foo/` .

page.canonical_url

The full, canonical URL to the current page as determined by the value assigned to the site_url configuration setting. The value includes the domain and any subdirectory included in the `site_url` . base_url should not be used with this variable.

page.edit_url

The full URL to the source page in the source repository. Typically used to provide a link to edit the source page. base_url should not be used with this variable.

page.is_homepage

Evaluates to `True` for the homepage of the site and `False` for all other pages. This can be used in conjunction with other attributes of the `page` object to alter the behavior. For example, to display a different title on the homepage:

```
{% if not page.is_homepage %}{{ page.title }} - {% endif %}{{ site_name }}
```

page.previous_page

The page object for the previous page or `None` . The value will be `None` if the current page is the first item in the site navigation or if the current page is not included in the navigation at all. When the value is a page object, the usage is the same as for `page` .

page.next_page

The page object for the next page or `None` . The value will be `None` if the current page is the last item in the site navigation or if the current page is not included in the navigation at all. When the value is a page object, the usage is the same as for `page` .

page.parent

The immediate parent of the page in the site navigation. `None` if the page is at the top level.

page.children

Pages do not contain children and the attribute is always `None` .

page.active

When `True` , indicates that this page is the currently viewed page. Defaults to `False` .

page.is_section

Indicates that the navigation object is a "section" object. Always `False` for page objects.

page.is_page

Indicates that the navigation object is a "page" object. Always `True` for page objects.

page.is_link

Indicates that the navigation object is a "link" object. Always `False` for page objects.

3.2.4.2. Navigation Objects

Navigation objects contained in the nav template variable may be one of section objects, page objects, and link objects. While section objects may contain nested navigation objects, pages and links do not.

Page objects are the full page object as used for the current page with all of the same attributes available. Section and Link objects contain a subset of those attributes as defined below:

Section

A `section` navigation object defines a named section in the navigation and contains a list of child navigation objects. Note that sections do not contain URLs and are not links of any kind. However, by default, MkDocs sorts index pages to the top and the first child might be used as the URL for a section if a theme chooses to do so.

The following attributes are available on `section` objects:

section.title
The title of the section.

section.parent
The immediate parent of the section or `None` if the section is at the top level.

section.children
An iterable of all child navigation objects. Children may include nested sections, pages and links.

section.active
When `True`, indicates that a child page of this section is the current page and can be used to highlight the section as the currently viewed section. Defaults to `False`.

section.is_section

Indicates that the navigation object is a "section" object. Always `True` for section objects.

section.is_page

Indicates that the navigation object is a "page" object. Always `False` for section objects.

section.is_link

Indicates that the navigation object is a "link" object. Always `False` for section objects.

Link

A `link` navigation object contains a link which does not point to an internal MkDocs page. The following attributes are available on `link` objects:

link.title

The title of the link. This would generally be used as the label of the link.

link.url

The URL that the link points to. The URL should always be an absolute URLs and should not need to have `base_url` prepended.

link.parent

The immediate parent of the link. `None` if the link is at the top level.

link.children

Links do not contain children and the attribute is always `None`.

link.active

External links cannot be "active" and the attribute is always `False`.

link.is_section

Indicates that the navigation object is a "section" object. Always `False` for link objects.

link.is_page

Indicates that the navigation object is a "page" object. Always `False` for link objects.

link.is_link

Indicates that the navigation object is a "link" object. Always `True` for link objects.

3.2.4.3. Extra Context

Additional variables can be passed to the template with the `extra` configuration option. This is a set of key value pairs that can make custom templates far more flexible.

For example, this could be used to include the project version of all pages and a list of links related to the project. This can be achieved with the following `extra` configuration:

```yaml
extra:
    version: 0.13.0
    links:
        - https://github.com/mkdocs
        - https://docs.readthedocs.org/en/latest/builds.html#mkdocs
        - https://www.mkdocs.org/
```

And then displayed with this HTML in the custom theme.

```
{{ config.extra.version }}

{% if config.extra.links %}
  <ul>
  {% for link in config.extra.links %}
      <li>{{ link }}</li>
  {% endfor %}
  </ul>
{% endif %}
```

3.2.5. Template Filters

In addition to Jinja's default filters, the following custom filters are available to use in MkDocs templates:

3.2.5.1. url

Normalizes a URL. Absolute URLs are passed through unaltered. If the URL is relative and the template context includes a page object, then the URL is returned relative to the page object. Otherwise, the URL is returned with base_url prepended.

```
<a href="{{ page.url|url }}">{{ page.title }}</a>
```

3.2.5.2. tojson

Safety convert a Python object to a value in a JavaScript script.

```
<script>
    var mkdocs_page_name = {{ page.title|tojson|safe }};
</script>
```

3.2.6. Search and themes

As of MkDocs version 0.17 client side search support has been added to MkDocs via the `search` plugin. A theme needs to provide a few things for the plugin to work with the theme.

While the `search` plugin is activated by default, users can disable the plugin and themes should account for this. It is recommended that theme templates wrap search specific markup with a check for the plugin:

```
{% if 'search' in config['plugins'] %}
    search stuff here...
{% endif %}
```

At its most basic functionality, the search plugin will simply provide an index file which is no more than a JSON file containing the content of all pages. The theme would need to implement its own search functionality client-side. However, with a few settings and the necessary templates, the plugin can provide a complete functioning client-side search tool based on lunr.js.

The following HTML needs to be added to the theme so that the provided JavaScript is able to properly load the search scripts and make relative links to the search results from the current page.

```
<script>var base_url = '{{ base_url }}';</script>
```

With properly configured settings, the following HTML in a template will add a full search implementation to your theme.

```
<h1 id="search">Search Results</h1>

<form action="search.html">
  <input name="q" id="mkdocs-search-query" type="text" >
</form>

<div id="mkdocs-search-results">
  Sorry, page not found.
</div>
```

The JavaScript in the plugin works by looking for the specific ID's used in the above HTML. The form input for the user to type the search query must be identified with `id="mkdocs-search-query"`

and the div where the results will be placed must be identified with
`id="mkdocs-search-results"` .

The plugin supports the following options being set in the theme's configuration file,
`mkdocs_theme.yml` :

3.2.6.1. include_search_page

Determines whether the search plugin expects the theme to provide a dedicated search page via a template located at `search/search.html` .

When `include_search_page` is set to `true` , the search template will be built and available at `search/search.html` . This method is used by the `readthedocs` theme.

When `include_search_page` is set to `false` or not defined, it is expected that the theme provide some other mechanisms for displaying search results. For example, the `mkdocs` theme displays results on any page via a modal.

3.2.6.2. search_index_only

Determines whether the search plugin should only generate a search index or a complete search solution.

When `search_index_only` is set to `false` , then the search plugin modifies the Jinja environment by adding its own `templates` directory (with a lower precedence than the theme) and adds its scripts to the `extra_javascript` config setting.

When `search_index_only` is set to `true` or not defined, the search plugin makes no modifications to the Jinja environment. A complete solution using the provided index file is the responsibility of the theme.

The search index is written to a JSON file at `search/search_index.json` in the site_dir. The JSON object contained within the file may contain up to three objects.

```
{
    config: {...},
    docs: [...],
    index: {...}
}
```

If present, the `config` object contains the key/value pairs of config options defined for the plugin in the user's `mkdocs.yml` config file under `plugings.search` . The `config` object was new in MkDocs version 1.0.

The `docs` object contains a list of document objects. Each document object is made up of a `location` (URL), a `title` , and `text` which can be used to create a search index and/or display search results.

If present, the `index` object contains a pre-built index which offers performance improvements for larger sites. Note that the pre-built index is only created if the user explicitly enables the prebuild_index config option. Themes should expect the index to not be present, but can choose to use the index when it is available. The `index` object was new in MkDocs version 1.0.

3.2.7. Packaging Themes

MkDocs makes use of Python packaging to distribute themes. This comes with a few requirements.

To see an example of a package containing one theme, see the MkDocs Bootstrap theme and to see a package that contains many themes, see the MkDocs Bootswatch theme.

> **Note**
>
> It is not strictly necessary to package a theme, as the entire theme can be contained in the `custom_dir` . If you have created a "one-off theme," that should be sufficient. However, if you intend to distribute your theme for others to use, packaging the theme has some advantages. By packaging your theme, your users can more easily install it, they can rely on a default configuration being defined, and they can then take advantage of the custom_dir to make tweaks to your theme to better suit their needs.

