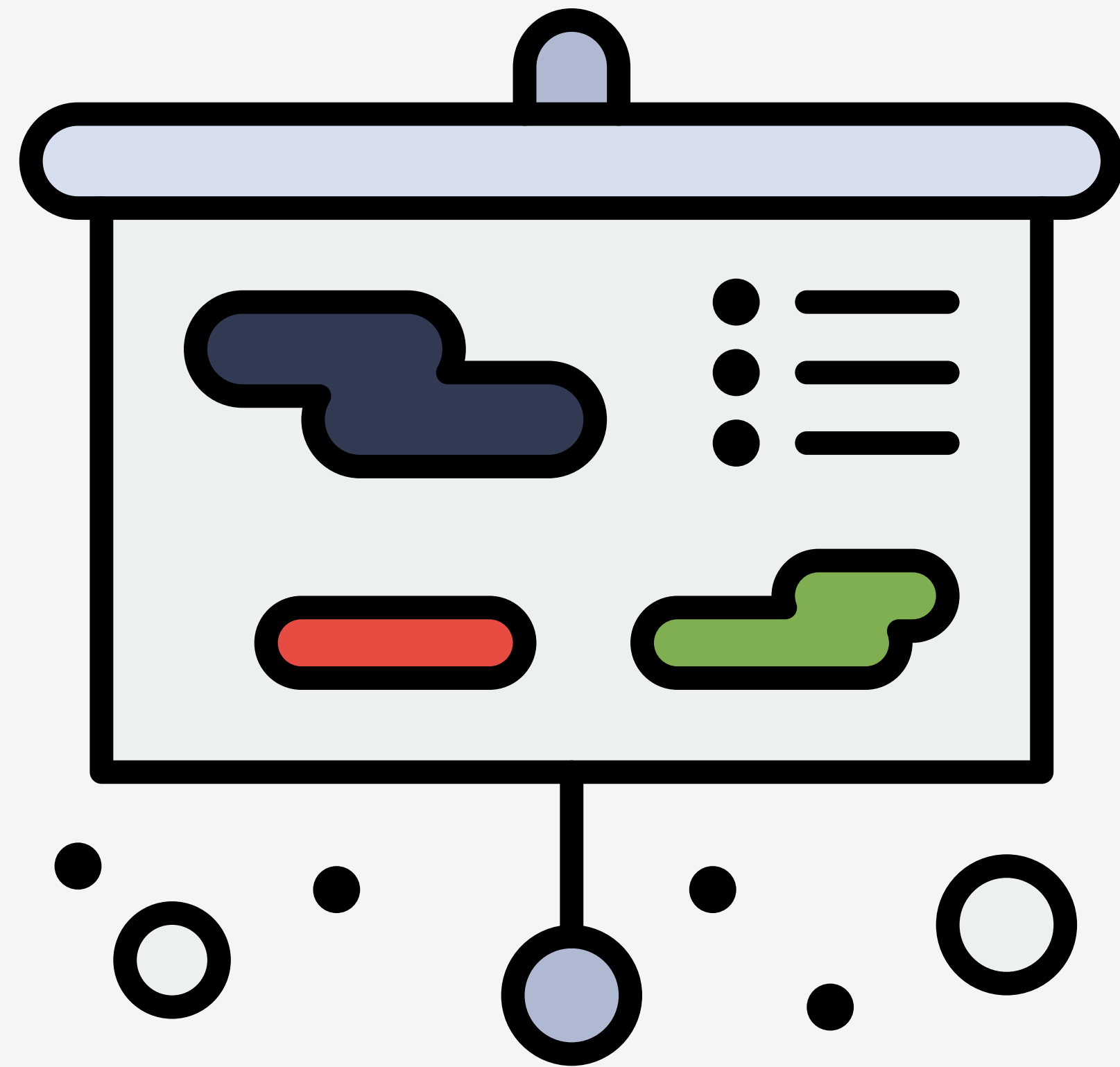

RESPONSIVE WEB DESIGN II

Lecture 17

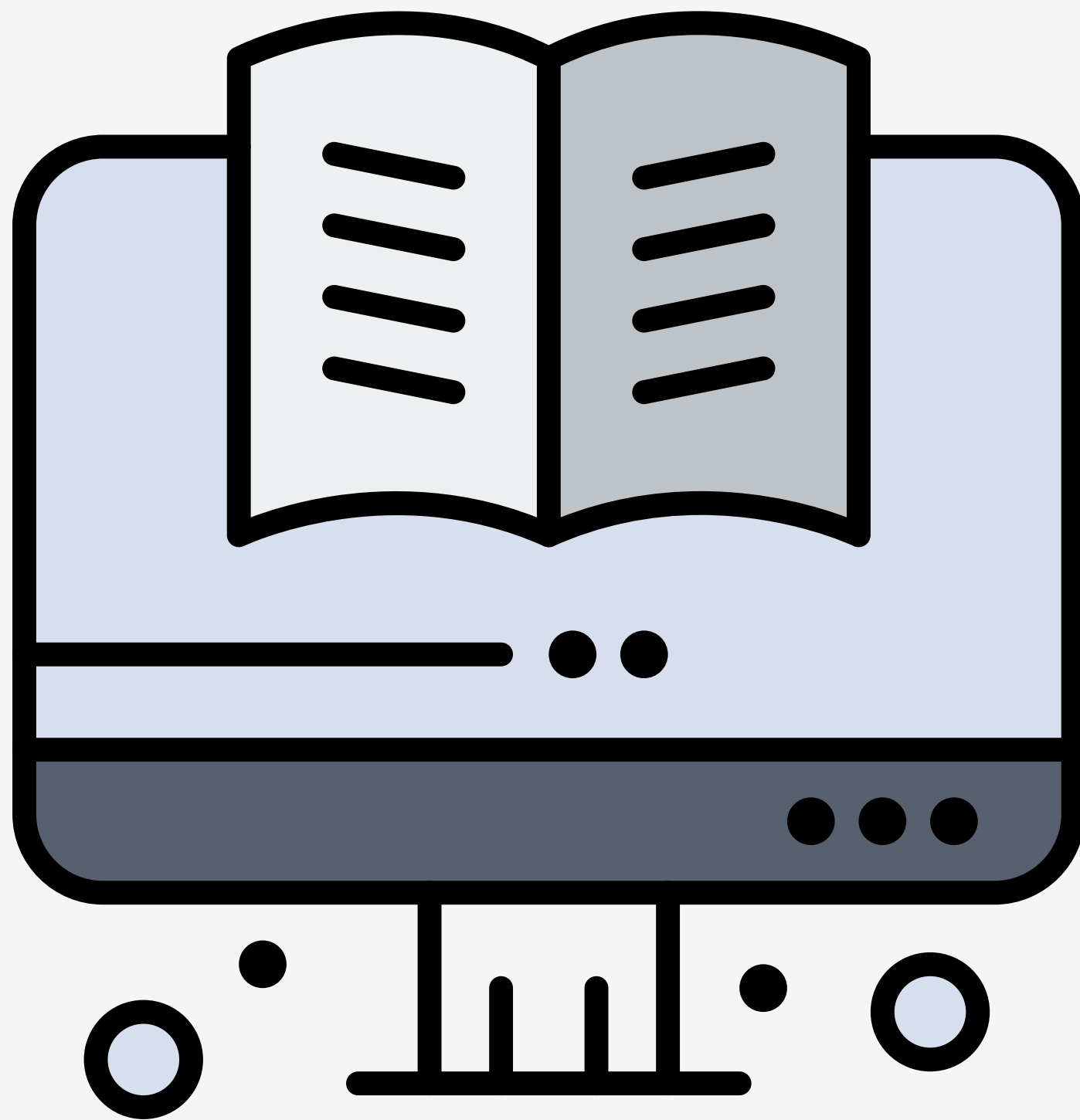
TODAY'S TOPICS



- Introduction to Sass
