

# Advanced website features

# **Other common features**

# URLs

The website will mirror your file structure, including subdirectories

**/about.qmd**



**example.com/about.html**

**/research/project1.qmd**



**example.com/research/project1.html**

# Custom URLs

What if you want nicer URLs?

**[example.com/about/](#)**

**[example.com/research/project1/](#)**

# `index.html`

By convention, web servers treat `index.html` files in a special way.

**`example.com/about/`**

implicitly means

**`example.com/about/index.html`**

# index.qmd

If you want **example.com/about/**, put your about page in a folder named **about** and rename it to **index.qmd**.

**/about.qmd**



**/about/index.qmd**



**example.com/about/**

# User-friendly URLs

You could have a blog post like this (ew):

**/2025-10-16\_blog-post-title.qmd**



**example.com/2025-10-16\_blog-post-title.html**

Or like this (yay):

**/blog/2025/10/16/blog-post-title/index.qmd**



**example.com/blog/2025/10/16/blog-post-title/**

# Your turn

1. Make your `about.qmd` page appear at `about/`

05:00



# Page layout

## Page layouts

- **article**: Default layout with main text area between 600px–900px
- **full**: Expand main text area into the sidebar and margin area if there's no content there
- **custom**: Provide your own HTML template for the main text area

## Component widths



## YAML settings

```
format:
  html:
    page-layout: article # or "full" or "custom"
    grid:
      sidebar-width: 300px
      body-width: 900px
      margin-width: 300px
      gutter-width: 1.5rem
```

# Your turn

1. Change the page layout settings for one of your page
2. *Bonus task!* Try using the Bootstrap CSS Grid layout system with `page-layout: custom`.
  - [See the documentation here](#)
  - [See an IRL example here \(and code here\)](#)

## Advice:

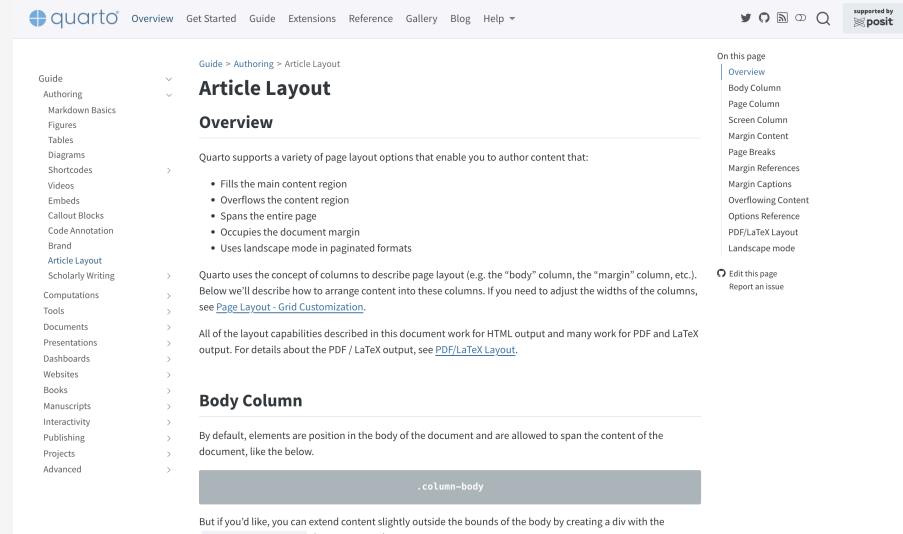
- Don't try to make all these changes at once! Remember the workflow: edit, save, preview, repeat.
- Use the documentation at [Quarto.org > Guide > Advanced > Page Layout](#).

10:00

# Article layout

Quarto offers *a ton* of article-specific layout options:

- Make content go into the margin
- Make content go across the whole page
- Put content in the margins



# Your turn

1. Add an image that spans across the whole page
2. Add some text to the margin

## Advice:

- Don't try to make all these changes at once! Remember the workflow: edit, save, preview, repeat.
- Use the documentation at [Quarto.org > Guide > Authoring > Article Layout](#).