3.2.7.1. Package Layout

The following layout is recommended for themes. Two files at the top level directory called `MANIFEST.in` and `setup.py` beside the theme directory which contains an empty `__init__.py` file, a theme configuration file (`mkdocs_theme.yml`), and your template and media files.

```
.
|-- MANIFEST.in
|-- theme_name
|    |-- __init__.py
|    |-- mkdocs_theme.yml
|    |-- main.html
|    |-- styles.css
`-- setup.py
```

The `MANIFEST.in` file should contain the following contents but with theme_name updated and any extra file extensions added to the include.

```
recursive-include theme_name *.ico *.js *.css *.png *.html *.eot *.svg *.ttf *.woff
recursive-exclude * __pycache__
recursive-exclude * *.py[co]
```

The `setup.py` should include the following text with the modifications described below.

```python
from setuptools import setup, find_packages

VERSION = '0.0.1'

setup(
    name="mkdocs-themename",
    version=VERSION,
    url='',
    license='',
    description='',
    author='',
    author_email='',
    packages=find_packages(),
    include_package_data=True,
    entry_points={
        'mkdocs.themes': [
            'themename = theme_name',
        ]
    },
    zip_safe=False
)
```

Fill in the URL, license, description, author and author email address.

The name should follow the convention `mkdocs-themename` (like `mkdocs-bootstrap` and `mkdocs-bootswatch`), starting with MkDocs, using hyphens to separate words and including the name of your theme.

Most of the rest of the file can be left unedited. The last section we need to change is the entry_points. This is how MkDocs finds the theme(s) you are including in the package. The name on the left is the one that users will use in their mkdocs.yml and the one on the right is the directory containing your theme files.

The directory you created at the start of this section with the main.html file should contain all of the other theme files. The minimum requirement is that it includes a `main.html` for the theme. It **must** also include a `__init__.py` file which should be empty, this file tells Python that the directory is a package.

3.2.7.2. Theme Configuration

A packaged theme is required to include a configuration file named `mkdocs_theme.yml` which is placed in the root of your template files. The file should contain default configuration options for the theme. However, if the theme offers no configuration options, the file is still required and can be left blank. A theme which is not packaged does not need a `mkdocs_theme.yml` file as that file is not loaded from `theme.custom_dir`.

The theme author is free to define any arbitrary options deemed necessary and those options will be made available in the templates to control behavior. For example, a theme might want to make a sidebar optional and include the following in the `mkdocs_theme.yml` file:

```
show_sidebar: true
```

Then in a template, that config option could be referenced:

```
{% if config.theme.show_sidebar %}
<div id="sidebar">...</div>
{% endif %}
```

And the user could override the default in their project's `mkdocs.yml` config file:

```
theme:
    name: themename
    show_sidebar: false
```

In addition to arbitrary options defined by the theme, MkDocs defines a few special options which alters its behavior:

locale

This option mirrors the theme config option of the same name. If this value is not defined in the `mkdocs_theme.yml` file and the user does not set it in `mkdocs.yml` then it will default to `en` (English). The value is expected to match the language used in the text provided by the theme (such a "next" and "previous" links) and should be used as the value of the `<html>` tag's `lang` attribute. See Supporting theme localization/ translation for more information.

Note that during configuration validation, the provided string is converted to a `Locale` object. The object contains `Locale.language` and `Locale.territory` attributes and will resolve as a string from within a template. Therefore, the following will work fine:

```
<html lang="{ config.theme.locale }">
```

If the locale was set to `fr_CA` (Canadian French), then the above template would render as:

```
<html lang="fr_CA">
```

If you did not want the territory attribute to be included, then reference the `language` attribute directly:

```
<html lang="{ config.theme.locale.language }">
```

That would render as:

```
<html lang="fr">
```

static_templates

This option mirrors the theme config option of the same name and allows some defaults to be set by the theme. Note that while the user can add templates to this list, the user cannot remove templates included in the theme's config.

extends

Defines a parent theme that this theme inherits from. The value should be the string name of the parent theme. Normal Jinja inheritance rules apply.

Plugins may also define some options which allow the theme to inform a plugin about which set of plugin options it expects. See the documentation for any plugins you may wish to support in your theme.

3.2.7.3. Distributing Themes

With the above changes, your theme should now be ready to install. This can be done with pip, using `pip install .` if you are still in the same directory as the setup.py.

Most Python packages, including MkDocs, are distributed on PyPI. To do this, you should run the following command.

```
python setup.py register
```

If you don't have an account setup, you should be prompted to create one.

For a much more detailed guide, see the official Python packaging documentation for Packaging and Distributing Projects.

3.2.8. Supporting theme Localization/Translation

While the built-in themes provide support for localization/translation of templates, custom themes and third-party themes may choose not to. Regardless, the `locale` setting of the `theme` configuration option is always present and is relied upon by other parts of the system. Therefore, it is recommended that all third-party themes use the same setting for designating a language regardless of the system they use for translation. In that way, users will experience consistent behavior regardless of the theme they may choose.

The method for managing translations is up to the developers of a theme. However, if a theme developer chooses to use the same mechanisms used by the built-in themes, the sections below outline how to enable and make use of the same commands utilized by MkDocs.

3.2.8.1. Enabling the Localization/Translation commands

MkDocs includes some helper commands which are light wrappers around pybabel's commands. To use the commands on your own theme, add the following to your theme's `setup.py` script:

```
from mkdocs.commands.setup import babel_cmdclass

setup(
    ...
    cmdclass=babel_cmdclass
)
```

Note that `cmdclass=babel_cmdclass` was added an a parameter passed to the `setup` function.

> **Warning**
>
> As **pybabel is not installed by default** and most users will not have pybabel installed, theme developers and/or translators should make sure to have installed the necessary dependencies (using `pip install mkdocs[i18n]`) in order for the commands to be available for use.

3.2.8.2. Using the Localization/Translation commands

Since the translation commands are embedded in the `setup.py` script of your custom theme they should be called from the root of your theme's working tree as follows:

```
python setup.py <command_name> [OPTIONS]
```

Each command provides a detailed list of options available with the `-h/--help` option.

For an overview of the workflow used by MkDocs to translate the built-in themes, see the appropriate section of the Contributing Guide and the Translation Guide.

Default values for many of the options to the commands can be defined in a `setup.cfg` file. Create a section using the command name as the section name, and the long option name as the key. See MkDocs' own setup.cfg file for an example.

A summary of changes/additions to the behavior and options of the upstream pybabel commands are summarized below.

compile_catalog

The `-t/--theme` option has been added to this command. The `theme` specified must be a `theme` defined as a entry point in the same `setup.py` script. Other themes will not be recognized. If only one `theme` has been defined as an entry point, then that `theme` will be used as the default if none is specified by this option. If more than one `theme` is defined as entry points, then no default is set and a `theme` must be specified by this option. The command only operates on one theme at a time. Therefore, the command needs to be run once for each theme included in a package.

When a `theme` is specified, the directory of that `theme` as defined in the entry point is used to define a default value of the `-d/--directory` option. The `--directory` option is set to `{theme_dir}/locales`. If a `directory` is passed to the `--directory` option, then the `theme` option is ignored.

extract_messages

The `-t/--theme` option has been added to this command. The `theme` specified must be a `theme` defined as a entry point in the same `setup.py` script. Other themes will not be recognized. If only one `theme` has been defined as an entry point, then that `theme` will be used as the default if none is specified by this option. If more than one `theme` is defined as entry points, then no default is set and a `theme` must be specified by this option. The command only operates on one theme at a time. Therefore, the command needs to be run once for each theme included in a package.

When a `theme` is specified, the directory of that `theme` as defined in the entry point is used to define a default value for the `--input-dirs` and `--output-file` options. The `--input-dirs` option is set to the `theme` directory and `--output-file` is set to `{theme_dir}/{domain}.pot`. If a path is provided to either option, then the `theme` option is ignored for that option.

The `--domain` option has been added to this command and can be used to override the `domain` used for the `output-file` based on the `theme`. Defaults to `messages`.

The `-F/--mapping-file` option defaults to the mapping file used by MkDocs' built-in themes. However, if that mapping file does not meet your theme's needs to can override it by providing your own and passing the path of that file into the option.

init_catalog

The `-t/--theme` option has been added to this command. The `theme` specified must be a `theme` defined as a entry point in the same `setup.py` script. Other themes will not be recognized. If only one `theme` has been defined as an entry point, then that `theme` will be used as the default if none is specified by this option. If more than one `theme` is defined as entry points, then no default is set and a `theme` must be specified by this option. The command only operates on one theme at a time. Therefore, the command needs to be run once for each theme included in a package.

When a `theme` is specified, the directory of that `theme` as defined in the entry point is used to define a default value for the `-i/--input-file` and `-d/--output-dir` options. The `--input-file` option is set to `{theme_dir}/{domain}.pot` (`domain` defaults to `messages`) and `--output-dir` is set to `{theme_dir}/locales`. If a path is provided to either option, then the `theme` option is ignored for that option.

update_catalog

The `-t/--theme` option has been added to this command. The `theme` specified must be a `theme` defined as a entry point in the same `setup.py` script. Other themes will not be recognized. If only one `theme` has been defined as an entry point, then that `theme` will be used as the default if none is specified by this option. If more than one `theme` is defined as entry points, then no default is set and a `theme` must be specified by this option. The command only operates on one theme at a time. Therefore, the command needs to be run once for each theme included in a package.

