**Cactus 3 is out!**

We're happy to announce Cactus 3. It brings a set of great new features like asset fingerprinting, an asset pipeline, pretty urls, native Mac filesystem events, automatic nameserver configuration, support for multiple deployment backends (Google Sites) and more. Large parts of the code have been rewritten, accompanied by an extensive suite of unit tests. Many thanks to Thomas Orozco and other contributors.

**What is Cactus**

Cactus is a simple but powerful [static website generator](http://mickgardner.com/2011/04/27/An-Introduction-To-Static-Site-Generators.html) using Python and the [Django template system](http://docs.djangoproject.com/en/dev/topics/templates/). Cactus also makes it easy to develop locally and deploy your site to S3 directly. It works great for company, portfolio, personal, support websites and blogs.

To get a quick overview [watch this short video tutorial](https://vimeo.com/46999791).

Cactus is based on the idea that most dynamic features on websites these days can be done using Javascript while the actual site can stay static. Static websites are easy to host and typically very fast.

I developed Cactus because I wanted a standard, easy system that designers at [Sofa](http://madebysofa.com/) could use to build and deploy fast websites. So typical users would be designers that are tech-savvy, want to use templates, but don't like to mess with setting up django or S3.

Since then it has evolved quite a bit with a plugin system that supports blogging, spriting, versioning and is extensible.

You can find more discussion about static site generators in this [Hacker News discussion](http://news.ycombinator.com/item?id=2233620).

**Examples**

* [http://www.cactusformac.com](http://www.cactusformac.com/) - Cactus app site
* [http://www.framerjs.com](http://www.framerjs.com/) - Framer website
* [https://crate.io](https://crate.io/) - Crate, Distributed Database for Docker (source: <https://github.com/crate/crate-web>)

There is also an example blog project included.

**Super quick tutorial for the impatient**

Install Cactus with the following one liner

sudo easy\_install cactus

If you saw no errors, you can now generate a new project

cactus create ~/www.mysite.com

To start editing and previewing your site type the following. Then point your browser to localhost:8000 and start editing. Cactus will automatically rebuild your project and refresh your browser on changes.

cd ~/www.mysite.com

cactus serve

Once you are ready to deploy your site to S3 you can run the following. You will need your [Amazon access keys](https://payments.amazon.com/sdui/sdui/helpTab/Checkout-by-Amazon/Advanced-Integration-Help/Using-Your-Access-Key). If you don't have one yet, [read how to get one here](http://www.hongkiat.com/blog/amazon-s3-the-beginners-guide/#Gettting_an_Amazon_S3_Account).

cactus deploy

Voila. Your website generated by Cactus and hosted on S3!

**Extended guide**

**Creating a new project**

You can create a new project by generating a new project structure like this. Make sure the destination folder does not exist yet.

cactus create [path]

If you did not see any errors, the path you pointed to should now look like this.

- .build Generated site (upload this to your host)

- pages Your actual site pages

- index.html

- sitemap.xml

- robots.txt

- error.html A default 404 page

- templates Holds your django templates

- base.html

- static Directory with static assets

- images

- css

- js

- plugins A list of plugins. To enable remove disabled from the name

**Making your site**

After generating your site you can start building by adding pages to contents, which can rely on templates. So for example if you want a page /articles/2010/my-article.html you would create the file with directories in your pages folder. Then you can edit the file and use django's template features.

**Building your site**

When you build your site it will generate a static version in the build folder that you can upload to any host. Basically it will render each page from your pages folder, copy it over to the build folder and add all the static assets to it so it becomes a self contained website. You can build your site like this:

cd [your-cactus-path]

cactus build

Your rendered website can now be found in the (hidden) [path]/.build folder. Cactus can also run a small webserver to preview your site and update it when you make any changes. This is really handy when developing to get live visual feedback.

You can run it like this:

cactus serve

**Linking and contexts**

Cactus makes it easy to relatively link to pages and static assets inside your project by using the template tags {% static %} and {% url %}. For example if you are at page /blog/2011/Jan/my-article.html and would like to link to /contact.html you would write the following:

<a href="{% url '/contact.html' %}">Contact</a>

Just use the URL you would normally use: don't forget the leading slash.

