Hyde Documentation

Release 0.8.9

Hyde Developers

March 10, 2016

Contents

CHAPTER 1

Overview

Hyde is a static website generator written in Python. While Hyde took life as Jekyll‘s evil twin, it has since been completely consumed by the dark side and has taken on an identity of its own.

Hyde desires to fulfill the lofty goal of removing the pain points involved in creating and maintaining static websites.

CHAPTER 2

Spotlight

- Support for powerful template languages like Jinja2 complemented with custom tags and filters.
- Rich *object model* and overridable hierarchical metadata available for use in templates.
- Configurable sorting, tagging, and grouping support.
- Extensible Plugins architecture with text preprocessing and HTML postprocessing support for complex content transformations.
- Instant preview using built-in webserver that regenerates content if needed.

CHAPTER 3

Installation

Hyde is available on PyPI.

Installing hyde is as simple as running the following command:

```
$ python -m pip install hyde
```

CHAPTER 4

Quickstart

After installing `hyde`, creating and generating a website is extremely simple.

To create a new Hyde website:

```
$ hyde -s /path/to/your/site create
```

To generate the HTML:

```
$ cd /path/to/your/site
$ hyde gen
```

To serve the generated content, use Hyde's built-in web server:

```
$ hyde serve
```

The website is now accessible at http://localhost:8080.

The webserver regenerates the necessary files to serve your request. So, you can make your changes and simply refresh your browser to view them.

For all the supported options, read the Command Line Reference documentation or run:

```
$ hyde --help
```

CHAPTER 5

Your First Hyde Website

Hyde uses the `basic` layout to generate your website by default. When you view your generated website, you will see the following dummy pages:

#FIXME: images

You can now continue to edit the content, layout and styles to customize it to your needs. Please see the Templates documentation for more information.

CHAPTER 6

Source

Hyde is socially coded. Feel free to fork.

Contents:

6.1 Installation

Installing hyde is as simple as running the following command:

```
python -m pip install hyde
```

However, based on your choice and use of plugins you may need to install additional packages. The requirements for each plugin is outlined in the corresponding Plugins documentation.

6.1.1 Dependencies

While your mileage may vary, we consider the following to be essential for generating a static website with Hyde. These are a part of the requirements file, and the above command will download and install all of them as part of Hyde.

It is also recommended that you use virtualenv to separate the Hyde environment from other python projects. Note that installing Hyde using `pip` would install all of the below. However, if you'd like finer grained control over the packages, you can install these individually:

1. `argparse`: argparse is required if you are on python 2.6.
2. `commando`: commando is a wrapper on top of argparse to give better syntax and support for multi-command applications.
3. `Jinja2`: While Hyde will support many more template languages in the future, currently only Jinja2 is wholly supported and recommended.
4. `Markdown`: While there are plans to add support for other markups (textile, restructured text, asciidoc etc..,), markdown is the one thats currently completely supported.
5. `Pyyaml`: Much of Hyde's Configuration is done using YAML.
6. `pygments`: For syntax highlighting.
7. `Typogrify`: Typogrify automatically fixes and enhances the typographical accuracy of your content. While this is not a technical requirement for Hyde, it is absolutely essential to create good looking content.

The following commands can be used to install the dependencies for Hyde individually:

```
python -m pip install argparse
python -m pip install commando
python -m pip install jinja2
python -m pip install markdown
python -m pip install pyyaml
python -m pip install pygments
python -m pip install smartypants
python -m pip install typogrify
```

6.2 Command Line Reference

The `hyde` command line supports the following subcommands:

`create`	Initialize a new site at a given path.
`gen`	Generate the website to a configured deploy folder.
`serve`	Start a local HTTP server that regenerates based on the requested file.
`publish`	Publish the generated site based on configuration.

6.2.1 create

Create a new Hyde website:

```
hyde create

hyde [-s </site/path>] [-v] create [-l <layout>] [-f] [-h]
```

Options:

`-s SITEPATH`, `--sitepath SITEPATH`

Specify where the site should be created.

A new site will only be created if `SITEPATH` does not exist. To overwrite an existing directory, use the `--force` option.

Defaults to the current directory.

`-f`, `--force`

Overwrite files and folders at the given site path.

`hyde create` will raise an exception if the target directory is not empty unless this option is specified.

Defaults to false.