- **Project:** Web Portfolio

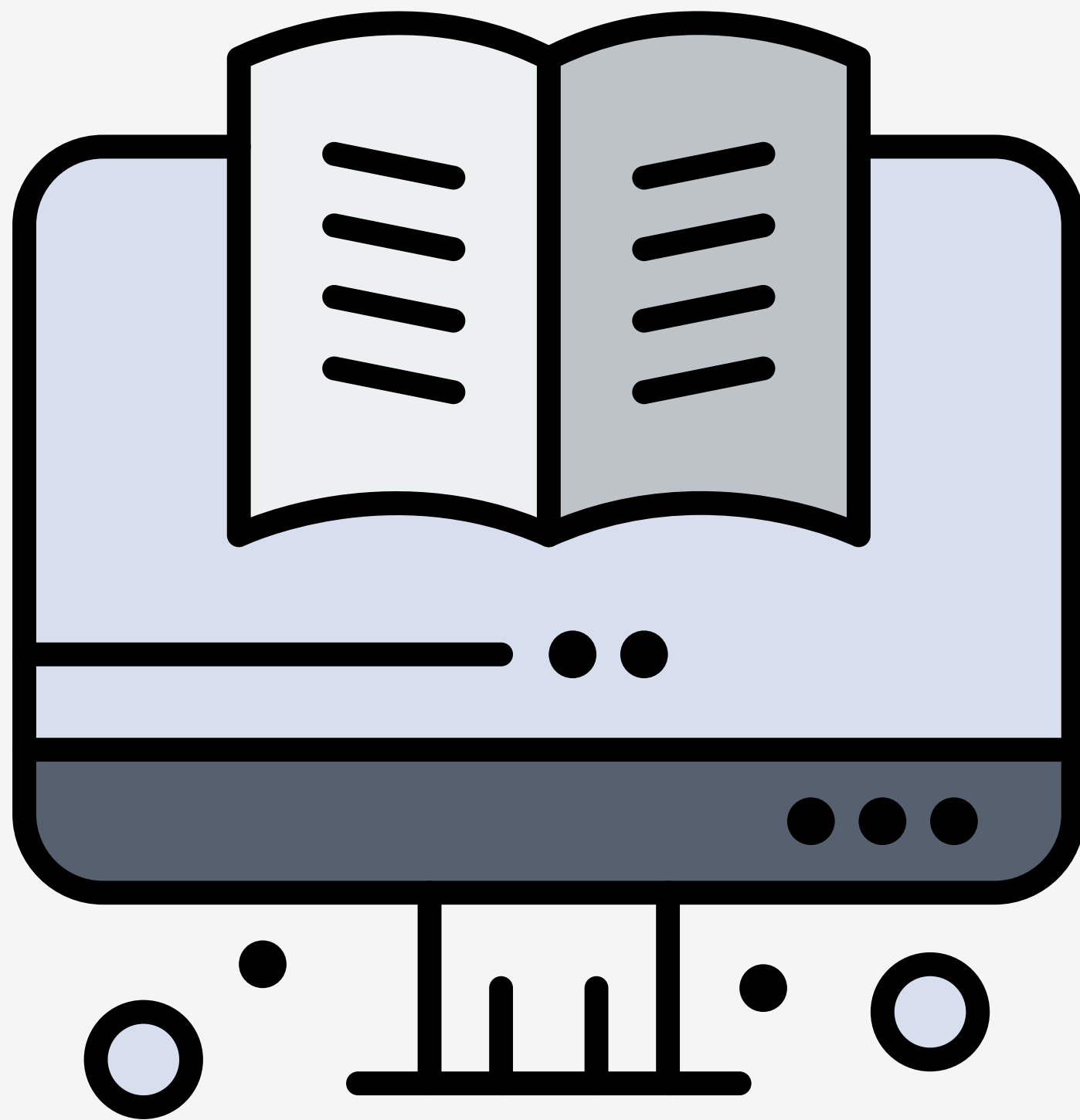
SASS

SASS



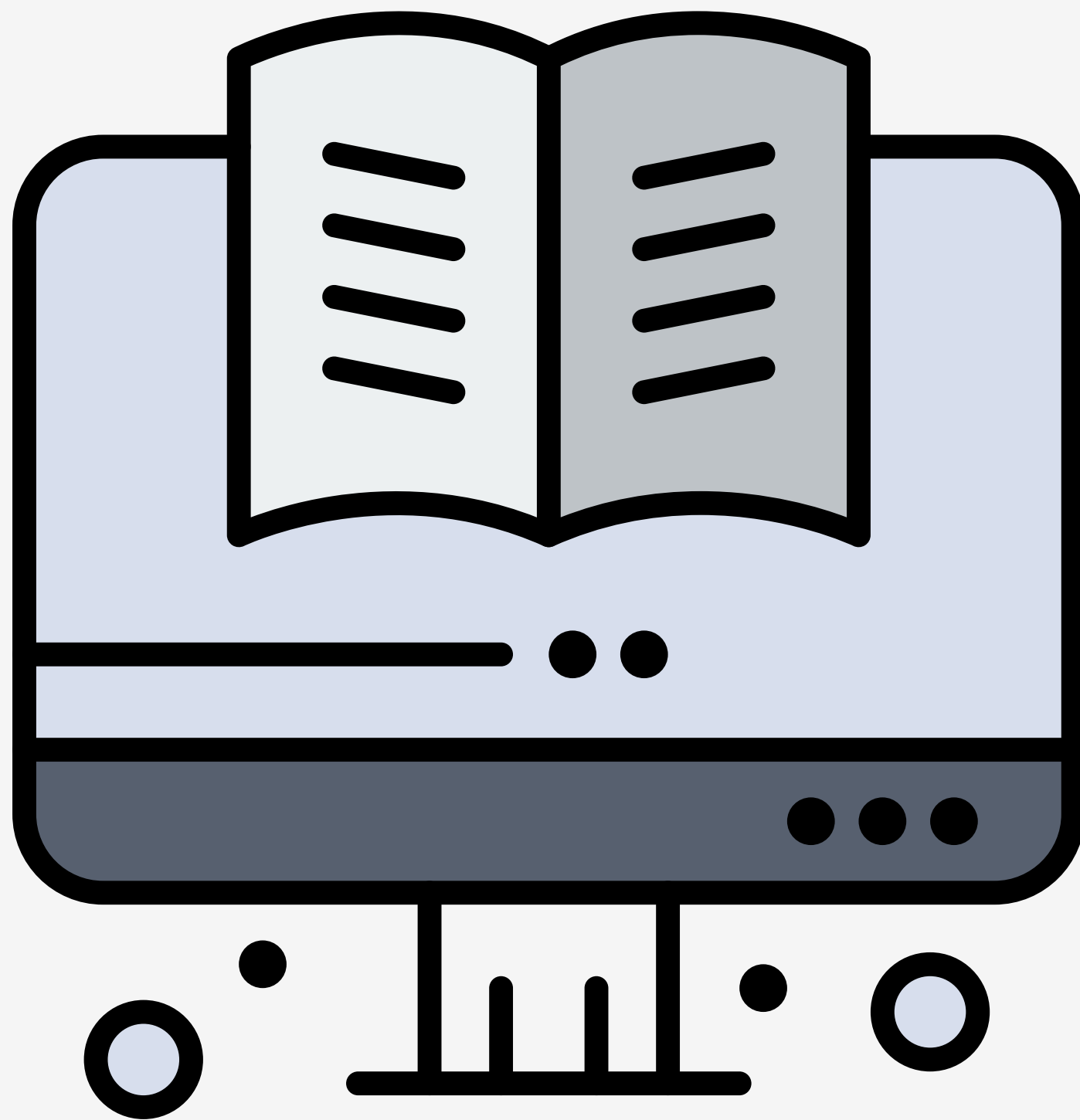
- Syntactically Awesome Stylesheets
- CSS Preprocess
- Sass introduced new concepts to CSS:
 - Variables
 - Nesting
 - Conditional Statements
 - Loops
 - Functions
 - Mixins