10:00

The background of the image is a dense, repeating pattern of hexagons in various shades of red, ranging from light pink to a deeper, more saturated red. The hexagons are of different sizes and are arranged in a way that creates a complex, crystalline texture.

# Freezing

# Re-running code

your-document.qmd

```
1 ---
2 title: "My Complex Document"
3 format: html
4 ---
5
6 Here we run all our models and ther
7
8 ```{r}
9 # Intensive code!
10 ```
11
12 ```{r}
13 # Intensive code!
14 ```
15
16 ```{r}
17 # Intensive code!
18 ```
```

- Sometimes you'll have a lot of code chunks that take a long time to run.
- Quarto will re-run these chunks each time you render the site.
- Good for reproducibility! Bad for speed!

# Freeze computational output

your-document.qmd

```
1 ---
2 title: "My Complex Document"
3 format: html
4 execute:
5   freeze: auto
6 ---
7
8 Here we run all our models and ther
9
10 ```{r}
11 # Intensive code!
12 ```
13
14 ```{r}
15 # Intensive code!
16 ```
17
18 ```{r}
```

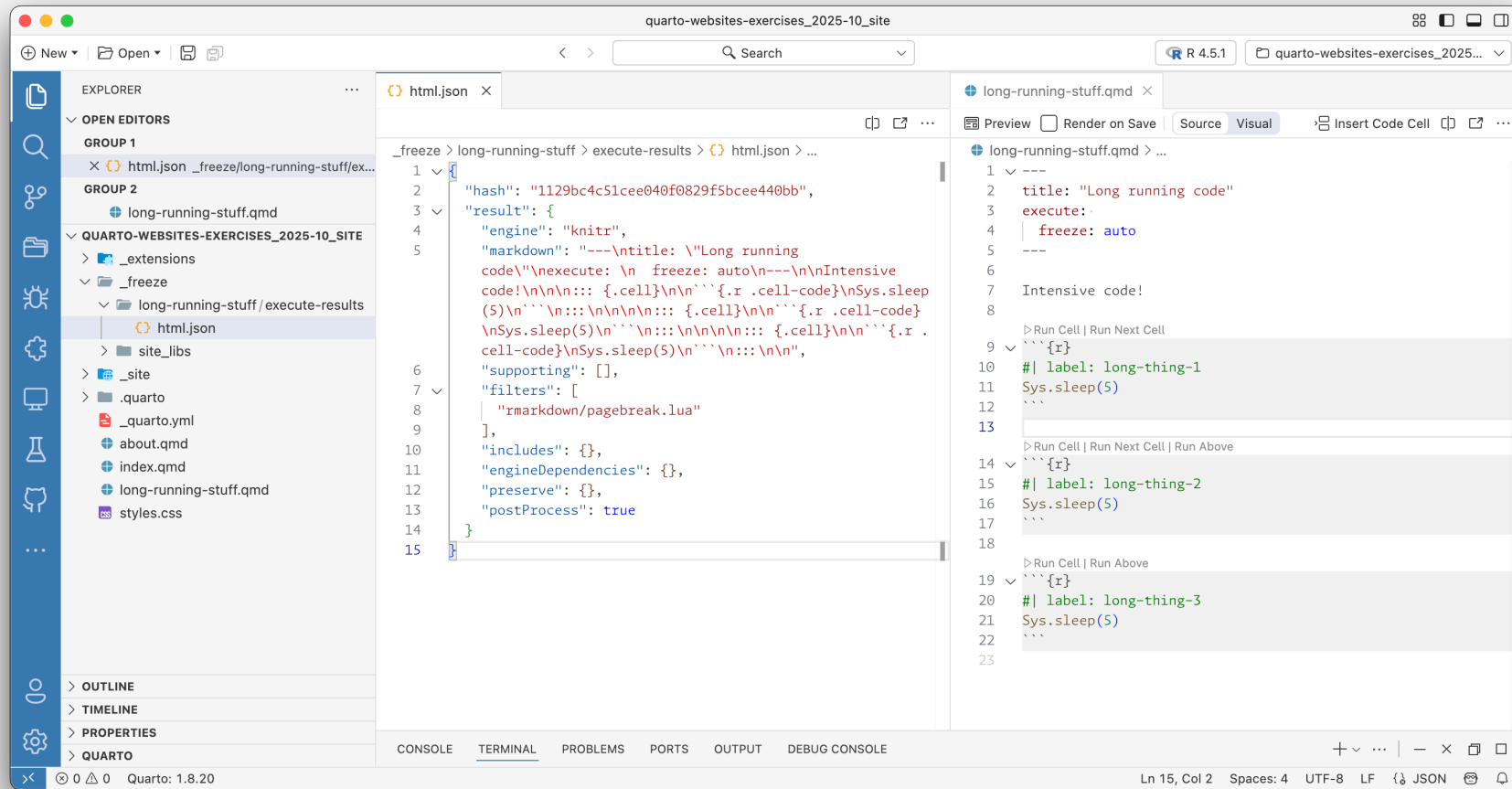
Use `freeze: auto` to only re-run the code if the source file changes.