`-l LAYOUT`, `--layout LAYOUT`

The name of the layout to use for creating the initial site. Hyde provides three layouts: `basic`, `test`, and `doc`.

Hyde tries to locate the specified layout in the following folders:

1. In the `layouts` folder under the path specified by the `HYDE_DATA` environment variable.
2. In `hyde`'s own `layouts` folder.

Assuming the `HYDE_DATA` environment variable is empty and the folder `~/test` is empty, the following command will create a new Hyde site at `~/test` with the contents of the `layouts/doc` folder:

```
hyde -s ~/test create -l doc
```

Defaults to `basic`.

`-v`, `--verbose`

Log detailed messages to the console.

Defaults to false. Show only essential messages if this option is omitted.

`-h`, `--help`

Display the help text for the `create` command.

6.2.2 generate

Generate the given website:

```
hyde gen

hyde [-s <site/path>] [-v] gen [-r] [-d <deploy/path>] [-c <config/path>] [-h]
```

Options:

`-s SITEPATH`, `--sitepath SITEPATH`

The path to the site to be generated.th to the site to be generated.

Defaults to the current directory.

`-r`, `regen`

Regenerate the entire website. By default, `hyde gen` performs incremental generation. While Hyde does a good job at understanding dependencies, its far from perfect. When there are cases where the incremental generation does not yield the desired results, you can provide this option to generate the website from scratch.

Defaults to incremental generation.

`-d DEPLOY_PATH`, `--deploy-path DEPLOY_PATH`

Location where the site should be generated. This option overrides any setting specified in the Hyde configuration. The path is assumed to be relative to the site path unless a preceding path separator is found.

Defaults to the option specified in the config file or the `deploy` folder under the current site path if no config entry exists.

`-c CONFIG`, `--config-path CONFIG`

Specify an alternate configuration file to use for generating the site. This is useful if you have two different configurations for you production versus development websites. The path is assumed to be relative to the site path unless a preceding path separator is found.

The following command will use `production.yaml` as the configuration file and generate the website at `~/test` to the `~/production_site` directory:

```
cd ~/test
hyde gen -c production.yaml -d ~/production_site
```

Defaults to `site.yaml`.

`-v`, `--verbose`

Log detailed messages to the console.

Defaults to false. Show only essential messages if this option is omitted.

`-h`, `--help`

Display the help text for the `gen` command.

6.2.3 serve

Start the built in web server that also regenerates based on the request if there are changes:

```
hyde serve

hyde [-s </site/path>] [-v] gen [-d </deploy/path>] [-c <config/path>] [-h]
```

Options:

`-s SITEPATH`, `--sitepath SITEPATH` `-d DEPLOY_PATH`, `--deploy-path DEPLOY_PATH` `-c CONFIG`, `--config-path CONFIG`

Since the `serve` command auto generates if there is a need, it needs the same parameters as the `gen` command. The above parameters serve the same purpose here as in the `gen` command.

`-a ADDRESS`, `--address ADDRESS`

The address to serve the website.

Defaults to `localhost`.

`-p PORT`, `--port PORT`

The port to serve the website.

The following command will serve the website at http://localhost:8181:

```
hyde serve -p 8181
```

Defaults to 8080.

`-h`, `--help`

Display the help text for the `serve` command.

6.2.4 publish

Publish the site based on configuration. Currently, the only supported publishing target is a git repository. See the publisher documentation for more information.

Options:

`-s SITEPATH, --sitepath SITEPATH`

The path to the site to be generated.

Defaults to the current working directory.

`-p CONFIG`

The key for configuration section in the site configuration that has the settings for the publisher. For example, the following configuration, when invoked with `hyde publish -p github` will use the `Git` publisher to publish the generated site to `hyde/hyde.github.com` repository:

```
publisher:
    github:
        type: hyde.ext.publishers.dvcs.Git
        path: ../hyde.github.com
        url: git@github.com:hyde/hyde.github.com.git
```

6.3 Configuration

Hyde is configured using one or more `yaml` files. On top of all the niceties `yaml` provides out of the box, Hyde also adds a few power features to manage complex websites.

If a site has a `site.yaml` file in the root directory, it is used as the configuration for generating the website. This can be overridden by providing a command line option. See the command line reference for details.