When a `theme` is specified, the directory of that `theme` as defined in the entry point is used to define a default value for the `-i/--input-file` and `-d/--output-dir` options. The `--input-file` option is set to `{theme_dir}/{domain}.pot` (`domain` defaults to `messages`) and `--output-dir` is set to `{theme_dir}/locales` . If a path is provided to either option, then the `theme` option is ignored for that option.

3.2.8.3. Example custom theme Localization/Translation workflow

> **Note**
>
> If your theme inherits from an existing theme which already provides translation catalogs, your theme's translations will be merged with the parent theme's translations during a MkDocs build.
>
> This means that you only need to concentrate on the added translations. Yet, you will still benefit from the translations of the parent theme. At the same time, you may override any of parent theme's translations!

Let's suppose that you're working on your own fork of the mkdocs-basic-theme and want to add translations to it.

You would first modify the `setup.py` like this:

```
--- a/setup.py
+++ b/setup.py
@@ -1,4 +1,5 @@
 from setuptools import setup, find_packages
+from mkdocs.commands.setup import babel_cmdclass

 VERSION = '1.1'

@@ -18,5 +19,6 @@ setup(
            'basictheme = basic_theme',
        ]
    },
-    zip_safe=False
+    zip_safe=False,
+    cmdclass=babel_cmdclass
 )
```

Next, you would edit the templates by wrapping text in your HTML sources with `{% trans %}` and `{% endtrans %}` as follows:

```
--- a/basic_theme/base.html
+++ b/basic_theme/base.html
@@ -88,7 +88,7 @@

 <body>

-  <h1>This is an example theme for MkDocs.</h1>
+  <h1>{% trans %}This is an example theme for MkDocs.{% endtrans %}</h1>

   <p>
     It is designed to be read by looking at the theme HTML which is heavily
```

Then you would follow the Translation Guide as usual to get your translations running.

3.2.8.4. Packaging Translations with your theme

While the Portable Object Template (`pot`) file created by the `extract_messages` command and the Portable Object (`po`) files created by the `init_catalog` and `update_catalog` commands are useful for creating and editing translations, they are not used by MkDocs directly and do not need to be included in a packaged release of a theme. When MkDocs builds a site with translations, it only makes use of the binary `mo` files(s) for the specified locale. Therefore, when packaging a theme, you would need to make the following addition to your `MANIFEST.in` file:

```
recursive-include theme_name *.mo
```

Then, before building your Python package, you will want to ensure that the binary `mo` file for each locale is up-to-date by running the `compile_catalog` command for each locale. MkDocs expects the binary `mo` files to be located at `locales/<locale>/LC_MESSAGES/messages.mo` , which the `compile_catalog` command automatically does for you. See Testing theme translations for details.

> **Note**
>
> As outlined in our Translation Guide, the MkDocs project has chosen to include the `pot` and `po` files in our code repository, but not the `mo` files. This requires us to always run `compile_catalog` before packaging a new release regardless of whether any changes were made to a translation or not. However, you may chose an alternate workflow for your theme. At a minimum, you need to ensure that up-to-date `mo` files are included at the correct location in each release. However, you may use a different process for generating those `mo` files if you chose to do so.

4. About

4.1. Release Notes

4.1.1. Upgrading

To upgrade MkDocs to the latest version, use pip:

```
pip install -U mkdocs
```

You can determine your currently installed version using `mkdocs --version`:

```
$ mkdocs --version
mkdocs, version 1.0 from /path/to/mkdocs (Python 3.6)
```

4.1.2. Maintenance team

The current and past members of the MkDocs team.

- @tomchristie
- @d0ugal
- @waylan
- @oprypin
- @ultrabug

4.1.3. Version 1.2.3 (2021-10-12)

- Built-in themes now also support these languages:
  - Simplified Chinese (#2497)
  - Japanese (#2525)
  - Brazilian Portuguese (#2535)
  - Spanish (#2545, previously #2396)
- Third-party plugins will take precedence over built-in plugins with the same name (#2591)
- Bugfix: Fix ability to load translations for some languages: core support (#2565) and search plugin support with fallbacks (#2602)
- Bugfix (regression in 1.2): Prevent directory traversal in the dev server (#2604)

- Bugfix (regression in 1.2): Prevent webserver warnings from being treated as a build failure in strict mode (#2607)

- Bugfix: Correctly print colorful messages in the terminal on Windows (#2606)

- Bugfix: Python version 3.10 was displayed incorrectly in `--version` (#2618)

Other small improvements; see commit log.

4.1.4. Version 1.2.2 (2021-07-18)

- Bugfix (regression in 1.2): Fix serving files/paths with Unicode characters (#2464)

- Bugfix (regression in 1.2): Revert livereload file watching to use polling observer (#2477)

  This had to be done to reasonably support usages that span virtual filesystems such as non-native Docker and network mounts.

  This goes back to the polling approach, very similar to that was always used prior, meaning most of the same downsides with latency and CPU usage.

- Revert from 1.2: Remove the requirement of a `site_url` config and the restriction on `use_directory_urls` (#2490)

- Bugfix (regression in 1.2): Don't require trailing slash in the URL when serving a directory index in `mkdocs serve` server (#2507)

  Instead of showing a 404 error, detect if it's a directory and redirect to a path with a trailing slash added, like before.

- Bugfix: Fix `gh_deploy` with config-file in the current directory (#2481)

- Bugfix: Fix reversed breadcrumbs in "readthedocs" theme (#2179)

- Allow "mkdocs.yaml" as the file name when '--config' is not passed (#2478)

- Stop treating ";" as a special character in URLs: urlparse -> urlsplit (#2502)

- Improve build performance for sites with many pages (partly already done in 1.2) (#2407)

4.1.5. Version 1.2.1 (2021-06-09)

- Bugfix (regression in 1.2): Ensure 'gh-deploy' always pushes.

4.1.6. Version 1.2 (2021-06-04)

4.1.6.1. Major Additions to Version 1.2

Support added for Theme Localization (#2299)

The `mkdocs` and `readthedocs` themes now support language localization using the `theme.locale` parameter, which defaults to `en` (English). The only other supported languages in this release are `fr` (French) and `es` (Spanish). For details on using the provided translations, see the user guide. Note that translation will not happen by default. Users must first install the necessary dependencies with the following command:

```
pip install mkdocs[i18n]
```

Translation contributions are welcome and detailed in the Translation Guide.

Developers of third party themes may want to review the relevant section of the Theme Development Guide.

Contributors who are updating the templates to the built-in themes should review the Contributing Guide.

The `lang` setting of the `search` plugin now defaults to the language specified in `theme.locale`.

Support added for Environment Variables in the configuration file (#1954)

Environments variables may now be specified in the configuration file with the `!ENV` tag. The value of the variable will be parsed by the YAML parser and converted to the appropriate type.

```
somekey: !ENV VAR_NAME
otherkey: !ENV [VAR_NAME, FALLBACK_VAR, 'default value']
```

See Environment Variables in the Configuration documentation for details.

Support added for Configuration Inheritance (#2218)

A configuration file may now inherit from a parent configuration file. In the primary file set the `INHERIT` key to the relative path of the parent file.

```
INHERIT: path/to/base.yml
```

The two files will then be deep merged. See Configuration Inheritance for details.

Update `gh-deploy` command (#2170)

The vendored (and modified) copy of ghp_import has been replaced with a dependency on the upstream library. As of version 1.0.0, ghp-import includes a Python API which makes it possible to call directly.

MkDocs can now benefit from recent bug fixes and new features, including the following:

- A `.nojekyll` file is automatically included when deploying to GitHub Pages.
- The `--shell` flag is now available, which reportedly works better on Windows.
- Git author and committer environment variables should be respected (#1383).

Rework auto-reload and HTTP server for `mkdocs serve` (#2385)

`mkdocs serve` now uses a new underlying server + file watcher implementation, based on http.server from standard library and watchdog. It provides similar functionality to the previously used livereload library (which is now dropped from dependencies, along with tornado).

This makes reloads more responsive and consistent in terms of timing. Multiple rapid file changes no longer cause the site to repeatedly rebuild (issue #2061).

Almost every aspect of the server is slightly different, but actual visible changes are minor. The logging outputs are only similar to the old ones. Degradations in behavior are not expected, and should be reported if found.

Offset the local site root according to the sub-path of the `site_url` (#2424)

When using `mkdocs serve` and having the `site_url` specified as e.g. `http://example.org/sub/path/`, now the root of the locally served site becomes `http://127.0.0.1:8000/sub/path/` and all document paths are offset accordingly.

A `build_error` event was added (#2103)

Plugin developers can now use the `on_build_error` hook to execute code when an exception is raised while building the site.

See `on_build_error` in the Plugins documentation for details.

Three new exceptions: BuildError PluginError and Abort (#2103)

MkDocs now has tree new exceptions defined in `mkdocs.exceptions` : `BuildError` , `PluginError` , and `Abort` :

- `PluginError` can be raised from a plugin to stop the build and log an error message without traceback.