SASS FILE EXTENSIONS



- Files must be saved with a Sass file extension
- The `.SCSS` extension uses a syntax that is a superset of CSS
- The `.sass` extension uses the original indented syntax

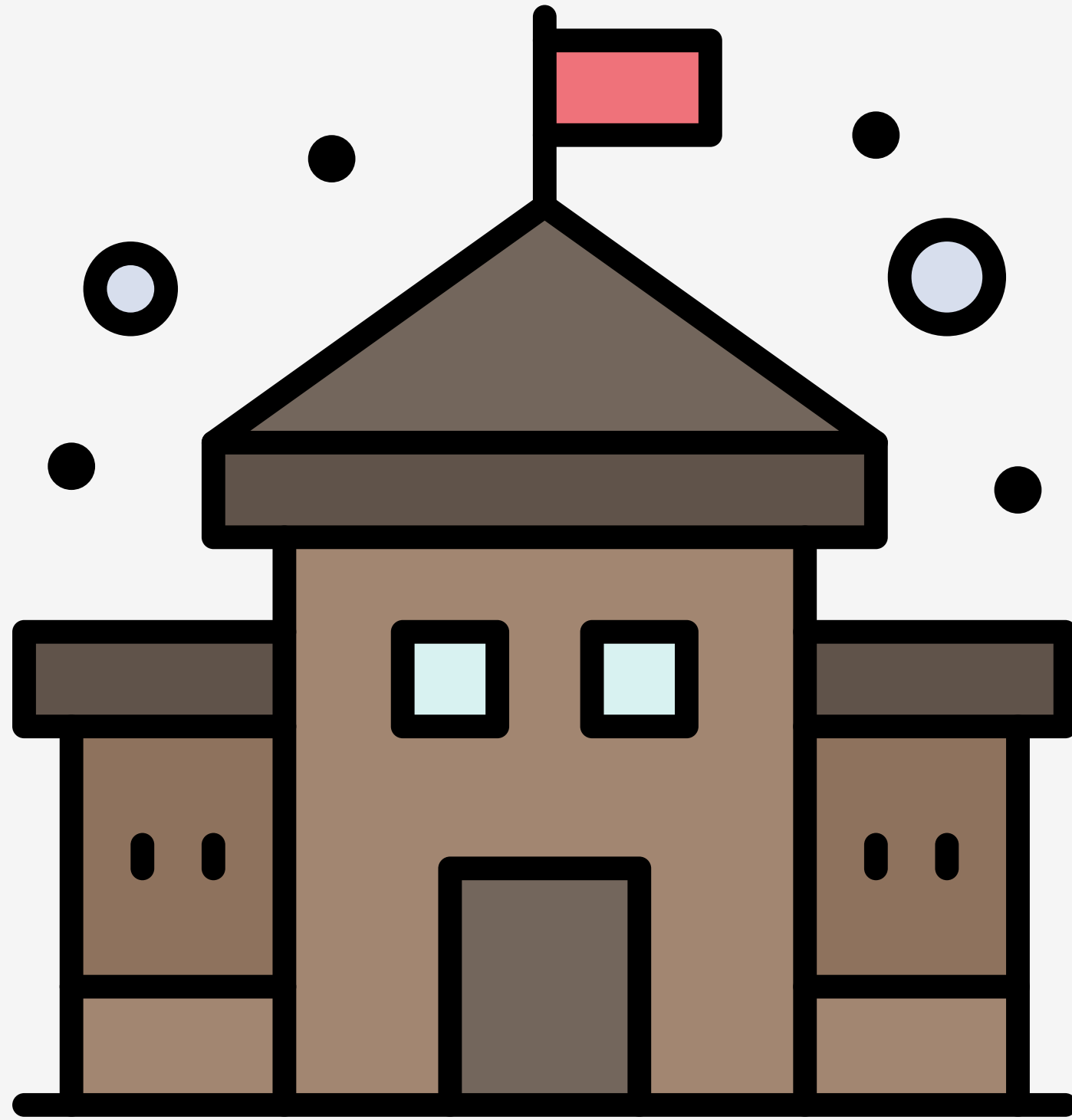
SASS COMPILING



- Sass must be compiled into CSS
- Three implementation
 - Dart Sass
 - LibSass
 - Ruby Sass
- Live Sass Compiler Extension

SASS VARIABLES

SASS VARIABLES