6.3.1 Inheritance

Configuration files can inherit from another file using the `extends` option. For example, the following configuration will inherit all properties from `site.yaml` and override the `mode` property:

```
extends: site.yaml
mode: production
```

This is useful for customizing the site to operate in different modes. For example, when running locally you may want your media files to come from `/media` but on production you may have a subdomain and want the media to come from `http://abc.example.com/media`.

This can be accomplished by creating an `extended` configuration file and overriding the `media_url` property.

Todo

Add link to real example config.

The following settings can be defined for use in templates:

6.3.2 Paths & URLs

`media_root`	The root path where media files (images, CSS, JavaScript, etc.) can be found. This may be used by plugins for special processing. If your website does not have a folder that contains all media, you can safely omit this property. Defaults to `media`.
`media_url`	The url prefix for serving media files. If you are using a CDN like Amazon S3 or the Rackspace cloud and host all of your media files from there, you can make use of this property to specify the prefix for all media files. Defaults to `/media`.
`base_url`	The base url from which the site is served. If you are hosting the website in a subdomain or as part of a larger website, you can specify this property to indicate the path of this project in your website. Defaults to `/`.

6.3.3 Plugins and Templates

`template`	The template engine to use for processing the website can be specified in the configuration file as a Python class path. Currently, only `Jinja2` is supported. Reserved for future use. Defaults to `hyde.ext.templates.jinja.jinja2template`.
`plugins`	Plugins are specified as list of Python class paths. Events that are raised during the generation of the website are issued to the plugins in the same order as they are listed here. You can learn more about how the individual plugins are configured in the plugin documentation. Optional. By default, no plugins are enabled.

6.3.4 Context Data

The context section contains key / value pairs that are simply passed on to the templates.

For example, given the following configuration, the statement `{{ app.current_version }}` in any template will output `0.6`:

```
context:
    data:
        app:
            version: 0.6
```

By default, no context variables are provided.

6.3.5 Context Data Providers

Providers are special constructs used to import data into the context. For example, data from a database can be exported as `yaml` and imported as a provider. For example, the following snippets would import the data in `app-versions.yaml` into `context[versions]`. This data can then be used directly in templates in this manner: `{{ versions.latest }}`.

```
# site.yaml
context:
    providers:
        versions: app-versions.yaml
```

```
# app-versions.yaml
latest: 0.6
```

6.3.6 Markdown

Extensions and extension configuration for markdown can be configured in the `markdown` property. You can read about markdown extensions in the Python-Markdown documentation.

The following configuration will use the `def_list`, `tables`, `headerid` extensions in Python-Markdown.

6.4 Plugins

Hyde is built with a nuclear engine that is stripped down but powerful. Even core features like `metadata` are added as plugins to keep the engine bloat free.

Hyde's plugin architecture is event driven; plugins get notified during the course of the generation to allow them to alter/guide the generation process.

If you are interested in creating plugins for Hyde, you can read the developer documentation.

6.4.1 Configuration

Hyde's plugins get loaded if they are listed in the plugins section of Configuration. Plugins also accept additional parameters in their respective sections. For example, `MyAwesomePlugin` will get parameters from `myawesome` section in the configuration file.

6.4.2 In the box

Hyde is shipped with the following plugins:

6.4.3 Metadata

Metadata

class `hyde.ext.plugins.meta.`**`MetaPlugin`**(*site*)

Metadata plugin for hyde. Loads meta data in the following order:

1.meta.yaml: files in any folder

2.frontmatter: any text file with content enclosed within three dashes or three equals signs.

Example:

```
---
abc: def
---
```

Supports YAML syntax.

AutoExtend

class `hyde.ext.plugins.meta.`**`AutoExtendPlugin`**(*site*)

The plugin class for extending templates using metadata.

Sorter

class `hyde.ext.plugins.meta.`**`SorterPlugin`**`(`*site*`)`

Sorter plugin for hyde. Adds the ability to do sophisticated sorting by expanding the site objects to support prebuilt sorting methods. These methods can be used in the templates directly.

Configuration example:

```
#yaml

sorter:
    kind:
        # Sorts by this attribute name
        # Uses `attrgetter` on the resource object
        attr: source_file.kind

        # The filters to be used before sorting
        # This can be used to remove all the items
        # that do not apply. For example,
        # filtering non html content
        filters:
            source_file.kind: html
            is_processable: True
            meta.is_listable: True
```

6.4.4 CSS

Less CSS

class `hyde.ext.plugins.css.`**`LessCSSPlugin`**`(`*site*`)`

The plugin class for less css

6.4.5 Text Replacement

Blockdown

class `hyde.ext.plugins.text.`**`BlockdownPlugin`**`(`*site*`)`

The plugin class for block text replacement.