← Individual files  
or site-wide ↓

\_quarto.yml

```
1 project:
2   type: website
3
4 execute:
5   freeze: auto
6
7 website:
8   title: "Example site"
```

Code output is stored in `_freeze`





# Freezing $\neq$ caching

- Freezing *only* stores the text of the code output.
- Changing the file will force a re-run.
- Consider caching if you don't want the code to re-run (see **documentation**):

```
execute:  
  cache: true
```

# Listings

# Automatic content

We can automatically add all the pages in a folder to a sidebar:

```
_quarto.yml
1 website:
2   sidebar:
3     contents:
4       - href: index.qmd
5         text: Home
6       - section: Research
7         contents:
8           - research/project1.qmd
9           - research/project2.qmd
10          - research/project3.qmd
```

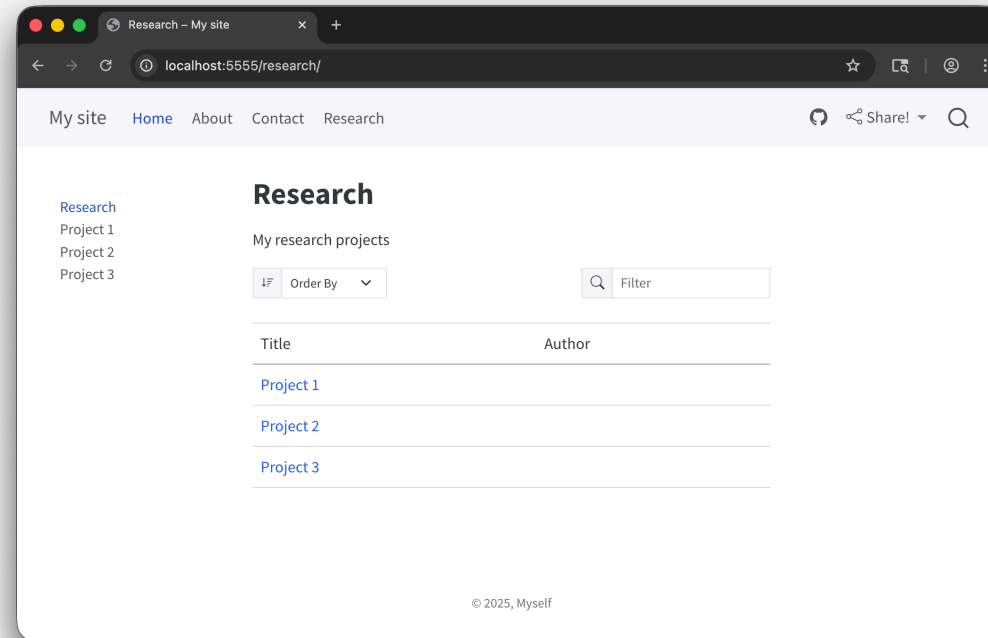
```
_quarto.yml
1 website:
2   sidebar:
3     contents:
4       - href: index.qmd
5         text: Home
6       - section: Research
7         contents: research/*
```

