Eleventy Static site generator

START HERE

`npx @11ty/eleventy`
Build into _site

`npx @11ty/eleventy --watch`
Rebuild on change

`npx @11ty/eleventy --serve`
Dev server, live reload

`npx @11ty/eleventy --quiet`
Less console noise

`npx @11ty/eleventy --input=`*`<dir>`***
Choose the source folder

`npx @11ty/eleventy --dryrun`
Run without writing files

`npx @11ty/eleventy --output=`*`<dir>`***
Choose the output folder

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

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

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.

Layouts and includes live in _includes, global data in _data (both relative to the input 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

`npx @11ty/eleventy --watch`
Rebuild on file changes, no server

`npx @11ty/eleventy --quiet`
Reduce console logging

`npx @11ty/eleventy --dryrun`
Run without writing any files

`npx @11ty/eleventy --help`
List current commands and flags

FRONT MATTER & LAYOUTS

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

`title:` ***`<text>`***
Set a page variable

`permalink:` ***`<path>`***
Choose the output path

FRONT MATTER & LAYOUTS (cont.)

`permalink: false`
Render but do not write a file

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

`date:` ***`<date>`***
Override the date used for sorting

`{{ 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>

`_data/`*`<dir>`*`/`*`<name>`*`.json`
Nested global data as <dir>.<name>

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

DATA CASCADE (cont.)

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)

`collections.post | reverse`
Newest first (non-mutating filter)

`pagination:`
Front matter block to split data into pages

`data:` ***`<path>`***
Dataset to paginate

`size:` ***`<n>`***
Items per generated page

`pagination.items`
Items on the current page

`pagination.pageNumber`
Current page index, starts at 0

`pagination.href.next`
URL of the next page

Collections sort by date ascending by default; ties fall back to the file path.

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

CONFIG FILE (cont.)

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

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

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

`eleventyConfig.addPlugin(`*`<plugin>`*`)`
Enable a plugin

RECIPES

Minimal project

```
npm init -y
npm install --save-dev @11ty/eleventy
echo '# Hello' > index.md
npx @11ty/eleventy --serve
```

Blog posts as a collection

```
# posts/posts.11tydata.json
{ "tags": "post", "layout": "post.njk" }
# then, in a template
{% for post in collections.post %}
  <a href="{{ post.url }}">{{ post.data.title }}</a>
{% endfor %}
```

Pass assets through

```
// eleventy.config.js
export default function (eleventyConfig) {
  eleventyConfig.addPassthroughCopy("css");
};
export const config = { dir: { input: "src", output: "dist" } };
```

Paginated listing

```
---
pagination:
  data: collections.post
  size: 10
permalink: "blog/{{ pagination.pageNumber }}/index.html"
---
```