Mark

class `hyde.ext.plugins.text.`**`MarkingsPlugin`**`(`*site*`)`

The plugin class for mark text replacement.

Refer

class `hyde.ext.plugins.text.`**`ReferencePlugin`**`(`*site*`)`

The plugin class for reference text replacement.

Textlinks

class `hyde.ext.plugins.text.`**`TextlinksPlugin`**(*site*)

The plugin class for text link replacement.

Syntext

class `hyde.ext.plugins.text.`**`SyntextPlugin`**(*site*)

The plugin class for syntax text replacement.

6.4.6 Structure

Folder Flattener

class `hyde.ext.plugins.structure.`**`FlattenerPlugin`**(*site*)

The plugin class for flattening nested folders.

6.5 Web Server

The Hyde web server is a simple, bare-bones webserver based on Python's Simple HTTP Request Handler.

The following command starts the built in webserver:

```
hyde serve
```

You can now access your website at `http://localhost:8080`.

The webserver regenerates the necessary files to serve your request. So, you can make your changes and simply refresh your browser to view them.

6.5.1 Special Parameters

The Hyde webserver supports just one special parameter: `refresh`.

If you add `?refresh` to the URL, the server regenerates the site completely and refreshes your browser. Note that for larger sites, this may take a few seconds to complete.

6.5.2 Dependencies

Information about dependencies between pages are stored in your site root directory in the `.hyde_deps` file. If regeneration is not consistent with your expectations, you can simply delete this file, and Hyde will build the dependency tree again.

6.6 Templates

6.6.1 Creating Layouts

Hyde is designed to support more than one template language for creating layouts, however, it currently only supports Jinja2. This section of the documentation will focus on creating templates using Jinja2. If you are not familiar with

Jinja2, it's a good idea to go through the most excellent Jinja2 documentation.

6.6.2 Site Structure

Hyde encourages separation of content from layout. The following shows a typical structure of a Hyde website:

```
-- content/
|   -- about.html
|   -- blog/
|   -- index.html
|   -- layout/
|   |   -- base.j2
|   |   -- macros.j2
|   -- media/
|   |   -- css/
|   |   -- images/
|   |   -- js/
|   -- portfolio/
|   -- projects/
-- site.yml
```

good objective is to have all the files in content contain as little layout as possible and be written with a text oriented markup language like markdown. While its not always possible to achieve 100% separation, hyde provides several nice tools to get very close to that goal.

6.6.3 Context Variables

Hyde by default makes the following variables available for templates:

- `site`: Represents the container object of the entire site.
- `node`: The node (folder) where the current file resides.
- `resource`: The resource (file) that is currently being processed.
- Context variables: all variables defined under the `context` section of the site configuration are available to the templates.

Read more information about the `site`, `mode`, and `resource` variables in the *site model documentation*.

Read more information about context variables in the *configuration documentation*.

6.7 Changelog

6.7.1 Version TBA

- Experimental Python 3 support

6.7.2 Version 0.8.9 (2015-11-09)

- Add mailing list and IRC channel information to README.
- Relax the version numbers (with an upper bound on major version) for dependencies.

Thanks to @jonafato:

- Update command and fswrap dependencies.
- Use setuptools directly.
- Improve travis builds. - add caching support. - prevent duplicate builds. - add support for new container based builds.
- Move tests out of the hyde package.

Thanks to @bheesham:

- Bugfix: Fix favicon and apple touch icon paths in basic template.

Thanks to @jd:

- Bugfix: Paginator fails when there is no meta variable.
- Add tox support.
- Fix setup.py and make it pep8 compliant.
- Add pages list to resources in paginator plugin.

Thanks to @mtahmed:

- Bugfix: Ensure that *self.q* is assigned to None when PyQuery is unavilable.

Thanks to @ftao:

- Bugfix: Remove typo in sphinx plugin commando import.

Thanks to Alexey Biryukov:

- Hyde now supports assigning jinja2 custom tests similar to custom filters in site.yaml.