- `BuildError` should not be used by third-party plugins developers and is reserved for internal use only.

- `Abort` is used internally to abort the build and display an error without a traceback.

See `Handling errors` in the Plugins documentation for details.

Search Indexing Strategy configuration

Users can now specify which strategy they wish to use when indexing their site for search. A user can select between the following options:

- **full**: Adds page title, section headings, and full page text to the search index.

- **sections**: Adds page titles and section headings only to the search index.

- **titles**: Adds only the page titles to the search index.

See `Search Indexing` in the configuration documentation for details.

4.1.6.2. Backward Incompatible Changes in 1.2

- The site_url configuration option is now **required**. If it is not set, a warning will be issued. In a future release an error will be raised (#2189).

  The use_directory_urls configuration option will be forced to `false` if site_url is set to an empty string. In that case, if `use_directory_urls` is not explicitly set to `false` , a warning will be issued (#2189).

  > **Note**
  >
  > This was reverted in release 1.2.2

- The `google_analytics` configuration option is deprecated as Google appears to be phasing it out in favor of its new Google Analytics 4 property. See the documentation for your theme for alternatives which can be configured as part of your theme configuration. For example, the mkdocs and readthedocs themes have each added a new `theme.analytics.gtag` configuration option which uses the new Google Analytics 4 property. See Google's documentation on how to Upgrade to a Google Analytics 4 property. Then set `theme.analytics.gtag` to the "G-" ID and delete the `google_analytics` configuration option which contains a "UA-" ID. So long as the old

"UA-" ID and new "G-" ID are properly linked in your Google account, and you are using the "G-" ID, the data will be made available in both the old and new formats by Google Analytics. See #2252.

- A theme's files are now excluded from the list of watched files by default when using the `--livereload` server. This new default behavior is what most users need and provides better performance when editing site content. Theme developers can enable the old behavior with the `--watch-theme` option. (#2092).

- The `mkdocs` theme now removes the sidebar when printing a page. This frees up horizontal space for better rendering of content like tables (#2193).

- The `mkdocs.config.DEFAULT_SCHEMA` global variable has been replaced with the function `mkdocs.config.defaults.get_schema()`, which ensures that each instance of the configuration is unique (#2289).

- The `mkdocs.utils.warning_filter` is deprecated and now does nothing. Plugins should remove any reference to is as it may be deleted in a future release. To ensure any warnings get counted, simply log them to the `mkdocs` log (i.e.: `mkdocs.plugins.pluginname`).

- The `on_serve` event (which receives the `server` object and the `builder` function) is affected by the server rewrite. `server` is now a `mkdocs.livereload.LiveReloadServer` instead of `livereload.server.Server`. The typical action that plugins can do with these is to call `server.watch(some_dir, builder)`, which basically adds that directory to watched directories, causing the site to be rebuilt on file changes. That still works, but passing any other function to `watch` is deprecated and shows a warning. This 2nd parameter is already optional, and will accept only this exact `builder` function just for compatibility.

- The `python` method of the `plugins.search.prebuild_index` configuration option is pending deprecation as of version 1.2. It is expected that in version 1.3 it will raise a warning if used and in version 1.4 it will raise an error. Users are encouraged to use an alternate method to generate a prebuilt index for search.

- The `lunr` and `lunr[languages]` dependencies are no longer installed by default. The dependencies are only needed for the rare user who pre-builds the search index and uses the `python` option, which is now pending deprecation. If you use this feature, then you will need to manually install `lunr` and `lunr[languages]`. A warning is issued if the dependencies are needed but not installed.

4.1.6.3. Other Changes and Additions to Version 1.2

- Bugfix: Properly process navigation child items in `_get_by_type` when filtering for sections (#2203).

- Official support for Python 3.9 has been added and support for Python 3.5 has been dropped.

- Bugfix: Fixes an issue that would result in a partially cut-off navigation item in the ReadTheDocs theme (#2297).

- Structure Files object now has a `remove` method to help plugin developers manipulate the Files tree. The corresponding `src_paths` has become a property to accommodate this possible dynamic behavior. See #2305.

- Updated highlight.js to 10.5.0. See #2313.

- Bugfix: Search plugin now works with Japanese language. See #2178.

- Documentation has been refactored (#1629).

- Restore styling of tables in the `readthedocs` theme (#2028).

- Ensure `site_url` ends with a slash (#1785).

- Correct documentation of `pages` template context variable (#1736).

- The `lunr` dependency has been updated to 0.5.9, and `lunr.js` to the corresponding 2.3.9 version (#2306).

- Color is now used in log messages to identify errors, warnings and debug messages.

- Bugfix: Identify homepage when `use_directory_urls` is `False` (#2362).

4.1.7. Version 1.1.2 (2020-05-14)

- Bugfix: Normalize IP addresses and change unsupported address error to a warning (#2108).

4.1.8. Version 1.1.1 (2020-05-12)

- Bugfix: Allow compressed sitemap to be deterministic by supporting the `SOURCE_DATE_EPOCH` environment variable (#2100).

- Bugfix: Use `README.md` as `index.html` even if `use_directory_urls` is false (#2081).

- Bugfix: Ignore links which start with a backslash (#1680).

- Bugfix: Pass `builder` to the `on_serve` event so that it can be passed to `server.watch` by plugins (#1952).

- Bugfix: Use `lunr[languages]==0.5.8` to avoid `nltk` incompatibilities (#2062).

- Bugfix: Ensure wheel is Python 3 only (#2021).

- Bugfix: Clean up `dev_addr` validation and disallow `0.0.0.0` (#2022).

- Add support for `min_search_length` parameter for search plugin (#2014).

- Bugfix: `readthedocs` theme `code` colors (#2027).

4.1.9. Version 1.1 (2020-02-22)

4.1.9.1. Major Additions to Version 1.1

Support for Lunr.py as `prebuild_index` engine

Mkdocs now supports pre-building indices using Lunr.py, a pure Python implementation of Lunr.js, allowing the user to avoid installing a NodeJS environment if so desired. For more information please read the `prebuild_index` documentation.

`readthedocs` theme updated with upstream (#588 and #1374)

The `readthedocs` theme now more closely matches the upstream Sphinx theme (version 0.4.1). A number of new theme configuration settings were added which mirror the upstream configuration options. See the theme documentation for details.

Update `mkdocs` theme to Bootswatch 4.1.3 (#1563)

The `mkdocs` theme now supports all the features of Bootswatch 4.1. Additionally, 2 filenames were changed in this update. If you are using a theme which inherits from the `mkdocs` theme, the theme developer may need to update these filenames as follows.

```
css/bootstrap-custom.min.css => css/bootstrap.min.css
js/bootstrap-3.0.3.min.js => js/bootstrap.min.js
```

Improved configuration support on the command line (#1401)

The `build`, `serve`, and `gh-deploy` subcommands now support flags to control whether directory URLs should be created: `--use-directory-urls` / `--no-directory-urls`. In addition, the `gh-deploy` subcommand now supports all the configuration options that `build` and `serve` do, adding `--strict`, `--theme`, `--theme-dir`, and `--site-dir`.

Updated lunr-languages support (#1729)

The `lunr-languages` plugin has been updated to 1.4.0, adding support for Arabic (`ar`) and Vietnamese (`vi`) languages. In addition, the Dutch and Japanese language codes have been changed to their standard values: `nl` and `ja`, respectively. The old language codes (`du` and `jp`) remain as aliases but may be removed in a future version of MkDocs.

4.1.9.2. Other Changes and Additions to Version 1.1

- Bugfix: Ensure nested dot files in themes are ignored and document behavior (#1981).

- Update minimum dependency to Markdown 3.2.1.
- Updated minimum dependency to Jinja 2.10.1 to address security concerns (#1780).
- Update to lunr.js 2.3.8 (#1989).
- Add support for Python 3.8.
- Drop support for Python 3.4.
- Drop support for Python 2.7. MkDocs is PY3 only now (#1926).
- Bugfix: Select appropriate asyncio event loop on Windows for Python 3.8+ (#1885).
- Bugfix: Ensure nested index pages do not get identified as the homepage (#1919).
- Bugfix: Properly identify deployment version (#1879).
- Bugfix: Properly build `ValidationError` message for `custom_dir` (#1849).
- Bugfix: Exclude Markdown files and READMEs from theme (#1766).
- Bugfix: Account for encoded URLs (#1670).
- Bugfix: Ensure theme files do not override `docs_dir` files (#1671).
- Bugfix: Do not normalize URL fragments (#1655).
- Bugfix: Skip external URLs in sitemap.xml (#1742).
- Bugfix: Ensure theme files do not override docs_dir files on Windows (#1876)
- Add canonical tag to `readthedocs` theme (#1669).
- Improved error message for when `git` is not available.
- Add support for `nav_style` theme option for the `mkdocs` theme (#1930).
- Bugfix: Long/nested dropdowns now behave more consistently for the `mkdocs` theme (#1234).
- Bugfix: Multi-row nav headers in the `mkdocs` theme no longer obscure the document content (#716).
- Add support for `navigation_depth` theme option for the `mkdocs` theme (#1970).
- `level` attribute in `page.toc` items is now 1-indexed to match the level in `<hN>` tags (#1970).

