

Big Project

Part 1



What are we going to do today?

<

>



Go over some info

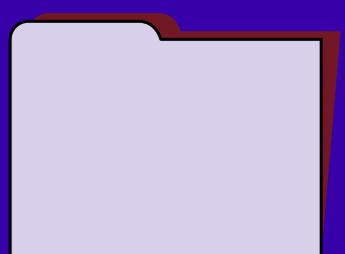
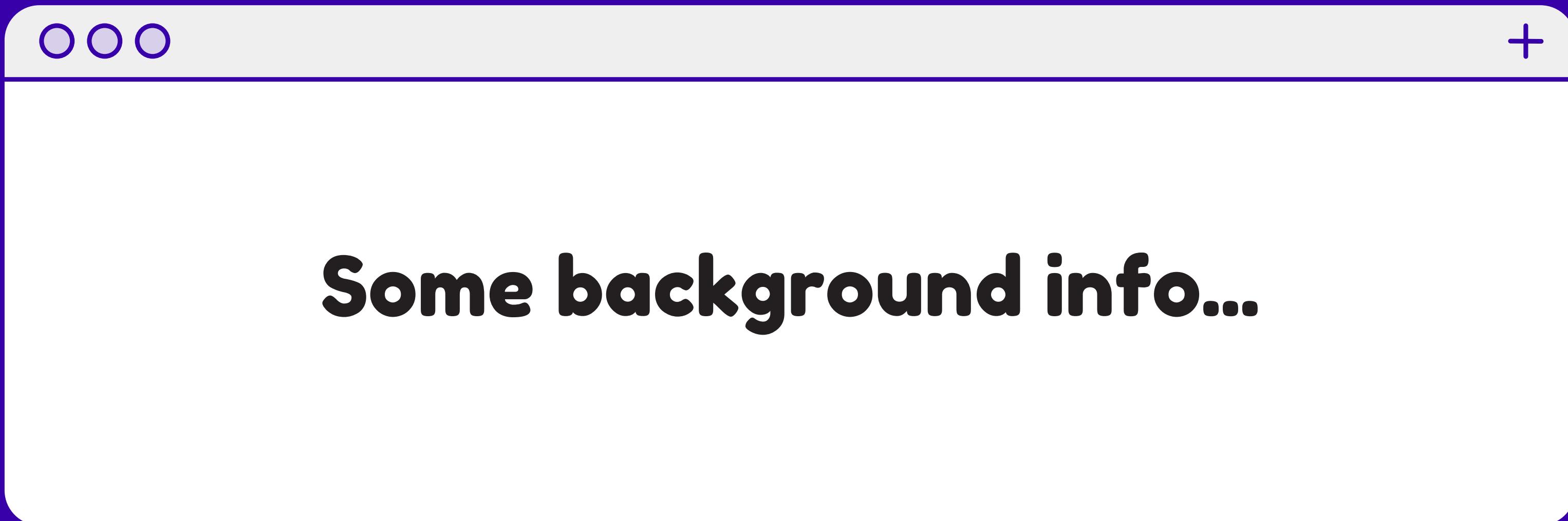
Build the UI

Flexbox

Adapt scrolling

Personalize UI

We hope you are having fun! ;)



What is our Kanban board, really?

- A series of columns where tasks go from one column to the next.
- In code, we can represent this as **unordered lists** to build the four columns.
 - Each column will then have its own unordered list of task items.



Why are the columns and items represented as unordered lists?

But how do we implement this in code?

- HTML: we add all the structure for the columns within our `<body>` `</body>`
- We need to create a "container" for our columns:
 - Check out the template in CSS. Can you find any class called `drag-container`?



We said each column is an unordered list...

- HTML
 - unordered list =
 - list items =
- We directly assign class to any type of tag by:
 - span.header
 - li.drag-column.backlog-column

* * * Do it for * * *
* * * yourself * * *

Some more HTML

- .class-name adds division with class "class-name"
- Tag span is implemented for headers
- To add content:
 - div with classes
 - div with id (use # instead of .)