Thanks to @llonchj:

- Huge updates to unit tests, development dependencies (flake8, pep8 etc).
- Travis support.
- The `h` binary is now deprecated and will be removed in one of the future releases.
- libsass support.

6.7.3 Version 0.8.8 (2013-08-20)

- Bugfix: Fix smartypants to 1.7.x to avoid the breaking changes in 1.8. (Issue #229)
- Bugfix: Upgrade Jinja to 2.7.1 - this fixes mitsuhiko/jinja2#243.

Thanks to @shym: * Fix various typos in documentation and code. (Issue #227)

Thanks to @maethor: * Make username optional in SSH publisher. (Issue #222)

Thanks to @merlinrebrovic: * Refresh the starter layout: (Issue #220)

- Conform to changes from the newest Hyde version.
- Clean up Markdown and add explanations.
- Increase font sizes.
- Remove HTML5 shiv and hgroup element (deprecated).
- Other minor improvements.

Thanks to @webmaster128: * Bugfix: Add spaces to menu in `starter` layout. (Issue #219)

6.7.4 Version 0.8.7 (2013-05-30)

- Bugfix: Ensure dependencies are handled properly when using the combine plugin. (Issue #120).
- Bugfix: Ensure external urls are encoded properly. (Issue #158).
  - **BREAKING**: While this change will correct the weird encoding behavior, this also changes the way `xxx_url` macros work.
  - Do not encode the url bases (`base_url`, `media_url`). Only the path component needs to be encoded.
- Bugfix: Fix subfolders for root paths on windows. (Issue #167).
  - Ensure that subfolders for `layout_root`, `content_root`, `media_root` and `deploy_root` works reliably on windows. Use `fully_expanded_path` for all path components.
- Bugfix: Context providers now accept all valid yaml data (Issue #189).
- Bugfix: Fixed `raise_exceptions` command line parameter handling.
- Better exception handling. (Issue #204)
  - Hyde now propagates exceptions
  - Hyde does not raise exceptions by default. `-x` flag is required for raising exceptions on error.
  - Traceback is available in the `verbose` mode
- Upgraded to commando 0.3.4
- Upgraded to Jinja 2.7
- Make sorter's prev/next links circular. (Issue #208)
- Bugfix: Include project artifacts in sdist. (Issue #211)
  - Add LICENSE, AUTHORS and CHANGELOG to MANIFEST.in
- Add "Hyde contributors" to LICENSE copyright.
- Upgrade `UglifyPlugin` to use 2.3.5 version of uglify. (Issue #214)
- Add support for draft blog posts. (Issue #213)
- Bugfix: Use `clearfix` class in `listing.j2`. (Issue #156)
- Bugfix: Use `relative_path` instead of url in `macros.j2`. (Issue #180)

6.7.5 Version 0.8.6 (2013-04-30)

Thanks to @QuLogic:

- Refactor: Plugins reorganized by function. (Issue #170)
- Add HG Dates Plugin. (Issue #177)
- Add Clever CSS Plugin. (Issue #178)
- Add Sassy CSS Plugin. (Issue #179)

Thanks to @sirlantis:

- Add support for custom jinja filters. (Issue #159)

Thanks to @gfuchedzhy

- Bugfix: Serve files without a resource. (Issue #92)

Thanks to @ilkerde:

- Add Require JS plugin. (Issue #201)

Thanks to @jakevdp:

- Add SSH publisher. (Issue #205)

Thanks to @herr-lehmann and @nud:

- Bugfix: Fix date time comparison in git plugin. (Issue#142, #199, #137)

Thanks to @rephorm, @gfuchedzhy and @vincentbernat:

- Add thumbnail plugin. (Issue #169, #89)

Thanks to @vincentbernat:

- Add Coffeescript plugin. (Issue #172)
- Add jpegtran plugin. (Issue #171)

Thanks to @jabapyth:

- Add extension support for restructured text. (Issue #206)

Thanks to @tarajane:

- Bugfix: Update the .clear styleName to be .clearfix instead. Base.j2 applies the 'clearfix' class to the 'banner' element, and not the 'clear' class. (Issue #156)

Thanks to @pib:

- Bugfix: Use `_transform` instead of `transform` in Expando. (Issue #152, #153)

