How to build a responsive website from scratch?

1. Sketch out how the layout will look on mobile, tablet, and desktop.
2. Start coding the basic layout, using semantic HTML and CSS.
3. Going section by section, build the rest of the layout.
4. In your CSS, follow a mobile-first approach, creating the styles for the smallest widths, then progressively greater widths.

Wireframe the layout

Things to note:

In the mobile version, all the sections basically stack on top of one another in one long column.

In the tablet version, the main section can now fit two columns across.

In the desktop version, we could have the website content extend full-width on a large monitor, but that would make it pretty difficult to read and scan the content. It would be like watching a movie in the theater from the front seats. For this reason, we cap the content at a certain width and center it on the webpage.

Map out the layout with HTML elements.

**Define the container**

We begin by defining a container, in which we place three elements. A header, a main, and a footer. Here’s the page structure:

<div class="flex-container">

<header></header>

<main></main>

<footer></footer>

</div>

Start adding basic CSS properties

The Universal Selector is the \* in CSS. Literally the asterisk character. A common use of this selector is as a reset. We’ve talked about how HTML elements have their own CSS styles...so we use this to reset all the margins and padding.

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

Next, let’s discuss how to get our website to fill the entire viewport

html,

body {

height: 100%;

}

body {

font-family: "Roboto", sans-serif;

min-height: 100vh;

background-color: papayawhip;

}

Let’s give some blanket styles to all the sections for now to make it easier for us to see what we’re doing as we build:

main, header, footer {

padding: 20px;

border: 1px solid #000000;

color: #141414;

}

Next, we apply the flexbox properties that will move our elements into place:

.flex-container {

display: flex;

flex-direction: column;

height: 100%;

}

header,

footer {

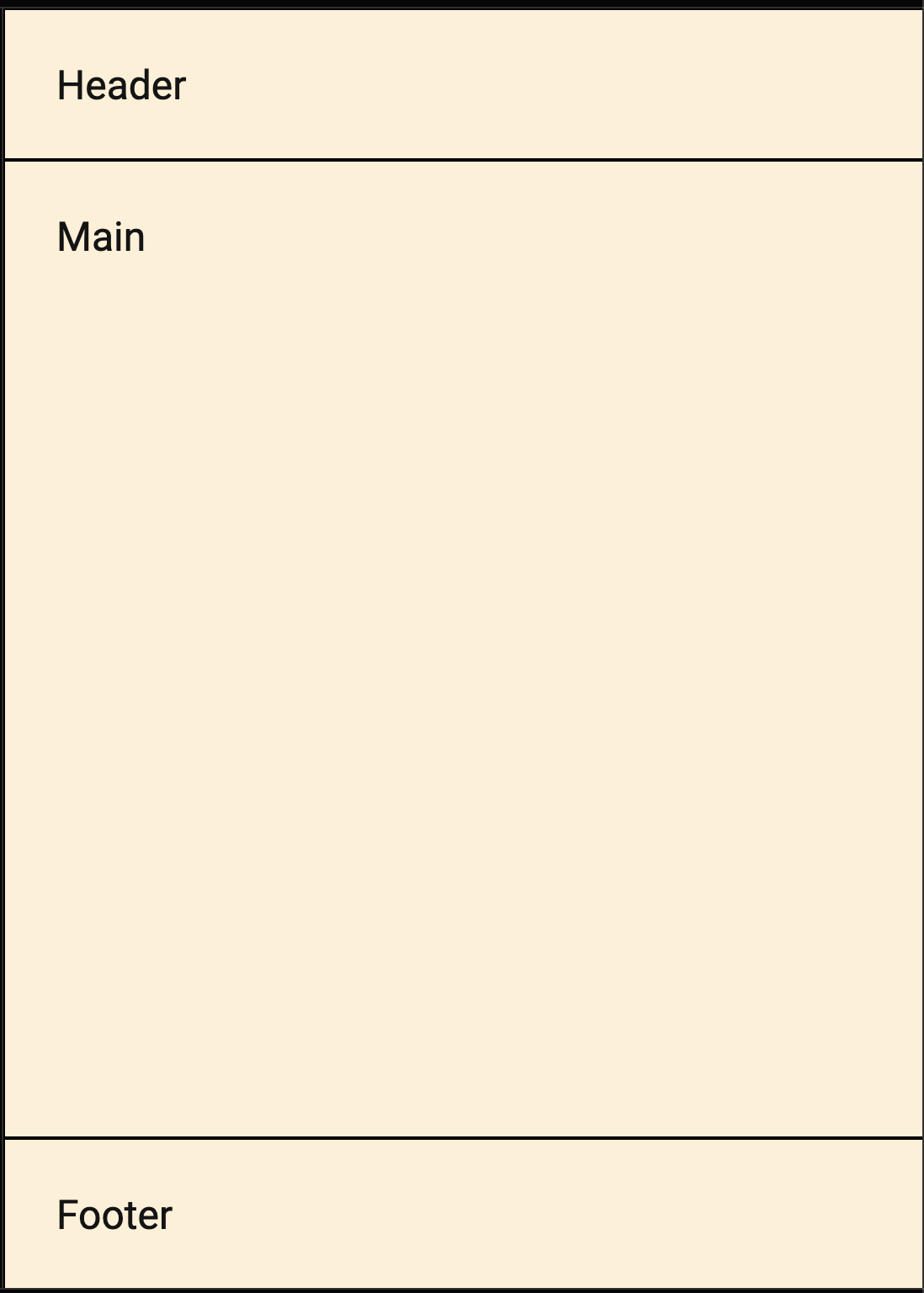
height: 60px;

}

main {

flex: 1;

}



Build out section by section

The Header

A header is usually located at the top of the website (or right below a top navigation menu). It often contains a logo or the website name:

<header>

<h2>houses or museums?</h2>

</header>

Navigation

A navigation bar contains a list of links to help visitors navigating through your website:

The Main Content

The layout in this section often depends on the target users. The most common layout is one (or combining them) of the following:

* 1-column (often used for mobile browsers)
* 2-column (often used for tablets and laptops)
* 3-column layout (only used for desktops)

<main>

<div class="title">

<h2>houses or museums?</h2>

</div>

<section id="english\_content" class="english">

<!-- put english content here →

<div class="column"></div>

<div class="column"></div>

</section>

<div class="title">

<h2>casas ou museus?</h2>

</div>

<section id="portuguese\_content" class="portuguese">

<!-- put portuguese content here →

<div class="column">

<p class="pullquote">Primeiro as Casas ou Museus?</p>

<p>Tudo de uma só vez: as casas, as escolas, os museus, as ..</p>

</div>

<div class="column"></div>

</section>

</main>

What do we want? We want to divide the content up into columns.

The Footer

The footer is placed at the bottom of your page.

<footer>

<p>Made with &#10084; in New York City.</p>

</footer>

Mobile first media query set up

In a mobile first layout we go from small to large and we use min-width in the media query.

/\* two column \*/

@media (min-width: 450px) {

.flex-container {

background: orange;

height: 100%;

}

main {

display: flex;

flex-direction: column;

}

section {

display: flex;

}

.column {

flex: 1;

}

}

/\* three column \*/

@media (min-width: 960px) {

.flex-container {

background: green;

height: 100%;

}

main {

display: flex;

flex-direction: column;

}

}

You get everything set up structurally BEFORE you hop into the detailed styling.