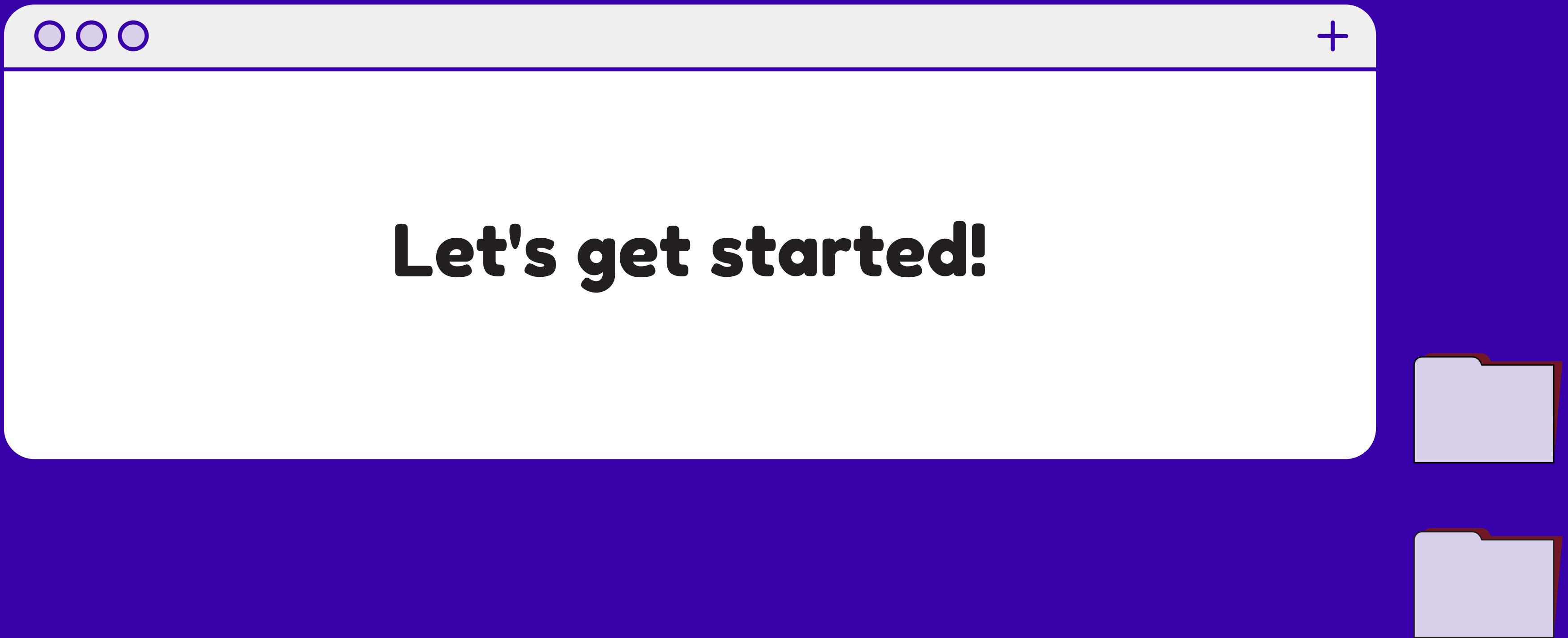
Remember we use classes declared in CSS to create the structure in HTML! (Usually the other way around you choose class name in html and then style / reference it in css)



A screenshot of a code editor displaying a React component. The component is defined with state and a render method. The render method returns a `React.Fragment` containing a `<div>` with the class `py-5`, another `<div>` with the class `container`, a `<Title>` component, a `<div>` with the class `row`, a `<ProductConsumer>` component, and a `<div>` with the class `col`. The `<ProductConsumer>` component has a prop `value` with a callback `console.log(value)`. The code uses syntax highlighting with colors like purple for state, green for components, and blue for classes.

```
state={products: storeProducts}
}
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="container">
          <Title name="our" title="product">
            <div className="row">
              <ProductConsumer>
                {(value) => {
                  | console.log(value)
                }}
              </ProductConsumer>
            </div>
          </div>
        </div>
      </React.Fragment>
    )
}
```

We hope you are having fun! ;)

