Basic HTML Code

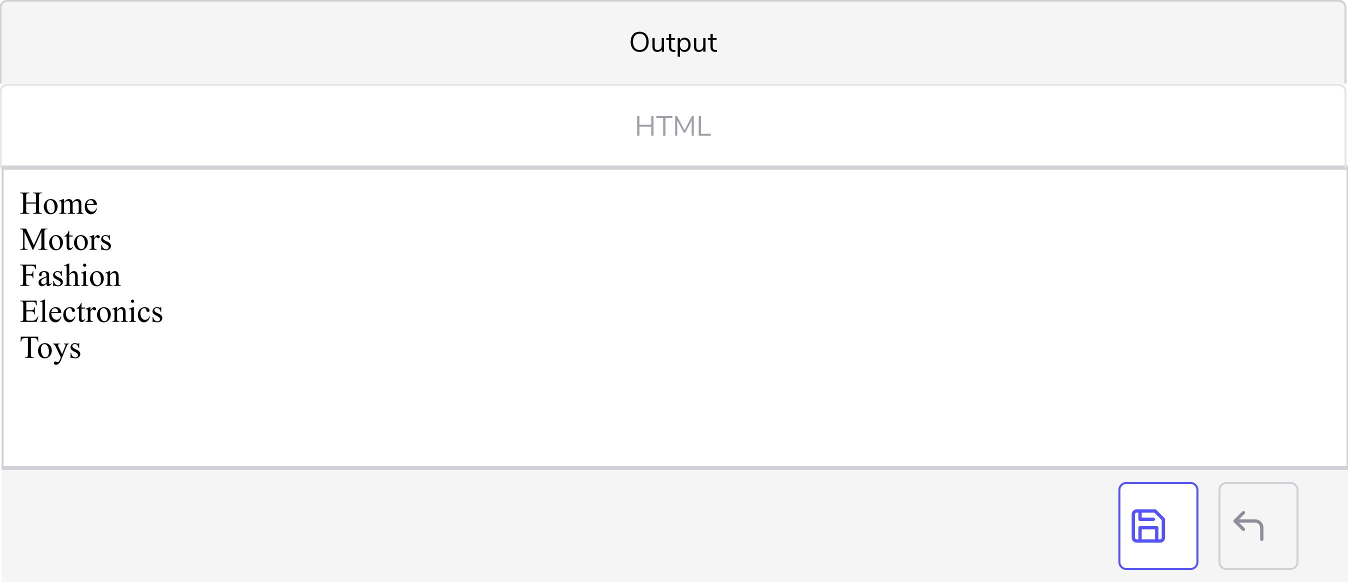
Let's start out with the basic HTML code to design a layout for our top-level menu.

WE'LL COVER THE FOLLOWING

- Dynamic vs Static HTML

Let’s dive right into the top-level menu by putting the elements into HTML.

When considering how to structure HTML, you should be thinking about how to group the things you want to build. For the top-level menu, we can create a group for the menu that encompasses the entire component and a group for each menu item. Let’s create some `div`s for our groups.



```
Home
Motors
Fashion
Electronics
Toys
```

We have a basic HTML layout for a menu, and because this will be static (it won’t change), we can define it directly in our HTML. This is the equivalent of “hard-coding” information, meaning that rather than relying on some external source for the menu items, these menu items are listed directly in the front-end source code.

Dynamic vs Static HTML

The alternative is to have data be taken from the server. Take a moment to think about when we'd want to hardcode data to HTML. (Don't worry if you don't know the answer.)

Q

When is it appropriate to hardcode data into HTML?

COMPLETED 0%

1 of 1

When data is different for each user, we have no way of implementing that in plain HTML. It certainly is possible with some server-side rendering, though. Server-side rendering is when the HTML is passed through the server first and has some data populated dynamically instead of being the same for every user. For example, `<div id="user">{{user.username}}</div>` might be a line in some HTML that's passed through a templating engine on the server to populate the portion in the braces before being served to clients (don't worry if this syntax looks foreign, it's not JavaScript). In any case, that's not considered hardcoding.

When the data set is large, however, we'd be repeating a lot of code to put it all on the HTML. This violates not only code hygiene by making a page unwieldy to scroll through, but also the DRY (Don't Repeat Yourself) principle, both of which hint at a more maintainable way to implement this.

Finally, data appearing on every page is not enough reason by itself to hardcode into the HTML. For example, a username shown on every page is different depending on the user.

When a user wants a page, loading data onto that page in server-side rendering can add time. For data that is rarely updated, manageable in size, and is the same for every user, hardcoding provides the benefit of having the data immediately available (since HTML loads before JavaScript).

In the next lesson, we’ll go over some styling conventions.