6.7.6 Version 0.8.5 (2013-04-17)

- Upgrade dependencies and setup for 0.8.5
- Remove `hyde.fs` use `fswrap` package instead.
- Remove logging functions from `hyde.util`. Use `commando.util` instead.
- Remove `hyde.loader`. Use `commando.util.load_python_object` instead.
- Bugfix: Use the released version of typogrify. (Issue #193)
- Bugfix: Fixed stylus `indent` issues with empty files. (Issue #161)
- Bugfix: Added support for plugin paths relative to site. (Issue #107)
- Bugfix: Folder Flattener updates node's `relative_deploy_path` & `url` attributes as well. (Issue #126)
- BREAKING: As part of the above fix, `resource.url` is prefixed with a `/`. (Issue #126)
- Added `simple_copy` feature to account for unprocessable files that are nonetheless required to be deployed (Issue #121)
- Bugfix: Relative path was used in the server as the sitepath (Issue #119)
- Plugins now support inclusion filters. (Issue #112)
  - `include_file_patterns` property accepts globs to filter by file name.
  - `include_paths` accepts paths relative to content.
  - `begin_node` and `node_complete` honor `include_paths`

  - begin_text_resource, text_resource_complete, begin_binary_resource and binary_resource_complete honor both.

- Bugfix: Unsorted combine files fixed. (Issue #111)
- Added an optional sorting parameter. (Issue #111)
- Bugfix: Modified combine plugin to process during begin_text_resource. (Issue #110)
- Modified combine plugin to support relative paths and recursion. (Issue #108)
- Added ability to specify safe characters in content_url, media_url functions and urlencode filter. (Issue #103)

Thanks to @idank

- Bugfix: Use check_output to avoid a traceback when subprocess command fails.
- Bugfix: Tag archive generator uses subscript syntax to avoid failure when tags contain '-' or space. (Issue #130)

Thanks to @jd

- Bugfix: Metadata Plugin: Do not try to read meta data on simple_copy files. (Issue #124, Issue #121)
- Bugfix: Force escape on title in Atom feed. (Issue #176)
- Add node.rwalk method for traversing the node in reverse. (Issue #176)

Thanks to @vinilios

- Added a helper method in Expando class to ease up non existing keys handling. (Issue #117)
- Some improvements in LessCSSPlugin to be able to build complex less projects (such as twitter bootstrap) (Issue #117)

Thanks to @Erkan-Yilmaz

- Fixed typos in README.

Thanks to @merlinrebrovic

- Updates and improvements to the starter template.
  - Cleans up CSS.
  - Handles page title endings more elegantly.
  - Renders the advanced menu below the basic one.
  - Corrects and updates content.
  - Explains how to generate and serve the template.
  - Makes it more straightforward to contribute.

Thanks to @joshgerdes:

- Made urlencoding safe character list configurable. (Issue #150)

Thanks to @irrelative:

- Bugfix: Avoid index error if there aren't pages when iterating for paginator. (Issue #190)

Thanks to @davefowler:

- Bugfix: Infinate recursion error with resource dependencies. (Issue #155, Issue#200)

Thanks to @adube:

- Bugfix: Fix atom.j2 to use `relative_path` instead of `url` when referencing templates. (Issue #155, Issue#203)

6.7.7 Version 0.8.4 (2011-11-09)

- Bugfix: Configuration now gets reloaded when server regenerates (Issue #70)
- Bugfix: Added styles for codebox (Issue #69)
- Tagger now generates archives upfront in begin_site (Issue #72)
- **Breaking**: The default nodemeta file has been changed to meta.yaml
- Added test for codehilite markdown extension (Issue #82)
- Added rst_directive.py from the pygments repository (Issue #82)
- Added support for ignoring nodes (Issue #80)
- Hyde now ignores .hg, .svn and .git by default (Issue #80)
- Added support for default publisher (Issue #83)
- Added `urlencode` and `urldecode` filters. (Issue #102)
- Bugfix: Fixed tests for Issue #88
- Added tests for sorting groups
- Added support for loading modules from the site path. Thanks to @theomega for the idea (Issue #78 & #79)
- Added docutils to dev-req.txt
- Bugfix: Fixed uglify-js tests

Thanks to @nud

- `$PATH` based executable discovery for `CLTransformer` plugins. (Issue #100)
- Bugfix: Fix class name of `test_stylus` (Issue #97)