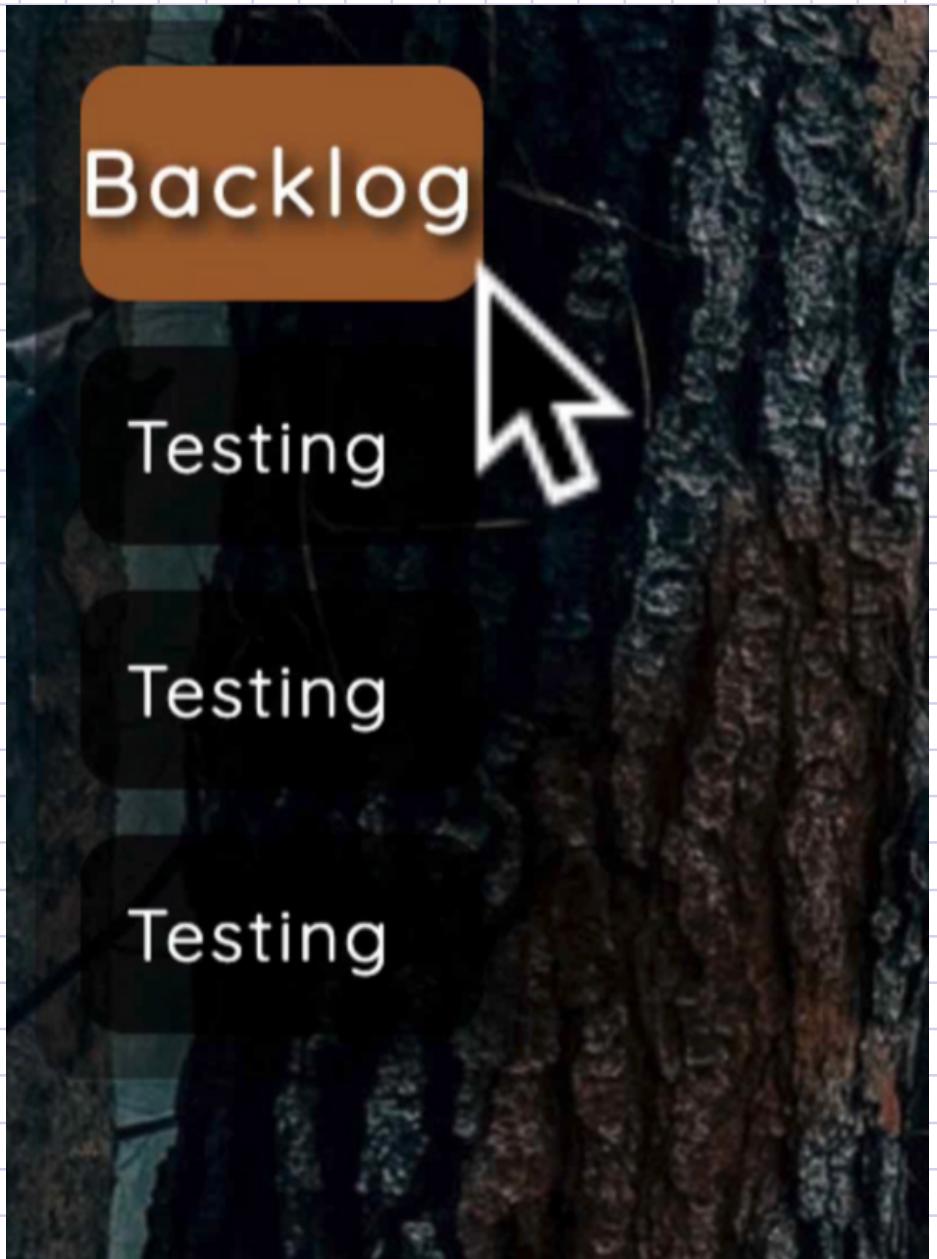


If you were not here last week ...