- Sass variables are used hold values
- Sass variable names begin with a \$
- A colon (:) separate the name from the value
- Sass Variable can be used anywhere in the code

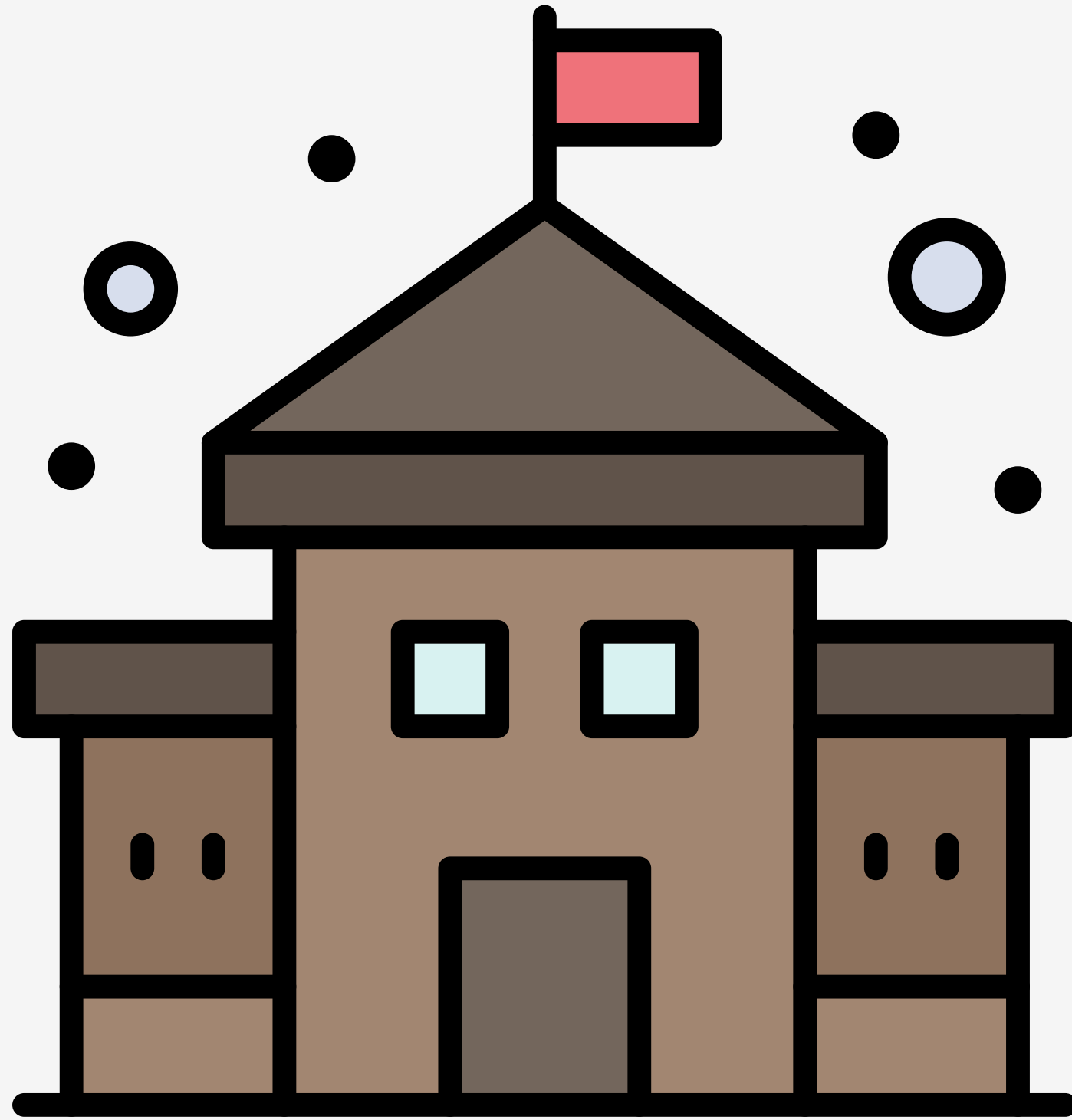
SASS VARIABLES

```
/* Sass */
$offwhite: #EEE8D6;
$darkblue: #022933;

body {
  color: $offwhite;
  background-color: $darkblue;
}
```

```
/* CSS */
body {
  color: #EEE8D6;
  background-color: #022933;
}
```