Thanks to @gfuchedzhy

- Bugfix: Textlinks plugin: do nothing if resource doesn't use template. (Issue #96)
- Bugfix: Retain permissions in text files during generation (Issue #90)
- Bugfix: Added support for encoded urls to hyde server. (Issue #88)
- Bugfix: Converted `content_url` and `media_url` to encoded urls. (Issue #88)
- Bugfix: All occurrences of `str` replaced with `unicode`. (Issue #87)
- Bugfix: CLTransformer now gracefully handles arguments that have "=". (Issue #58)

Thanks to @vincentbernat

- Support for `output_format` configuration in markdown (Issue #89)

Thanks to @merlinrebrovic

- Hyde starter kit extended with advanced options (Issue #68)

Thanks to @tcheneau

- Added support for AsciiDoc. (Issue #76)

Thanks to @gr3dman

- Added paginator plugin and tests (Issue #73)

Thanks to @benallard

- Added restructuredText plugin (Issue #63)
- Added restructuredText filter (Issue #63)
- Added traceback support for errors when server is running (Issue #63)

Thanks to @rfk

- Added Sphinx Plugin (Issue #62)
- Bugfix: PyFS publisher now checks if the pyfs module is installed. (Issue #62)

6.7.8 Version 0.8.3 (2011-06-20)

- Bugfix: A bad bug in Expando that modified the `__dict__` has been fixed. (Issue #53)
- Tags now support metadata. Metadata can be provided as part of the tagger plugin configuration in `site.yaml`
- Ensured that the context data & providers behave in the same manner. Both get loaded as expandos. (Issue #29)
- `hyde serve` now picks up changes in config data automatically. (Issue #24)
- Bugfix: `hyde create` only fails when `content`, `layout` or `site.yaml` is present in the target directory. (Issue #21)
- Bugfix: Exceptions are now handled with `ArgumentParser.error`.
- Bugfix: Sorter excludes items that do not have sorting attributes. (Issue #18)
- Wrapped `<figure>` inside `<div>` to appease markdown. (Issue #17)
- Added `display:block` for html5 elements in basic template so that it works in not so modern browsers as well. (Issue #17)

Thanks to Joe Steeve.

- Changed deploy location for main.py and fixed entry point in `setup.py`. (Issue #56)

Thanks to @stiell

- Bugfix: Better mime type support in hyde server (Issue #50)
- Bugfix: Support empty extension in tagger archives (Issue #50)

Thanks to @gfuchedzhy

- Bugfix: Hyde server now takes the url cleaner plugin into account. (Issue #54)

Thanks to @vincentbernat

- Bugfix: Ensure image sizer plugin handles external urls properly. (Issue #52)

Thanks to @rfk

- Added PyPI publisher (Issue #49)
- Bugfix: Made `site.full_url` ignore fully qualified paths (Issue #49)

Thanks to @vincentbernat

- Added JPEG Optim plugin (Issue #47)
- Fixes to CLTransformer (Issue #47)

6.7.9 Version 0.8.2 (2011-05-10)

Thanks to @merlinrebrovic

- Added hyde starter kit (Issue #43)

Thanks to @vincentbernat

- Added git dates plugin (Issue #42)
- Added Image size plugin (Issue #44)
- Added silent, compress and optimization parameter support for less css plugin (Issue #40)
- Fixed plugin chaining issues (Issue #38)
- Added Language(translation) plugin (Issue #37)
- Bugfix: Made sorting tests more predictable (Issue #41)
- Bugfix: Added more standard paths for executables (Issue #41)
- Added Combine files plugin (Issue #39)
- Added ignore option in site configuration to igore based on wildcards (Issue #32)

Thanks to @pestaa

- Added support `UTF8` keys in `metadata` and `config` (Issue #33)

6.7.10 Version 0.8.1 (2011-05-09)

Thanks to @rfk.

- Updated to use nose 1.0 (Issue #28)
- Bugfix: LessCSSPlugin: return original text if not a .less file (Issue #28)
- PyFS publisher with mtime and etags support. (Issue #28)

6.7.11 Version 0.8 (2011-04-13)

- Relative path bugs in windows generation have been fixed.

6.7.12 Version 0.8rc3 (2011-04-12)

- Fixed a jinja2 loader path issue that prevented site generation in windows
- Fixed tests for stylus plugin to account for more accurate color manipulation in the latest stylus

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Index