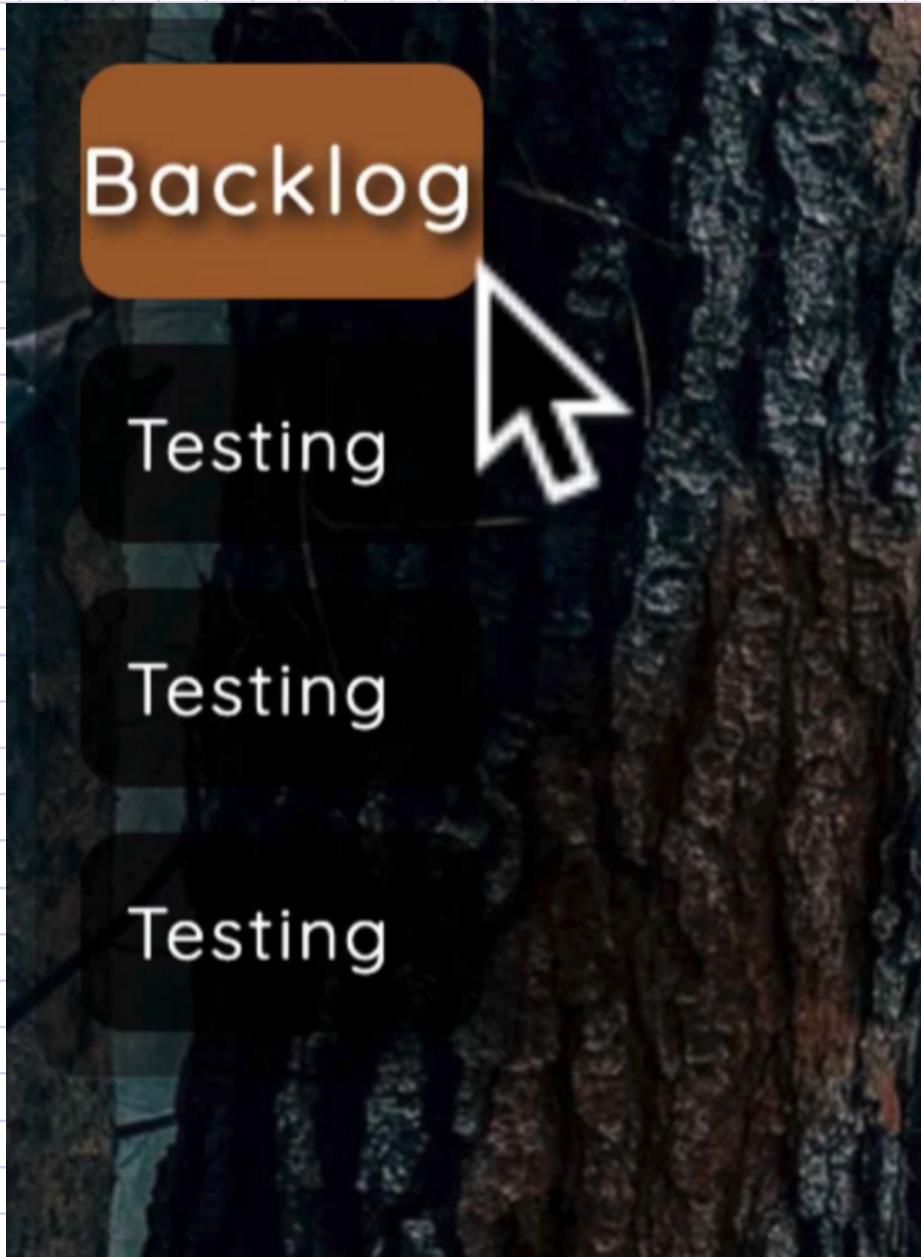
Building Columns

Our first column!