SASS VARIABLES **VS** CSS VARIABLES



- **Sass variables** are read by Sass compiler
- **CSS variables** are read by the browser
- **Sass variable** can only hold one value at a time
- **CSS variables** can hold different values for different elements
- If a **Sass variable** value is changed, only future uses are affected
- If a **CSS variable** is changed, all uses are affected

NESTING IN SASS

NESTING IN SASS



- Nesting is a way for creating a visual hierarchy in CSS
- A child style is placed inside of the parent
- **WARNING!** Do not overuse nesting.

NESTING

```
/* Sass */
ul {
  list-style: none;

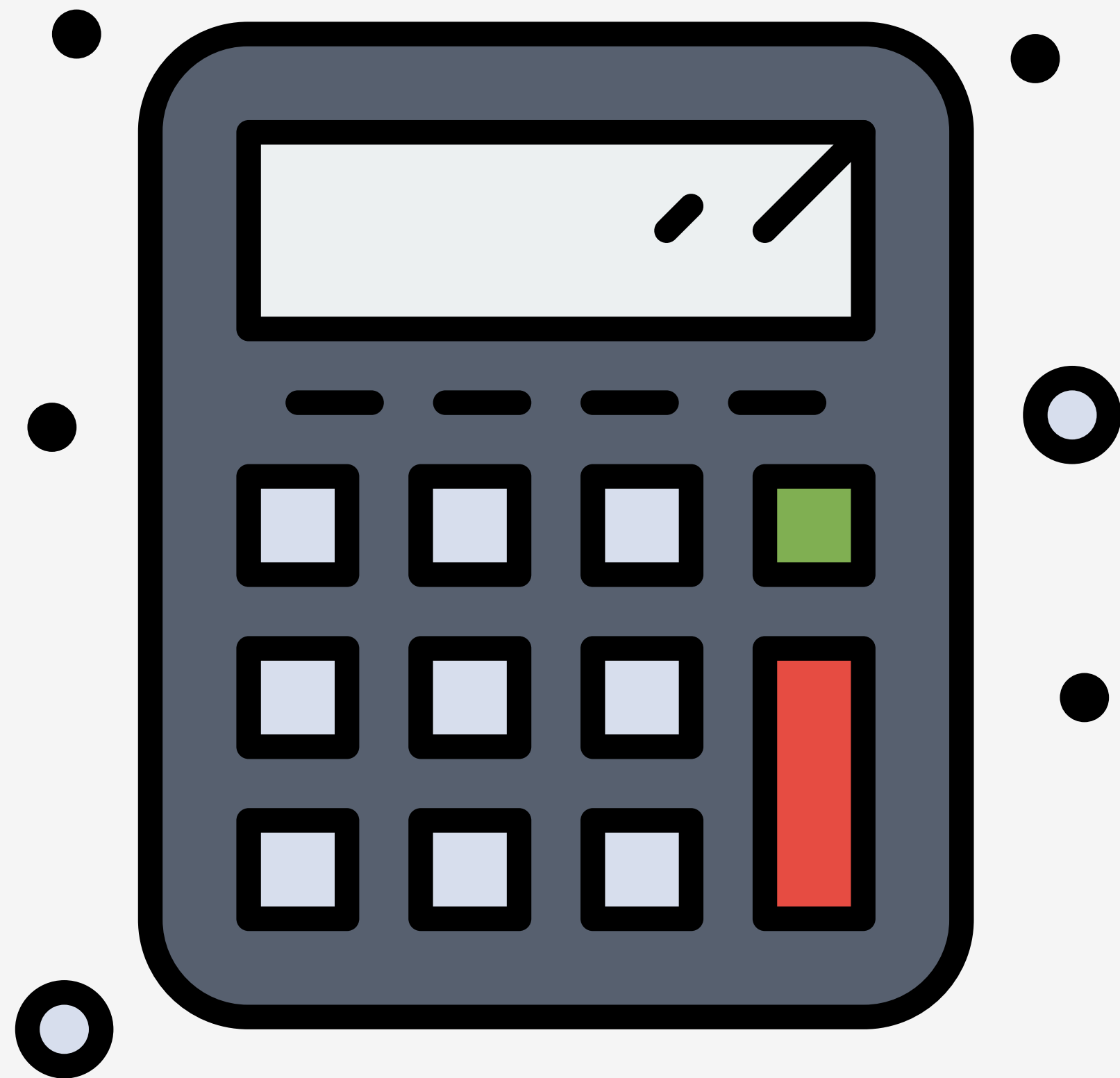
  li {
    margin-bottom: 20px;
    border-top: 1px dotted red;
    font-size: 2.0rem;
  }
}
```

```
/* CSS */
ul {
  list-style: none;
}

ul li {
  margin-bottom: 20px;
  border-top: 1px dotted red;
  font-size: 2.0rem;
}
```

