

---

# **Cyrax**

***Release 2.7***

October 11, 2016



<b>1</b>	<b>What's Cyrax?</b>	<b>1</b>
<b>2</b>	<b>Installation</b>	<b>3</b>
<b>3</b>	<b>Usage</b>	<b>5</b>
3.1	File formats . . . . .	5
3.2	Cyrax usage . . . . .	7



---

### What's Cyrax?

---

Cyrax is a static site generator using [Jinja2](#) template engine.

It's inspired from [Jekyll](#) and [Hyde](#) site generators and started when I realized that I'm dissatisfied with both of them by different reasons. When I tried to come up with name I remembered my favourite character from Mortal Kombat 3 so here we go.



---

## Installation

---

Well, it's quite simple, as usually:

```
pip install cyrax
```

If you've cloned [repository](#), then maybe you want to know that Cyrax depends on Jinja2 template library.





---

## Usage

---

Run `./cyrax`. Look at `content subdirectory` for example site.

Contents:

### 3.1 File formats

#### 3.1.1 Configuration format

Basically configuration format is a list of key-value pairs, where each pair should be located on a separate line, like this:

```
title: Some title
author: Some author
```

It can contain different data formats, namely:

- list

```
key: [list, of, values]
```

- dictionary

```
key: {key: value, key: value}
```

- date

```
key: date: yyyy-mm-dd HH:MM:SS
```

This format is currently the only one which is supported.

- boolean value

```
key: true
key: false
```

```
Case insensitive, with possible values for ``True``: `true`, `yes`,
`on`. For ``False``: `false`, `no`, `off`.
```

Everything else is a string, with an exception for fields with name `date` - value of those are checked to be a date value. In other case, string is returned.

### 3.1.2 Template formats

Jinja2 is used as a template engine, so basic template (leading to identification of page as `Page` object) is a Jinja2 template, which extends parent template.

Every Jinja2 template in Cyrax can use `{% meta %} ... {% endmeta %}` tag, which should contain text in *Configuration format*, setting up and overriding options for current page.

#### Parent template

Each entry has a `parent_html`, from which it will be inherited. It is determined by model name, as `_lowered model name.html`. In case it does not exist, `_base.html` is used (can be overridden, see `parent_tmpl` in *Configuration*).

### 3.1.3 Models

There are few models for different templates coming with Cyrax predefined, and you can define your own using a `sitecallback` *Configuration* variable by adding new models to `cyraxlib.models.TYPE_LIST`.

Models are just descendants of an `object`, defining `py:classmethod:check(entry)` which should determine if passed entry is an instance of a given model. Another requirement is `py:method:get_relative_url()`, which should return URL where rendered entry will be located, relative to site root.

Common configuration variables:

- `isdir` (default: `True`) - should current entry be rendered as a directory with `index.html` inside or a simple file.
- `type` (default: `None`) - predefine type of current entry, without running `check()` from every model.

#### NonHTML

Model for entries whose extension doesn't end with `.html`, usually last of models in `TYPE_LIST`. `isdir` returns `False` by default.

#### Page

Page is a model for single html pages and its `check()` returns `True` if path ends with `.html`. Does nothing special.

#### Post

Model for a blog post, determined by checking if its name is prepended with a date (either separated by dashes or by slashes, f.e. `'blog/2010/03-14-some-post.html'`).

Special behavior contains adding itself to a sorted collection of posts on a `Site` object (`site.posts`) and by adding tag from `tags` to a sorted collection of tags on `Site` object (`site.tags`).

Configuration variables:

- `tags` (default: `[]`) - list of tag names.

## Tag

Entry for a tag. Most of the special behavior is done by adding an event handler to `site-travered` signal, which adds virtual pages for tags.

## 3.2 Cyrax usage

Main executable is `cyrax`. Run it in a directory with site sources to compile your site. It expects file `settings.cfg` in a directory it's started in.

### 3.2.1 Configuration

`settings.cfg` is a main site configuration file and has to be written in a simple *Configuration format*. Possible configuration options used by Cyrax core:

- `parent_tmpl` - name of parent template for a `Page` object. Defaults to `_base.html`.
- `exclude` - list of filenames (or glob patterns) to exclude from resulting site.
- `sitecallback` - Python path to function, which takes a `Site` object as an argument before traversing tree of files and can perform some modifications to behavior of Cyrax. [Example](#).

Any other options will be set on your `Site` object, which is available in any template under name of `site`.