## What if we want fancier automatic pages?

# Listing pages

research.qmd

```
1 ---
2 title: "Research"
3 listing:
4   contents: "*.qmd"
5   type: table
6 ---
```



# Your turn

1. Edit `research/index.qmd` to be a listing page, showing all the pages in the `research` folder.
2. Experiment with changing the listing type to `default`, then `table`, then `grid`.

## Advice:

- Don't try to make all these changes at once! Remember the workflow: edit, save, preview, repeat.
- Use the documentation at [Quarto.org > Guide > Websites > Listing Pages](#).

10:00

# Metadata in listings

**Listing Example**

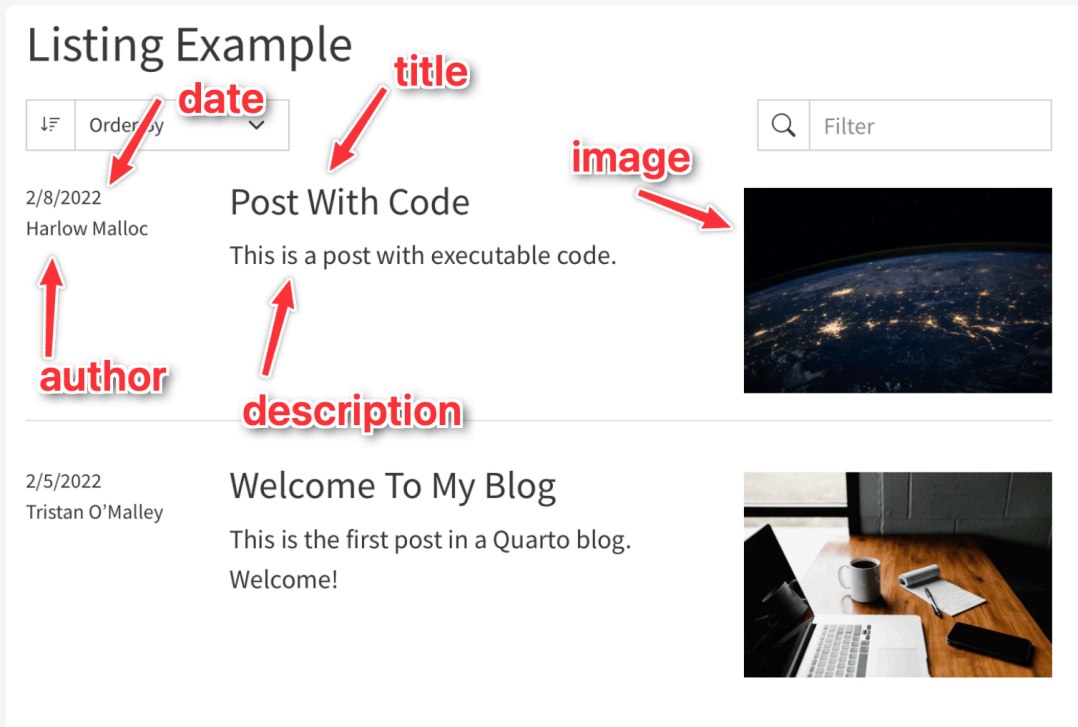
2/8/2022  
Harlow Malloc

**Post With Code**  
This is a post with executable code.

**2/5/2022**  
Tristan O'Malley

**Welcome To My Blog**  
This is the first post in a Quarto blog.  
Welcome!

**image**



post.qmd

```
1 ---
2 title: "Research"
3 author: "Your name"
4 description: "Something"
5 date: 2025-10-06
6 image: "img/thing.jpg"
7 ---
```