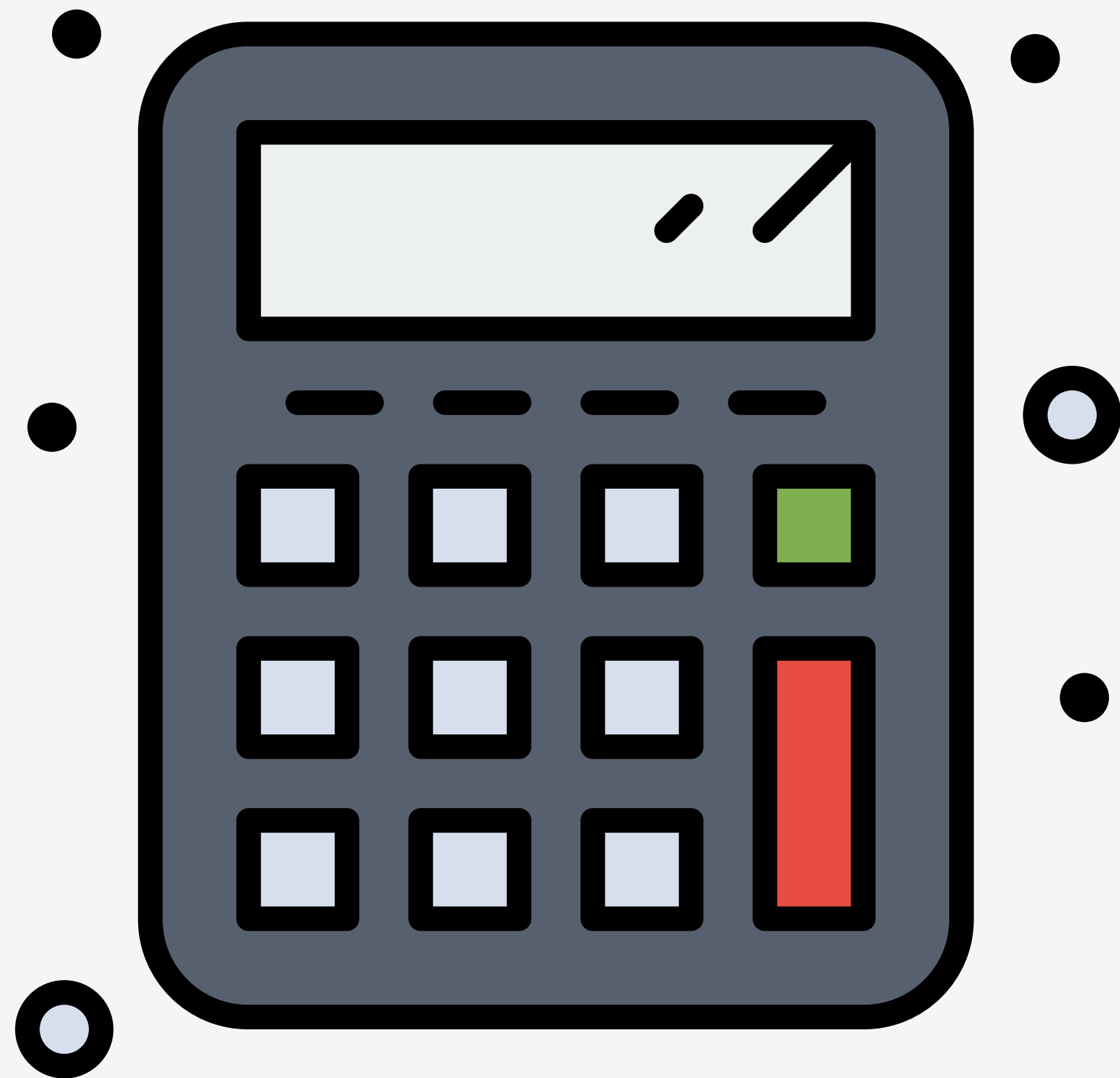
SASS PARTIALS

SASS PARTIALS



- **Partials** are Sass files that contain snippets of CSS and / or Sass
- **Partials** are included into other Sass files
- **Partials** are not directly compiled by Sass compiler
- **Partials** are used to modularized CSS
- **Partials** are created by adding an underscore () at the beginning of the filename.