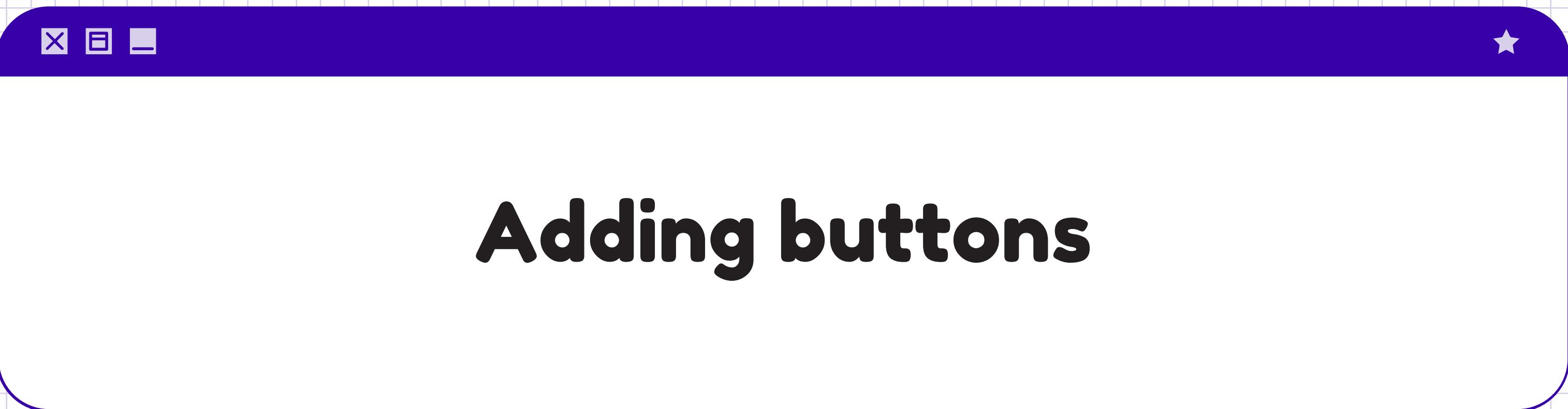


- Below our header/title create a division with class drag-container to hold our columns.
 - Inside our container, create a list of class drag-list.
- Create our backlog column:
 - inside drag-list create section with classes drag-column and backlog column
 - in that section create a class header using and inside that add a name of the column using <h1>
- Now for the content
 - In drag-column create section with id "backlog-content"
 - Create an unordered list inside backlog-content with id "backlog-list" and class "drag-item-list"
 - Add list item with class "drag-item" and text you like (Ex. Test)

Any problems?