4.1.10. Version 1.0.4 (2018-09-07)

- Bugfix: Ignore absolute links in Markdown (#1621).

4.1.11. Version 1.0.3 (2018-08-29)

- Bugfix: Warn on relative paths in navigation (#1604).

- Bugfix: Handle empty `theme_config.yml` files correctly (#1602).

4.1.12. Version 1.0.2 (2018-08-22)

- Bugfix: Provide absolute `base_url` to error templates (#1598).

4.1.13. Version 1.0.1 (2018-08-13)

- Bugfix: Prevent page reload when [Enter] is pressed in search box (#1589).

- Bugfix: Avoid calling `search` until all assets are ready (#1584).

- Bugfix: Exclude `README.md` if `index.md` is present (#1580).

- Bugfix: Fix `readthedocs` theme navigation bug with homepage (#1576).

4.1.14. Version 1.0 (2018-08-03)

4.1.14.1. Major Additions to Version 1.0

Internal Refactor of Pages, Files, and Navigation

Internal handling of pages, files and navigation has been completely refactored. The changes included in the refactor are summarized below.

- Support for hidden pages. All Markdown pages are now included in the build regardless of whether they are included in the navigation configuration (#699).

- The navigation can now include links to external sites (#989 #1373 & #1406).

- Page data (including titles) is properly determined for all pages before any page is rendered (#1347).

- Automatically populated navigation now sorts index pages to the top. In other words, The index page will be listed as the first child of a directory, while all other documents are sorted alphanumerically by file name after the index page (#73 & #1042).

- A `README.md` file is now treated as an index file within a directory and will be rendered to `index.html` (#608).

- The URLs for all files are computed once and stored in a files collection. This ensures all internal links are always computed correctly regardless of the configuration. This also allows all internal links to be validated, not just links to other Markdown pages. (#842 & #872).

- A new url template filter smartly ensures all URLs are relative to the current page (#1526).

- An on_files plugin event has been added, which could be used to include files not in the `docs_dir` , exclude files, redefine page URLs (i.e. implement extensionless URLs), or to manipulate files in various other ways.

Backward Incompatible Changes

As part of the internal refactor, a number of backward incompatible changes have been introduced, which are summarized below.

URLs have changed when `use_directory_urls` **is** `False`

Previously, all Markdown pages would be have their filenames altered to be index pages regardless of how the use_directory_urls setting was configured. However, the path munging is only needed when `use_directory_urls` is set to `True` (the default). The path mangling no longer happens when `use_directory_urls` is set to `False` , which will result in different URLs for all pages that were not already index files. As this behavior only effects a non-default configuration, and the most common user-case for setting the option to `False` is for local file system (`file://`) browsing, its not likely to effect most users. However, if you have `use_directory_urls` set to `False` for a MkDocs site hosted on a web server, most of your URLs will now be broken. As you can see below, the new URLs are much more sensible.

Markdown file	Old URL	New URL
index.md	index.html	index.html
foo.md	foo/index.html	foo.html
foo/bar.md	foo/bar/index.html	foo/bar.html

Table 2 - Old and New ULR

Note that there has been no change to URLs or file paths when `use_directory_urls` is set to `True` (the default), except that MkDocs more consistently includes an ending slash on all internally generated URLs.

The `pages` configuration setting has been renamed to `nav`

The `pages` configuration setting is deprecated and will issue a warning if set in the configuration file. The setting has been renamed `nav` . To update your configuration, simply rename the setting to `nav` . In other words, if your configuration looked like this:

```
pages:
    - Home: index.md
    - User Guide: user-guide.md
```

Simply edit the configuration as follows:

```
nav:
    - Home: index.md
    - User Guide: user-guide.md
```

Listing 1 - `pages` become `nav`

In the current release, any configuration which includes a `pages` setting, but no `nav` setting, the `pages` configuration will be copied to `nav` and a warning will be issued. However, in a future release, that may no longer happen. If both `pages` and `nav` are defined, the `pages` setting will be ignored.

Template variables and `base_url`

In previous versions of MkDocs some URLs expected the base_url template variable to be prepended to the URL and others did not. That inconsistency has been removed in that no URLs are modified before being added to the template context.

For example, a theme template might have previously included a link to the `site_name` as:

```
<a href="{{ nav.homepage.url }}">{{ config.site_name }}</a>
```

And MkDocs would magically return a URL for the homepage which was relative to the current page. That "magic" has been removed and the url template filter should be used:

```
<a href="{{ nav.homepage.url|url }}">{{ config.site_name }}</a>
```

This change applies to any navigation items and pages, as well as the `page.next_page` and `page.previous_page` attributes. For the time being, the `extra_javascript` and `extra_css` variables continue to work as previously (without the `url` template filter), but they have been deprecated and the corresponding configuration values (`config.extra_javascript` and `config.extra_css` respectively) should be used with the filter instead.

```
{% for path in config['extra_css'] %}
    <link href="{{ path|url }}" rel="stylesheet">
{% endfor %}
```

Note that navigation can now include links to external sites. Obviously, the `base_url` should not be prepended to these items. However, the `url` template filter is smart enough to recognize the URL is absolute and does not alter it. Therefore, all navigation items can be passed to the filter and only those that need to will be altered.

```
{% for nav_item in nav %}
    <a href="{{ nav_item.url|url }}">{{ nav_item.title }}</a>
{% endfor %}
```

Path Based Settings are Relative to Configuration File (#543)

Previously any relative paths in the various configuration options were resolved relative to the current working directory. They are now resolved relative to the configuration file. As the documentation has always encouraged running the various MkDocs commands from the directory that contains the configuration file (project root), this change will not affect most users. However, it will make it much easier to implement automated builds or otherwise run commands from a location other than the project root.

Simply use the `-f/--config-file` option and point it at the configuration file:

```
mkdocs build --config-file /path/to/my/config/file.yml
```

As previously, if no file is specified, MkDocs looks for a file named `mkdocs.yml` in the current working directory.

Added support for YAML Meta-Data (#1542)

Previously, MkDocs only supported MultiMarkdown style meta-data, which does not recognize different data types and is rather limited. MkDocs now also supports YAML style meta-data in Markdown documents. MkDocs relies on the the presence or absence of the deliminators (`---` or `...`) to determine whether YAML style meta-data or MultiMarkdown style meta-data is being used.

Previously MkDocs would recognize MultiMarkdown style meta-data between the deliminators. Now, if the deliminators are detected, but the content between the deliminators is not valid YAML meta-data, MkDocs does not attempt to parse the content as MultiMarkdown style meta-data. Therefore, MultiMarkdown's style meta-data must not include the deliminators. See the MultiMarkdown style meta-data documentation for details.

Prior to version 0.17, MkDocs returned all meta-data values as a list of strings (even a single line would return a list of one string). In version 0.17, that behavior was changed to return each value as a single string (multiple lines were joined), which some users found limiting (see #1471). That behavior continues for MultiMarkdown style meta-data in the current version. However, YAML style meta-data supports the full range of "safe" YAML data types. Therefore, it is recommended that any complex meta-data make use of the YAML style (see the YAML style meta-data documentation for details). In fact, a future version of MkDocs may deprecate support for MultiMarkdown style meta-data.

Refactor Search Plugin

The search plugin has been completely refactored to include support for the following features:

- Use a web worker in the browser with a fallback (#1396).

- Optionally pre-build search index locally (#859 & #1061).

- Upgrade to lunr.js 2.x (#1319).

- Support search in languages other than English (#826).

- Allow the user to define the word separators (#867).

- Only run searches for queries of length > 2 (#1127).

- Remove dependency on require.js (#1218).

- Compress the search index (#1128).

Users can review the configuration options available and theme authors should review how search and themes interact.

`theme_dir` Configuration Option fully Deprecated

As of version 0.17, the custom_dir option replaced the deprecated `theme_dir` option. If users had set the `theme_dir` option, MkDocs version 0.17 copied the value to the `theme.custom_dir` option and a warning was issued. As of version 1.0, the value is no longer copied and an error is raised.

4.1.14.2. Other Changes and Additions to Version 1.0

- Keyboard shortcuts changed to not conflict with commonly used accessibility shortcuts (#1502.)

- User friendly YAML parse errors (#1543).

- Officially support Python 3.7.

- A missing theme configuration file now raises an error.

- Empty `extra_css` and `extra_javascript` settings no longer raise a warning.

- Add highlight.js configuration settings to built-in themes (#1284).

- Close search modal when result is selected (#1527).

- Add a level attribute to AnchorLinks (#1272).

- Add MkDocs version check to gh-deploy script (#640).