# Your turn

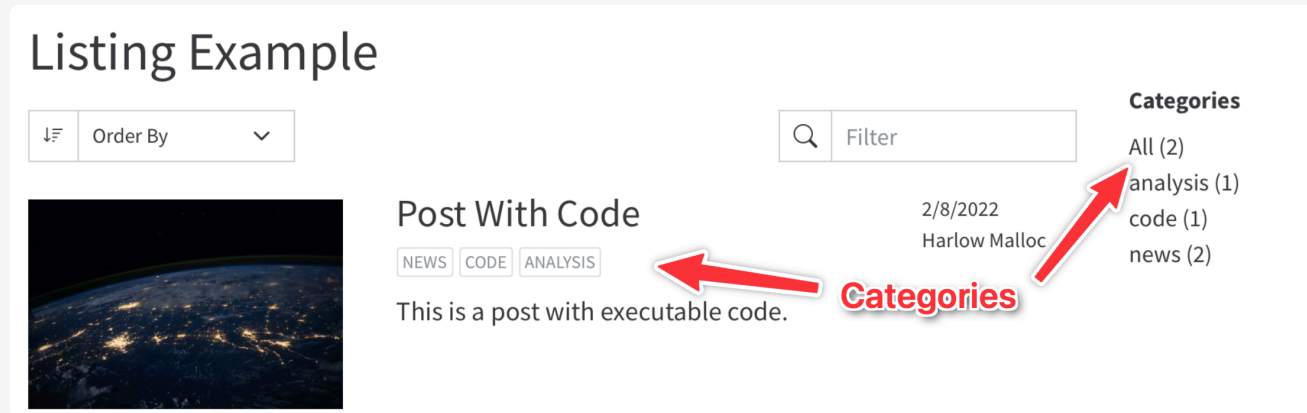
1. Add metadata to your different research `.qmd` files, including images. See how those are reflected in the listing page after rendering.
2. *Bonus task!* Modify how the listing page is sorted. Sort it by descending title.
  - [See the documentation here](#)
3. *Bonus task!* Change how the dates are shown in the listing page with the `date-format` option.
  - [See the documentation here](#)

## Advice:

- Don't try to make all these changes at once! Remember the workflow: edit, save, preview, repeat.
- Use the documentation at [Quarto.org > Guide > Websites > Document Listings](#).

10:00

# Categories



## Enable categories in the listing settings

```
listing-page.qmd
1 listing:
2   categories: true
```

## Specify categories in the document

```
post.qmd
1 ---
2 title: "Research"
3 categories:
4   - statistics
5   - r
6   - tidyverse
7 ---
```



# Your turn

1. Add categories to your different research `.qmd` files. See how those are reflected in the listing page after rendering.
2. *Bonus task!* Change the category list on the listing page to appear as a word cloud.
  - [See the documentation here](#)

## Advice:

- Don't try to make all these changes at once! Remember the workflow: edit, save, preview, repeat.
- Use the documentation at [Quarto.org > Guide > Websites > Document Listings > Categories](#).

10:00

# Extra advanced bonus stuff

## Multiple listings

Include lists of different document types on the same page (**example** + **code**)

## Custom listings

Go beyond `default`, `table`, and `grid` and make your own template (**example** + **code**)

## YAML lists

Generate listings of items in a YAML file instead of `.qmd` files (**example**)

# Your turn

## Make a blog!

1. Create a new folder named `blog` with a file named `index.qmd` in it.
2. Make a few folders and files named `blog/2025/10/16/thing/index.qmd`, etc.
3. Add some fake content in the blog posts. Add images, titles, descriptions, etc.
4. Make the blog posts appear as a listing page at `blog/index.qmd`.
5. Add `blog/index.qmd` to your top navigation bar.