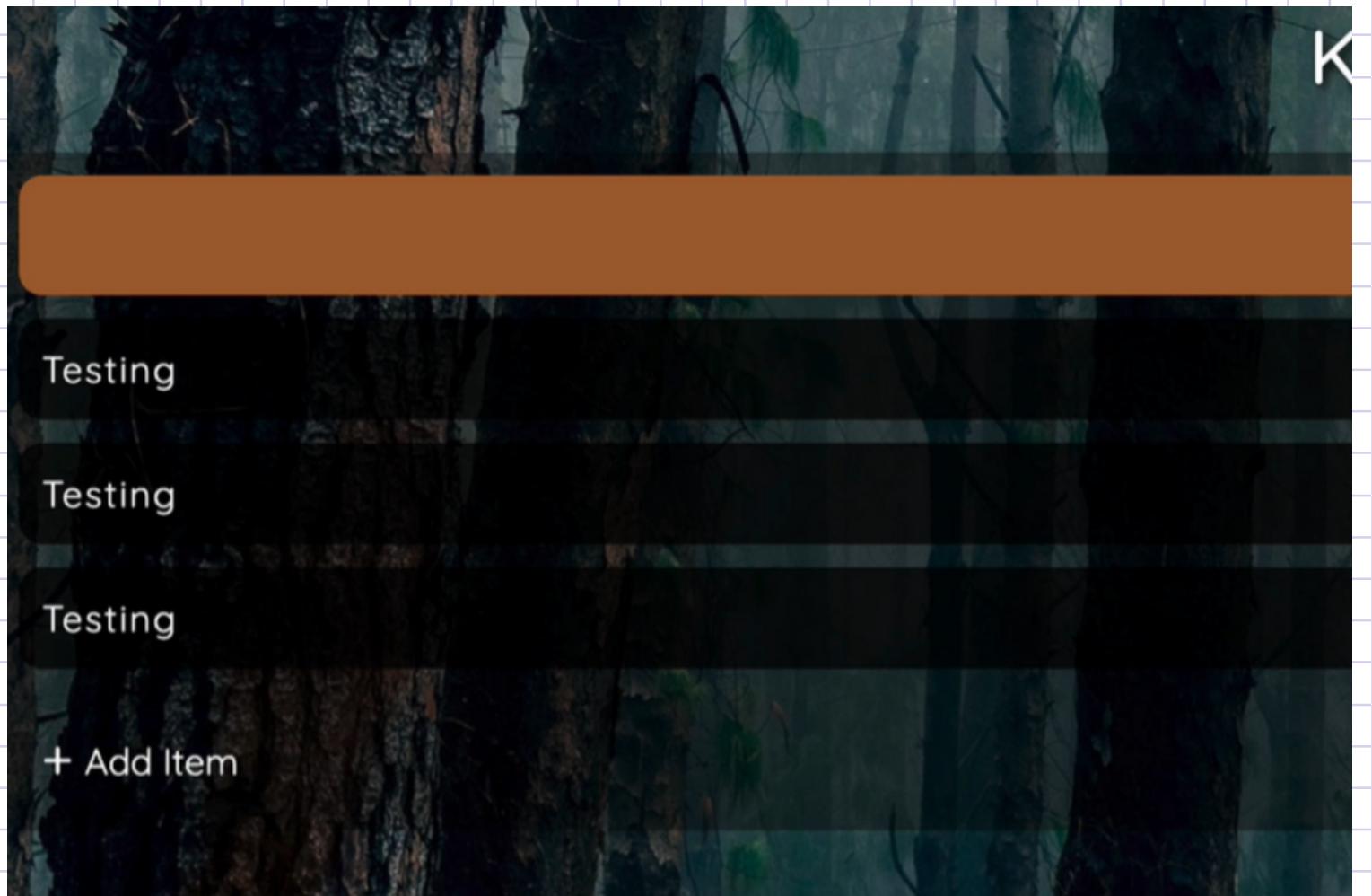


- We want our columns to fill the space in the web page.
 - This means we have to change something in our CSS file...
-
- Find the class: `.drag-column`
 - This is the class that is on every column in our board!
 - We have to add `flex:1;`
 - Evenly divide space between any drag-columns!



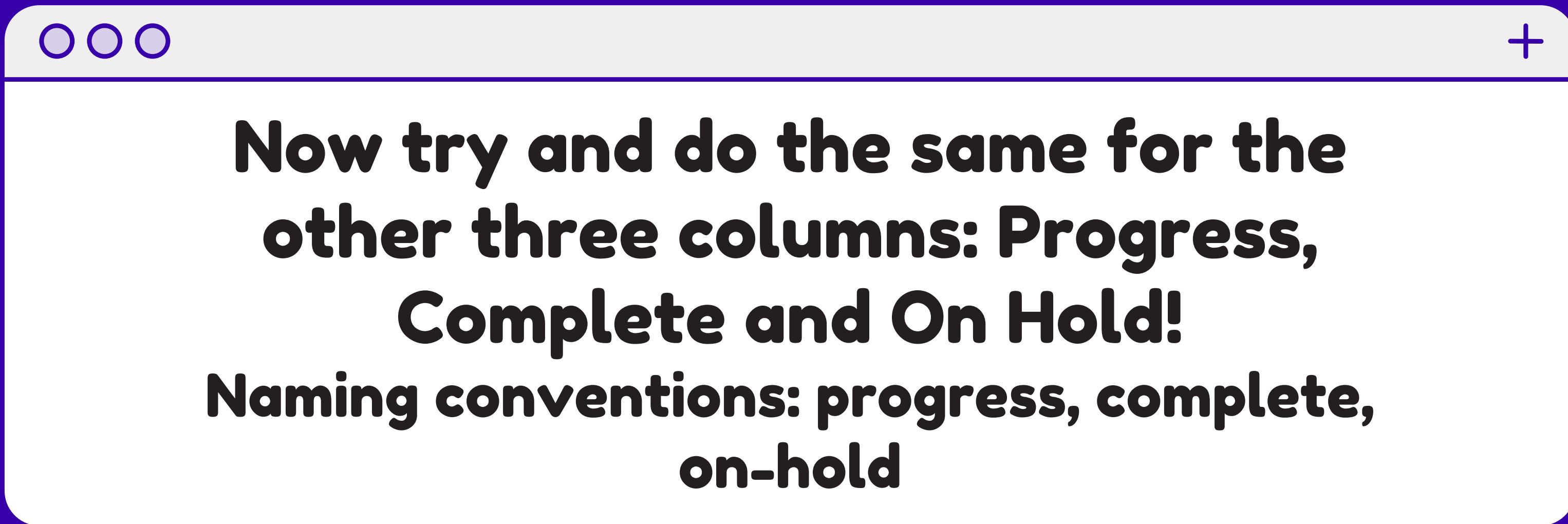
Adding buttons

Adding buttons



- We want to add a button group
- In backlog-column create division class "add-btn-group"
 - Inside add another div for "add-btn"
 - Create a span with class "plus-sign"
 - Create another span with text: Add Item
 - That's it for our add item!
 - Inside add-btn-group create a new section for button with class "add-btn solid"
 - Create a span with text "Save Item"
 - Inside add-btn-group create a new div class "add-container"
 - Inside this div, create a new div with class "add-item"

We hope you are having fun! ;)