- Improve Markdown extension error messages. (#782).

- Drop official support for Python 3.3 and set `tornado>=5.0` (#1427).

- Add support for GitLab edit links (#1435).

- Link to GitHub issues from release notes (#644).

- Expand {sha} and {version} in gh-deploy commit message (#1410).

- Compress `sitemap.xml` (#1130).

- Defer loading JS scripts (#1380).

- Add a title attribute to the search input (#1379).

- Update RespondJS to latest version (#1398).

- Always load Google Analytics over HTTPS (#1397).

- Improve scrolling frame rate (#1394).

- Provide more version info. (#1393).

- Refactor `writing-your-docs.md` (#1392).

- Workaround Safari bug when zooming to < 100% (#1389).

- Remove addition of `clicky` class to body and animations. (#1387).

- Prevent search plugin from re-injecting `extra_javascript` files (#1388).

- Refactor `copy_media_files` util function for more flexibility (#1370).

- Remove PyPI Deployment Docs (#1360).

- Update links to Python-Markdown library (#1360).

- Document how to generate manpages for MkDocs commands (#686).

4.1.15. Version 0.17.5 (2018-07-06)

- Bugfix: Fix Python 3.7 and PEP 479 incompatibility (#1518).

4.1.16. Version 0.17.4 (2018-06-08)

- Bugfix: Add multi-level nesting support to sitemap.xml (#1482).

4.1.17. Version 0.17.3 (2018-03-07)

- Bugfix: Set dependency `tornado>=4.1,<5.0` due to changes in 5.0 (#1428).

4.1.18. Version 0.17.2 (2017-11-15)

- Bugfix: Correct `extra_*` config setting regressions (#1335 & #1336).

4.1.19. Version 0.17.1 (2017-10-30)

- Bugfix: Support `repo_url` with missing ending slash. (#1321).
- Bugfix: Add length support to `mkdocs.toc.TableOfContext` (#1325).
- Bugfix: Add some theme specific settings to the search plugin for third party themes (#1316).
- Bugfix: Override `site_url` with `dev_addr` on local server (#1317).

4.1.20. Version 0.17.0 (2017-10-19)

4.1.20.1. Major Additions to Version 0.17.0

Plugin API. (#206)

A new Plugin API has been added to MkDocs which allows users to define their own custom behaviors. See the included documentation for a full explanation of the API.

The previously built-in search functionality has been removed and wrapped in a plugin (named "search") with no changes in behavior. When MkDocs builds, the search index is now written to `search/search_index.json` instead of `mkdocs/search_index.json`. If no plugins setting is defined in the config, then the `search` plugin will be included by default. See the configuration documentation for information on overriding the default.

Theme Customization. (#1164)

Support had been added to provide theme specific customizations. Theme authors can define default options as documented in Theme Configuration. A theme can now inherit from another theme, define various static templates to be rendered, and define arbitrary default variables to control behavior in the templates. The theme configuration is defined in a configuration file named `mkdocs_theme.yml` which should be placed at the root of your template files. A warning will be raised if no configuration file is found and an error will be raised in a future release.

Users can override those defaults under the theme configuration option of their `mkdocs.yml` configuration file, which now accepts nested options. One such nested option is the custom_dir option, which replaces the now deprecated `theme_dir` option. If users had previously set the `theme_dir` option, a warning will be issued, with an error expected in a future release.

If a configuration previously defined a `theme_dir` like this:

```
theme: mkdocs
theme_dir: custom
```

Then the configuration should be adjusted as follows:

```
theme:
    name: mkdocs
    custom_dir: custom
```

See the theme configuration option documentation for details.

Previously deprecated Template variables removed. (#1168)

Page Template

The primary entry point for page templates has been changed from `base.html` to `main.html`. This allows `base.html` to continue to exist while allowing users to override `main.html` and extend `base.html`. For version 0.16, `base.html` continued to work if no `main.html` template existed, but it raised a deprecation warning. In version 1.0, a build will fail if no `main.html` template exists.

Context Variables

Page specific variable names in the template context have been refactored as defined in Custom Themes. The old variable names issued a warning in version 0.16, but have been removed in version 1.0.

Any of the following old page variables should be updated to the new ones in user created and third-party templates:

Old Variable Name	New Variable Name
current_page	page
page_title	page.title
content	page.content
toc	page.toc
meta	page.meta
canonical_url	page.canonical_url
previous_page	page.previous_page
next_page	page.next_page

Additionally, a number of global variables have been altered and/or removed and user created and third-party templates should be updated as outlined below:

Old Variable Name	New Variable Name or Expression
current_page	page
include_nav	nav|length>1
include_next_prev	(page.next_page or page.previous_page)
site_name	config.site_name
site_author	config.site_author
page_description	config.site_description
repo_url	config.repo_url
repo_name	config.repo_name
site_url	config.site_url
copyright	config.copyright
google_analytics	config.google_analytics
homepage_url	nav.homepage.url
favicon	{{ base_url }}/img/favicon.ico

Auto-Populated extra_css and extra_javascript Fully Deprecated. (#986)

In previous versions of MkDocs, if the `extra_css` or `extra_javascript` config settings were empty, MkDocs would scan the `docs_dir` and auto-populate each setting with all of the CSS and JavaScript files found. On version 0.16 this behavior was deprecated and a warning was issued. In 0.17 any unlisted CSS and JavaScript files will not be included in the HTML templates, however, a warning will be issued. In other words, they will still be copied to the `site-dir`, but they will not have any effect on the theme if they are not explicitly listed.

All CSS and JavaScript files in the `docs_dir` should be explicitly listed in the `extra_css` or `extra_javascript` config settings going forward.

4.1.20.2. Other Changes and Additions to Version 0.17.0

- Add "edit Link" support to MkDocs theme (#1129)

- Open files with `utf-8-sig` to account for BOM (#1186)

- Symbolic links are now followed consistently (#1134)

- Support for keyboard navigation shortcuts added to included themes (#1095)

- Some refactoring and improvements to config_options (#1296)

- Officially added support for Python 3.6 (#1296)

- 404 Error page added to readthedocs theme (#1296))

- Internal refactor of Markdown processing (#713)

- Removed special error message for mkdocs-bootstrap and mkdocs-bootswatch themes (#1168)

- The legacy pages config is no longer supported (#1168)

- The deprecated `json` command has been removed (#481)

- Support for Python 2.6 has been dropped (#165)

- File permissions are no longer copied during build (#1292)

- Support query and fragment strings in `edit_uri` (#1224 & #1273)

4.1.21. Version 0.16.3 (2017-04-04)

- Fix error raised by autoscrolling in the readthedocs theme (#1177)

- Fix a few documentation typos (#1181 & #1185)

- Fix a regression to livereload server introduced in 0.16.2 (#1174)

4.1.22. Version 0.16.2 (2017-03-13)

- System root (`/`) is not a valid path for site_dir or docs_dir (#1161)

- Refactor readthedocs theme navigation (#1155 & #1156)

- Add support to dev server to serve custom error pages (#1040)

- Ensure nav.homepage.url is not blank on error pages (#1131)

- Increase livereload dependency to 2.5.1 (#1106)

4.1.23. Version 0.16.1 (2016-12-22)

- Ensure scrollspy behavior does not affect nav bar (#1094)

- Only "load" a theme when it is explicitly requested by the user (#1105)

4.1.24. Version 0.16 (2016-11-04)

4.1.24.1. Major Additions to Version 0.16.0

Template variables refactored. (#874)

Page Context

Page specific variable names in the template context have been refactored as defined in Custom Themes. The old variable names will issue a warning but continue to work for version 0.16, but may be removed in a future version.

Any of the following old page variables should be updated to the new ones in user created and third-party templates:

Old Variable Name	New Variable Name
current_page	page
page_title	page.title
content	page.content
toc	page.toc
meta	page.meta
canonical_url	page.canonical_url
previous_page	page.previous_page
next_page	page.next_page

Global Context

Additionally, a number of global variables have been altered and/or deprecated and user created and third-party templates should be updated as outlined below:

Previously, the global variable `include_nav` was altered programmatically based on the number of pages in the nav. The variable will issue a warning but continue to work for version 0.16, but may be removed in a future version. Use `{% if nav|length>1 %}` instead.

Previously, the global variable `include_next_prev` was altered programmatically based on the number of pages in the nav. The variable will issue a warning but continue to work for version 0.16, but may be removed in a future version. Use `{% if page.next_page or page.previous_page %}` instead.

Previously the global variable `page_description` was altered programmatically based on whether the current page was the homepage. Now it simply maps to `config['site_description']`. Use `{% if page.is_homepage %}` in the template to conditionally change the description.

The global variable `homepage_url` maps directly to `nav.homepage.url` and is being deprecated. The variable will issue a warning but continue to work for version 0.16, but may be removed in a future version. Use `nav.homepage.url` instead.