## Advice:

- Don't try to make all these changes at once! Remember the workflow: edit, save, preview, repeat.
- Use the documentation at [Quarto.org > Guide > Websites > Document Listings](#).

20:00

# Extensions

# Extensions

Quarto has **extensions** that let you add custom...

- shortcodes
- filters
- document formats and templates
- brands

# Shortcode extensions

## Font Awesome icons (**site** + **extension**)

Terminal

```
quarto add quarto-ext/fontawesome
```





Text	Output
<code>{{&lt; fa home &gt;}}</code>	
<code>{{&lt; fa wand-magic-sparkles &gt;}}</code>	
<code>{{&lt; fa brands apple &gt;}}</code>	
<code>{{&lt; fa brands r-project &gt;}}</code>	
<code>{{&lt; fa globe &gt;}}</code>	

# Use in Markdown:

my-document.qmd

```
- {{< fa book >}} [Book page](https://www.bayesrulesbook.com/)
- {{< fa person-chalkboard >}} [Teaching resources](https://bayes-rules.github.io/)
- {{< fa brands r-project >}} [R package](https://github.com/bayes-rules/bayesrules)
- {{< fa brands youtube >}} Alicia Johnson, ["Bayes Rules! An Introduction to Bayesian Modeling with R with Alicia Johnson"], presentation for R-Ladies Philly on October 18, 2021
```



-  **Book page**
-  **Project page**
-  **R package** and **package documentation**
-  Alicia Johnson, **"Bayes Rules! An Introduction to Bayesian Modeling with R with Alicia Johnson"**, presentation for R-Ladies Philly on October 18, 2021

# Use in YAML:

\_quarto.yml

```
website:  
  navbar:  
    left:  
      - text: "{{< fa person-chalkboard >}} Teaching"  
        href: teaching.qmd  
      - text: "{{< fa flask >}} Research"  
        href: research.qmd  
    right:  
      - text: "{{< fa brands google-scholar >}}"  
        href: "https://example.com"
```



Example site

 Teaching

 Research





# Filter extensions

## Preview Color (**extension**)

Terminal

```
quarto add mcanouil/quarto-preview-colour
```

Enable in YAML, either for one page or site-wide:

```
filters:  
  - preview-colour
```

We use #1696d2 for Democrats and #db2b27 for Republicans.



We use #1696d2🟦 for Democrats and #db2b27🔴 for Republicans.

# Lots of others!

There are hundreds of extensions: find them all at [Quarto.org > Extensions](#) or at [the Quarto Extensions directory](#).

Here are some common extensions for websites specifically:

Extension	Description
<a href="#">Font Awesome</a>	Include <a href="#">Font Awesome</a> icons
<a href="#">Academicons</a>	Include <a href="#">academicons</a> icons
<a href="#">Custom Callouts</a>	Go beyond Quarto's default <a href="#">note</a> , <a href="#">warning</a> , <a href="#">important</a> , <a href="#">tip</a> , and <a href="#">caution</a> callouts
<a href="#">Preview Color</a>	Add little circles to show hex colors
<a href="#">Embed PDFs</a>	Easily embed PDF files in HTML documents
<a href="#">Downloadthis</a>	Create download buttons

# Your turn

1. Install the [Font Awesome extension](#) in your website project.
2. Use Font Awesome shortcodes:
  - Include a Font Awesome icon in your navigation bar
  - Include a Font Awesome icon in the text of one of your pages
3. Install another Quarto extension and play around with it.




## Advice:

- Don't try to make all these changes at once! Remember the workflow: edit, save, preview, repeat.
- Use the documentation at the [Font Awesome extension page](#) and [Quarto.org > Extensions](#).

10:00



# Course outline

-  ~~Intro to Quarto~~
-  ~~Creating basic websites~~
-  ~~Advanced website features~~
- Publishing
- Customization and branding
- Interactivity