Adding a course

- Add your title and description
- Add your course images
- Configure your course

Create a new directory in `src/site/content/en/learn/`. The name of this directory will be your `projectKey` which you'll use later. This will also be your course's url. For example, the directory `src/site/content/en/css` has a `projectKey` of `css` and is accessible at web.dev/learn/css.

Add an `index.md` to the root of this directory. This will be your course's landing page. You can copy the one from Learn CSS as a starter.

Finally, add an `11tydata.js` file in this directory. The file name should match the directory name. So if your directory is named `css`, you will add a file named `css.11tydata.js`.

Add this content to your `11tydata.js` file. **Be sure to update the `projectKey` so that it matches your directory name**:

```
module.exports = function () {
  // The key is used to look up the data for the course in _data.
  // e.g. A course with a key of 'a11y' would have a corresponding
  // _data/courses/a11y directory.
  const projectKey = 'YOUR_PROJECT_KEY';

  return {
    layout: 'course',
    projectKey,
    searchTag: `course-${projectKey}`,
    // Exclude course content from the /tags/ pages.
    excludeFromTags: true,
    // Exclude course content from the /authors/ pages.
    excludeFromAuthors: true,
    eleventyComputed: {
```

```
      tags: (data) => {
        const {searchTag} = data;
        let tags = data?.courses?.[projectKey]?.meta?.tags || [];
        tags = [...tags, ...data.tags];
        if (!tags.includes(searchTag)) {
          tags.push(searchTag);
        }
        return tags;
      },
      thumbnail: (data) => {
        // Use thumbnail defined in the frontmatter or in the meta.yml file.
        return data.thumbnail || data?.courses?.[projectKey]?.meta?.thumbnail;
      },
      // Use this flag to prevent the course from showing up in production.
      permalink: (data) => {
        if (process.env.ELEVENTY_ENV === 'prod') {
          return false;
        }
        return (data.page.filePathStem + '.html').replace(/^\/en\//, '/');
      },
    },
  };
};
```

Add your title and description

Edit `src/site/_data/i18n/courses.yml` so it includes a title and description for your course. Our team will handle translating your course title and description so you only need to provide an `en` key.

Add your course images

You course will need a few images for the UI.

- `src/images/courses/YOUR_PROJECT_KEY/logo.svg` (140 x 40px)
- `src/images/courses/YOUR_PROJECT_KEY/background.svg` (347 x 240px)
- `src/images/courses/YOUR_PROJECT_KEY/card.svg` (480 x 480px)

You will also need a social media thumbnail (1200 x 675px). This thumbnail should be uploaded to our image CDN and added to the course's `meta.yml` file (explained below). The thumbnail should contain the web.dev logo as a watermark (example).

Configure your course

Create the following files:

- `src/site/_data/courses/YOUR_PROJECT_KEY/meta.yml`
- `src/site/_data/courses/YOUR_PROJECT_KEY/toc.yml`

The `meta.yml` file defines important information about your course such as its publish date, title, etc.

The `toc.yml` specifies your course's table of contents. Take a look at the Learn CSS toc.yml for an example.

A simple, linear course could look like:

```
# toc.yml
- url: /learn/css
- url: /learn/css/zalgo
- url: /learn/css/clearfix
```

However, it's also possible to create nested sections:

```
# toc.yml
- url: /learn/css
- title: i18n.courses.learn.esoterica
  sections:
  - url: /learn/css/zalgo
  - url: /learn/css/clearfix
  - title: i18n.courses.learn.even_more_estoerica
    sections:
    - url: /learn/css/highest_zindex_you_can_imagine
```

A course with nested sections will still order each page linearly. When they complete a section, they'll be prompted to move to the next section in the order it appears in the `toc.yml` (regardless of nesting level).

□□□ #2
□ □□□ □□□ 2026-10-31 22:38:45 UTC

□ □□□ □□□ 2026-10-31 22:38:46 UTC