The global variable `favicon` maps to the configuration setting `site_favicon`. Both the template variable and the configuration setting are being deprecated and will issue a warning but continue to work for version 0.16, and may be removed in a future version. Use `{{ base_url }}/img/favicon.ico` in your template instead. Users can simply save a copy of their custom favicon icon to `img/favicon.ico` in either their `docs_dir` or `theme_dir`.

A number of variables map directly to similarly named variables in the `config`. Those variables are being deprecated and will issue a warning but continue to work for version 0.16, but may be removed in a future version. Use `config.var_name` instead, where `var_name` is the name of one of the configuration variables.

Below is a summary of all of the changes made to the global context:

Old Variable Name	New Variable Name or Expression
current_page	page
include_nav	nav|length>1
include_next_prev	(page.next_page or page.previous_page)
site_name	config.site_name
site_author	config.site_author
page_description	config.site_description
repo_url	config.repo_url
repo_name	config.repo_name

Old Variable Name	New Variable Name or Expression
site_url	config.site_url
copyright	config.copyright
google_analytics	config.google_analytics
homepage_url	nav.homepage.url
favicon	{{ base_url }}/img/favicon.ico

Increased Template Customization. (#607)

The built-in themes have been updated by having each of their many parts wrapped in template blocks which allow each individual block to be easily overridden using the `theme_dir` config setting. Without any new settings, you can use a different analytics service, replace the default search function, or alter the behavior of the navigation, among other things. See the relevant documentation for more details.

To enable this feature, the primary entry point for page templates has been changed from `base.html` to `main.html`. This allows `base.html` to continue to exist while allowing users to override `main.html` and extend `base.html`. For version 0.16, `base.html` will continue to work if no `main.html` template exists, but it is deprecated and will raise a warning. In version 1.0, a build will fail if no `main.html` template exists. Any custom and third party templates should be updated accordingly.

The easiest way for a third party theme to be updated would be to simply add a `main.html` file which only contains the following line:

```
{% extends "base.html" %}
```

That way, the theme contains the `main.html` entry point, and also supports overriding blocks in the same manner as the built-in themes. Third party themes are encouraged to wrap the various pieces of their templates in blocks in order to support such customization.

Auto-Populated `extra_css` and `extra_javascript` Deprecated. (#986)

In previous versions of MkDocs, if the `extra_css` or `extra_javascript` config settings were empty, MkDocs would scan the `docs_dir` and auto-populate each setting with all of the CSS and JavaScript files found. This behavior is deprecated and a warning will be issued. In the next release, the auto-populate feature will stop working and any unlisted CSS and JavaScript files will not be

included in the HTML templates. In other words, they will still be copied to the `site-dir` , but they will not have any effect on the theme if they are not explicitly listed.

All CSS and JavaScript files in the `docs_dir` should be explicitly listed in the `extra_css` or `extra_javascript` config settings going forward.

Support for dirty builds. (#990)

For large sites the build time required to create the pages can become problematic, thus a "dirty" build mode was created. This mode simply compares the modified time of the generated HTML and source markdown. If the markdown has changed since the HTML then the page is re-constructed. Otherwise, the page remains as is. This mode may be invoked in both the `mkdocs serve` and `mkdocs build` commands:

```
mkdocs serve --dirtyreload
```

```
mkdocs build --dirty
```

It is important to note that this method for building the pages is for development of content only, since the navigation and other links do not get updated on other pages.

Stricter Directory Validation

Previously, a warning was issued if the `site_dir` was a child directory of the `docs_dir` . This now raises an error. Additionally, an error is now raised if the `docs_dir` is set to the directory which contains your config file rather than a child directory. You will need to rearrange you directory structure to better conform with the documented layout.

4.1.24.2. Other Changes and Additions to Version 0.16.0

- Bugfix: Support `gh-deploy` command on Windows with Python 3 (#722)
- Bugfix: Include .woff2 font files in Python package build (#894)
- Various updates and improvements to Documentation Home Page/Tutorial (#870)
- Bugfix: Support livereload for config file changes (#735)
- Bugfix: Non-media template files are no longer copied with media files (#807)
- Add a flag (-e/--theme-dir) to specify theme directory with the commands `mkdocs build` and `mkdocs serve` (#832)
- Fixed issues with Unicode file names under Windows and Python 2. (#833)
- Improved the styling of in-line code in the MkDocs theme. (#718)

- Bugfix: convert variables to JSON when being passed to JavaScript (#850)

- Updated the ReadTheDocs theme to match the upstream font sizes and colors more closely. (#857)

- Fixes an issue with permalink markers showing when the mouse was far above them (#843)

- Bugfix: Handle periods in directory name when automatically creating the pages config. (#728)

- Update searching to Lunr 0.7, which comes with some performance enhancements for larger documents (#859)

- Bugfix: Support SOURCE_DATE_EPOCH environment variable for "reproducible" builds (#938)

- Follow links when copying media files (#869).

- Change "Edit on..." links to point directly to the file in the source repository, rather than to the root of the repository (#975), configurable via the new `edit_uri` setting.

- Bugfix: Don't override config value for strict mode if not specified on CLI (#738).

- Add a `--force` flag to the `gh-deploy` command to force the push to the repository (#973).

- Improve alignment for current selected menu item in readthedocs theme (#888).

- `http://user.github.io/repo` => `https://user.github.io/repo/` (#1029).

- Improve installation instructions (#1028).

- Account for wide tables and consistently wrap inline code spans (#834).

- Bugfix: Use absolute URLs in nav & media links from error templates (#77).

4.1.25. Version 0.15.3 (2016-02-18)

- Improve the error message the given theme can't be found.

- Fix an issue with relative symlinks (#639)

4.1.26. Version 0.15.2 (2016-02-08)

- Fix an incorrect warning that states external themes will be removed from MkDocs.

4.1.27. Version 0.15.1 (2016-01-30)

- Lower the minimum supported Click version to 3.3 for package maintainers. (#763)

4.1.28. Version 0.15.0 (2016-01-21)

4.1.28.1. Major Additions to Version 0.15.0

Add support for installable themes

MkDocs now supports themes that are distributed via Python packages. With this addition, the Bootstrap and Bootswatch themes have been moved to external git repositories and python packages. See their individual documentation for more details about these specific themes.

- MkDocs Bootstrap

- MkDocs Bootswatch

They will be included with MkDocs by default until a future release. After that they will be installable with pip: `pip install mkdocs-bootstrap` and `pip install mkdocs-bootswatch`

See the documentation for Customizing Your Theme for more information about using and customizing themes and Custom themes for creating and distributing new themes

4.1.28.2. Other Changes and Additions to Version 0.15.0

- Fix issues when using absolute links to Markdown files. (#628)

- Deprecate support of Python 2.6, pending removal in 1.0.0. (#165)

- Add official support for Python version 3.5.

- Add support for site_description and site_author to the ReadTheDocs theme. (#631)

- Update FontAwesome to 4.5.0. (#789)

- Increase IE support with X-UA-Compatible. (#785)

- Added support for Python's `-m` flag. (#706)

- Bugfix: Ensure consistent ordering of auto-populated pages. (#638)

- Bugfix: Scroll the tables of contents on the MkDocs theme if it is too long for the page. (#204)

- Bugfix: Add all ancestors to the page attribute `ancestors` rather than just the initial one. (#693)

- Bugfix: Include HTML in the build output again. (#691)

- Bugfix: Provide filename to Read the Docs. (#721 and RTD#1480)

- Bugfix: Silence Click's unicode_literals warning. (#708)

4.1.29. Version 0.14.0 (2015-06-09)

- Improve Unicode handling by ensuring that all config strings are loaded as Unicode. (#592)

- Remove dependency on the six library. (#583)

- Remove dependency on the ghp-import library. (#547)

- Add `--quiet` and `--verbose` options to all sub-commands. (#579)

- Add short options (`-a`) to most command line options. (#579)

- Add copyright footer for readthedocs theme. (#568)

- If the requested port in `mkdocs serve` is already in use, don't show the user a full stack trace. (#596)

- Bugfix: Fix a JavaScript encoding problem when searching with spaces. (#586)

- Bugfix: gh-deploy now works if the mkdocs.yml is not in the git repo root. (#578)

- Bugfix: Handle (pass-through instead of dropping) HTML entities while parsing TOC. (#612)

- Bugfix: Default extra_templates to an empty list, don't automatically discover them. (#616)

4.1.30. Version 0.13.3 (2015-06-02)

- Bugfix: Reduce validation error to a warning if the site_dir is within the docs_dir as this shouldn't cause any problems with building but will inconvenience users building multiple times. (#580)

4.1.31. Version 0.13.2 (2015-05-30)

- Bugfix: Ensure all errors and warnings are logged before exiting. (#536)

- Bugfix: Fix compatibility issues with ReadTheDocs. (#554)

4.1.32. Version 0.13.1 (2015-05-27)

- Bugfix: Fix a problem with minimal configurations which only contain a list of paths in the pages config. (#562)

4.1.33. Version 0.13.0 (2015-05-26)

4.1.33.1. Deprecations to Version 0.13.0

Deprecate the JSON command

In this release the `mkdocs json` command has been marked as deprecated and when used a deprecation warning will be shown. It will be removed in a future release of MkDocs, version 1.0 at the latest. The `mkdocs json` command provided a convenient way for users to output the documentation contents as JSON files but with the additions of search to MkDocs this functionality is duplicated.

