Eleventy Static site generator

CONCEPTS

Eleventy turns every template in the input folder into a file in the output folder; data from the cascade is merged in on the way.

Data cascade
↓ merge
Template md, njk, liquid, html
↓ layout
Layout in _includes
↓ write
_site output folder

RUNNING

npx @11ty/eleventy
Build the site from the current folder

npx @11ty/eleventy --serve
Build, serve and live-reload

npx @11ty/eleventy --input=*<dir>*
Set the input directory

npx @11ty/eleventy --output=*<dir>*
Set the output directory

FRONT MATTER & LAYOUTS

layout: *<file>*
Wrap this page in a layout

title: *<text>*
Set a page variable

permalink: *<path>*
Choose the output path

tags: *<name>*
Add the page to a collection

{{ content | safe }}
Output the page body in a Nunjucks layout

{{ page.url }}
URL of the current page

In YAML front matter, quote any value that contains template syntax, e.g. permalink: "{{ title | slugify }}/index.html".

A file named about.md is written to about/index.html unless a permalink says otherwise.

DATA CASCADE

Priority, highest first

Source	Where
`Computed data`	eleventyComputed
`Template front matter`	the page itself
`Template data file`	<page>.11tydata.json
`Directory data file`	<dir>.11tydata.json
`Layout front matter`	layout file
`Config global data`	config API
`Global data file`	_data/*.json|js

_data/*<name>***.json**
Global data, available as <name>

*<dir>***/***<dir>***.11tydata.json**
Data shared by templates in a folder

Objects and arrays are deep-merged across the cascade, so tags from a directory file and a page combine.

COLLECTIONS & PAGINATION

collections.*<tag>*
All pages carrying that tag

{% for post in collections.post %}
Loop over a collection (Nunjucks)

pagination:
Front matter block to split data into pages

CONFIG FILE

eleventy.config.js
Config file name
(or .eleventy.js, .mjs, .cjs)

export default function (eleventyConfig) {}
Config entry point (ESM)

export const config = { dir: { input, output } }
Set folders in the config

eleventyConfig.addPassthroughCopy("*<path>***")**
Copy files to output untouched

eleventyConfig.addFilter("*<name>***", fn)**
Add a template filter

eleventyConfig.addCollection("*<name>***", fn)**
Define a custom collection