**Templates**

Cactus uses the Django templates. They should be very similar to other templating systems and have some nice capabilities like inheritance. In a nutshell: a variable looks like this {{ name }} and a tag like this {% block title %}Welcome{% endblock %}. You can read the [full documentation](https://docs.djangoproject.com/en/dev/topics/templates/) at the django site.

**Enabling Plugins**

To enable a plugin for your site, change the file name from [PLUGIN].disabled.py to [PLUGIN].py.

**Deploying**

Cactus can deploy your website directly to S3, all you need are your Amazon credentials and a bucket name. Cactus remembers these in a configuration file name config.json to make future deploys painless. The secret key is stored securely in the Keychain or similar services on other OSs.

cactus deploy

After deploying you can visit the website directly. Cactus also makes sure all your text files are compressed and adds caching headers.

**Extras**

**Blogs**

For the full example of how to build a blog on top of Cactus, see [CactusBlog](https://github.com/koenbok/CactusBlog/).

Blog plugin takes post title, author, and date from metadata. For example:

title: My first post

author: Koen Bok

date: 22-07-2012

{% extends "post.html" %}

{% block body %}

{% endblock %}

Modify config.json to set a custom blog path, default author name, or date pattern used to parse metadata. The defaults are:

"blog": {

"path": "blog",

"author": "Unknown",

"date-format": "%d-%m-%Y"

}

**YAML Variables**

By default you can declare variables to be included above each page, for example:

test\_text: Lorem Ipsum

<p>{{ test\_text }}</p>

You can declare the variables using YAML instead. Just surround the block with the --- and ... [Document Separators](http://yaml.org/spec/1.1/#id857577). Then the objects and arrays will be available inside the templates:

---

header\_text: Lorem Ipsum

custom\_object:

name: Lorem

description: Ipsum

custom\_array:

-

name: lorem

-

name: ipsum

...

{% for item in custom\_array %}

<p>{{ header\_text }}: {{ item.name }}</p>

{% endfor %}

<p>{{ custom\_object.name }} | {{ custom\_object.description }}</p>

The *PyYAML* library is used for this functionality.

**Asset pipeline**

Cactus comes with an asset pipeline for your static files. If you'd like to use it, make sure you use the {% static %} template tag to link to your static assets: they might be renamed in the process.

**Fingerprinting**

Modify config.json, and add the extensions you want to be fingerprinting:

"fingerprint": [

"js",

"css"

],

This lets you enable caching with long expiration dates. When a file changes, its name will reflect the change. Great for when you use a CDN.

**Optimization**

Modify config.json, and add the extensions you want to be optimizing:

"optimize": [

"js",

"css"

],

By default, Cactus will use:

* YUI for CSS minification
* Closure compiler for JS minification (YUI is built-in too, so you can use it!)

Check out plugins/static\_optimizes.py in your project to understand how this works. It's very easy to add your own optimizers!

**Site URL**

If you would like for your sitemap to have absolute paths you need to add a site-url key to your config.json

You can enable this by adding modifying your configuration and adding:

"site-url": "http://yoursite.com",

Note that you need to do this if you want your sitemap to be valid for Google Webmaster Tools.

**"Pretty" URLs**

If you would like to not have ".html" in your URLs, Cactus can rewrite those for you, and make "/my-page.html" look appear as "/my-page/", by creating the "/my-page/index.html" file.

You can enable this by adding modifying your configuration and adding:

"prettify": true

Note that if you're going to use this, you should definitely set your "Meta canonical" to the URL you're using so as to not hurt your search rankings:

<link rel="canonical" href="{{ CURRENT\_PAGE.absolute\_final\_url }}" />

**Nameserver configuration**

To set up a hosted zone and generate the correct nameserver records for your domain, make sure your bucket is a valid domain name, and run:

cactus domain:setup

Cactus will return with a set of nameservers that you can then enter with your registrar. To see the list again run:

cactus domain:list

If your domain is 'naked' (eg. without www), Cactus will add create an extra bucket that redirects the www variant of your domain to your naked domain (so [www.cactus.com](http://www.cactus.com/) to cactus.com). All the above is Amazon only for now.

**Extra files**

Cactus will auto generate a robots.txt and sitemap.xml file for you based on your pages.

This will help bots to index your pages for Google and Bing for example.