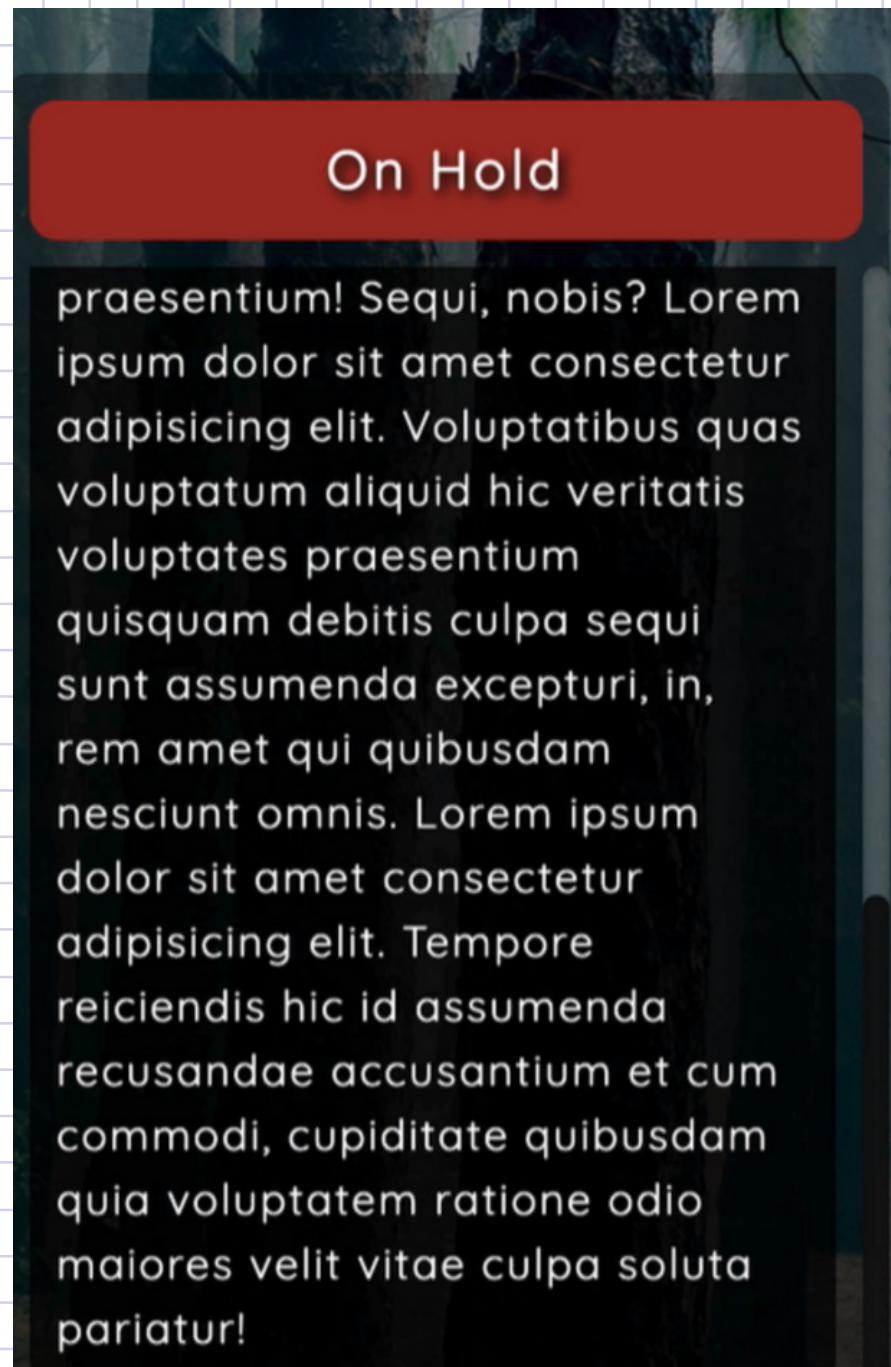


**You can copy-paste to make it a bit easier!
Be careful where the column starts.**



Adding scrolling

Scrolling



- Try adding a **VERY** long text to one of your test items! What happens?
- To solve this, we want to use a custom class from our CSS file called **.custom-scroll!**
- Where do you think we need to add this? Try it out!
- Hint: We want to have scrolling **within** our columns. What **<div>** holds our unordered list of items?