A new index with all the contents from a MkDocs build is created in the site_dir, so with the default value for the `site_dir` It can be found in `site/mkdocs/search_index.json` .

This new file is created on every MkDocs build (with `mkdocs build`) and no configuration is needed to enable it.

Change the pages configuration

Provide a new way to define pages, and specifically nested pages, in the mkdocs.yml file and deprecate the existing approach, support will be removed with MkDocs 1.0.

Warn users about the removal of builtin themes

All themes other than mkdocs and readthedocs will be moved into external packages in a future release of MkDocs. This will enable them to be more easily supported and updates outside MkDocs releases.

4.1.33.2. Major Additions to Version 0.13.0

Search

Support for search has now been added to MkDocs. This is based on the JavaScript library lunr.js. It has been added to both the `mkdocs` and `readthedocs` themes. See the custom theme documentation on supporting search for adding it to your own themes.

New Command Line Interface

The command line interface for MkDocs has been re-written with the Python library Click. This means that MkDocs now has an easier to use interface with better help output.

This change is partially backwards incompatible as while undocumented it was possible to pass any configuration option to the different commands. Now only a small subset of the configuration

options can be passed to the commands. To see in full commands and available arguments use `mkdocs --help` and `mkdocs build --help` to have them displayed.

Support Extra HTML and XML files

Like the extra_javascript and extra_css configuration options, a new option named extra_templates has been added. This will automatically be populated with any `.html` or `.xml` files in the project docs directory.

Users can place static HTML and XML files and they will be copied over, or they can also use Jinja2 syntax and take advantage of the global variables.

By default MkDocs will use this approach to create a sitemap for the documentation.

4.1.33.3. Other Changes and Additions to Version 0.13.0

- Add support for Markdown extension configuration options. (#435)

- MkDocs now ships Python wheels. (#486)

- Only include the build date and MkDocs version on the homepage. (#490)

- Generate sitemaps for documentation builds. (#436)

- Add a clearer way to define nested pages in the configuration. (#482)

- Add an extra config option for passing arbitrary variables to the template. (#510)

- Add `--no-livereload` to `mkdocs serve` for a simpler development server. (#511)

- Add copyright display support to all themes (#549)

- Add support for custom commit messages in a `mkdocs gh-deploy` (#516)

- Bugfix: Fix linking to media within the same directory as a markdown file called index.md (#535)

- Bugfix: Fix errors with Unicode filenames (#542).

4.1.34. Version 0.12.2 (2015-04-22)

- Bugfix: Fix a regression where there would be an error if some child titles were missing but others were provided in the pages config. (#464)

4.1.35. Version 0.12.1 (2015-04-14)

- Bugfix: Fixed a CSS bug in the table of contents on some browsers where the bottom item was not clickable.

4.1.36. Version 0.12.0 (2015-04-14)

- Display the current MkDocs version in the CLI output. (#258)

- Check for CNAME file when using gh-deploy. (#285)

- Add the homepage back to the navigation on all themes. (#271)

- Add a strict more for local link checking. (#279)

- Add Google analytics support to all themes. (#333)

- Add build date and MkDocs version to the ReadTheDocs and MkDocs theme outputs. (#382)

- Standardize highlighting across all themes and add missing languages. (#387)

- Add a verbose flag. (-v) to show more details about what the build. (#147)

- Add the option to specify a remote branch when deploying to GitHub. This enables deploying to GitHub pages on personal and repo sites. (#354)

- Add favicon support to the ReadTheDocs theme HTML. (#422)

- Automatically refresh the browser when files are edited. (#163)

- Bugfix: Never re-write URLs in code blocks. (#240)

- Bugfix: Don't copy dotfiles when copying media from the `docs_dir`. (#254)

- Bugfix: Fix the rendering of tables in the ReadTheDocs theme. (#106)

- Bugfix: Add padding to the bottom of all bootstrap themes. (#255)

- Bugfix: Fix issues with nested Markdown pages and the automatic pages configuration. (#276)

- Bugfix: Fix a URL parsing error with GitHub enterprise. (#284)

- Bugfix: Don't error if the mkdocs.yml is completely empty. (#288)

- Bugfix: Fix a number of problems with relative URLs and Markdown files. (#292)

- Bugfix: Don't stop the build if a page can't be found, continue with other pages. (#150)

- Bugfix: Remove the site_name from the page title, this needs to be added manually. (#299)

- Bugfix: Fix an issue with table of contents cutting off Markdown. (#294)

- Bugfix: Fix hostname for BitBucket. (#339)

- Bugfix: Ensure all links end with a slash. (#344)

- Bugfix: Fix repo links in the readthedocs theme. (#365)

- Bugfix: Include jQuery locally to avoid problems using MkDocs offline. (#143)

- Bugfix: Don't allow the docs_dir to be in the site_dir or vice versa. (#384)

- Bugfix: Remove inline CSS in the ReadTheDocs theme. (#393)

- Bugfix: Fix problems with the child titles due to the order the pages config was processed. (#395)

- Bugfix: Don't error during live reload when the theme doesn't exist. (#373)

- Bugfix: Fix problems with the Meta extension when it may not exist. (#398)

- Bugfix: Wrap long inline code otherwise they will run off the screen. (#313)

- Bugfix: Remove HTML parsing regular expressions and parse with HTMLParser to fix problems with titles containing code. (#367)

- Bugfix: Fix an issue with the scroll to anchor causing the title to be hidden under the navigation. (#7)

- Bugfix: Add nicer CSS classes to the HTML tables in bootswatch themes. (#295)

- Bugfix: Fix an error when passing in a specific config file with `mkdocs serve` . (#341)

- Bugfix: Don't overwrite index.md files with the `mkdocs new` command. (#412)

- Bugfix: Remove bold and italic from code in the ReadTheDocs theme. (#411)

- Bugfix: Display images inline in the MkDocs theme. (#415)

- Bugfix: Fix problems with no-highlight in the ReadTheDocs theme. (#319)

- Bugfix: Don't delete hidden files when using `mkdocs build --clean` . (#346)

- Bugfix: Don't block newer versions of Python-markdown on Python >= 2.7. (#376)

- Bugfix: Fix encoding issues when opening files across platforms. (#428)

4.1.37. Version 0.11.1 (2014-11-20)

- Bugfix: Fix a CSS wrapping issue with code highlighting in the ReadTheDocs theme. (#233)

4.1.38. Version 0.11.0 (2014-11-18)

- Render 404.html files if they exist for the current theme. (#194)

- Bugfix: Fix long nav bars, table rendering and code highlighting in MkDocs and ReadTheDocs themes. (#225)

- Bugfix: Fix an issue with the google_analytics code. (#219)

- Bugfix: Remove `__pycache__` from the package tar. (#196)

- Bugfix: Fix markdown links that go to an anchor on the current page. (#197)

- Bugfix: Don't add `prettyprint well` CSS classes to all HTML, only add it in the MkDocs theme. (#183)

- Bugfix: Display section titles in the ReadTheDocs theme. (#175)

- Bugfix: Use the polling observer in watchdog so rebuilding works on filesystems without inotify. (#184)

- Bugfix: Improve error output for common configuration related errors. (#176)

4.1.39. Version 0.10.0 (2014-10-29)

- Added support for Python 3.3 and 3.4. (#103)

- Configurable Python-Markdown extensions with the config setting `markdown_extensions`. (#74)

- Added `mkdocs json` command to output your rendered documentation as json files. (#128)

- Added `--clean` switch to `build`, `json` and `gh-deploy` commands to remove stale files from the output directory. (#157)

- Support multiple theme directories to allow replacement of individual templates rather than copying the full theme. (#129)

- Bugfix: Fix `<ul>` rendering in readthedocs theme. (#171)

- Bugfix: Improve the readthedocs theme on smaller displays. (#168)

- Bugfix: Relaxed required python package versions to avoid clashes. (#104)

- Bugfix: Fix issue rendering the table of contents with some configs. (#146)

- Bugfix: Fix path for embedded images in sub pages. (#138)

- Bugfix: Fix `use_directory_urls` config behavior. (#63)

- Bugfix: Support `extra_javascript` and `extra_css` in all themes. (#90)

- Bugfix: Fix path-handling under Windows. (#121)

- Bugfix: Fix the menu generation in the readthedocs theme. (#110)

- Bugfix: Fix the mkdocs command creation under Windows. (#122)

- Bugfix: Correctly handle external `extra_javascript` and `extra_css`. (#92)

- Bugfix: Fixed favicon support. (#87)

Did I try enough harder? ;-)