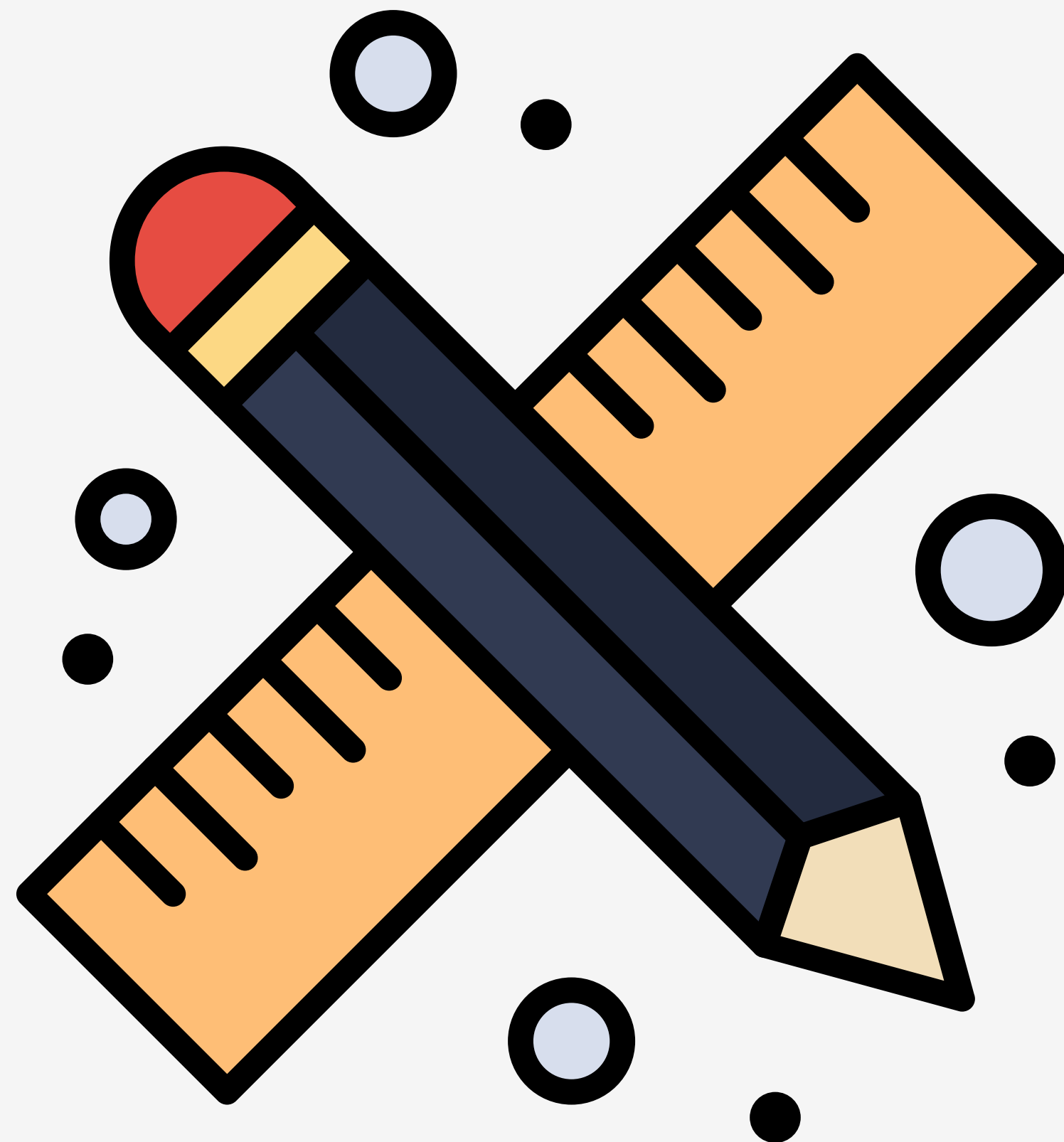
INCLUDING PARTIALS



- **Partials** are adding to another Sass file using the `@import` rule
- The main file will have access to any variables, mixins, or functions in the included **partials**
- **NOTE:** The `@use` rule is **NOW** the preferred way to include partials, but is only compatible with the Dart Sass

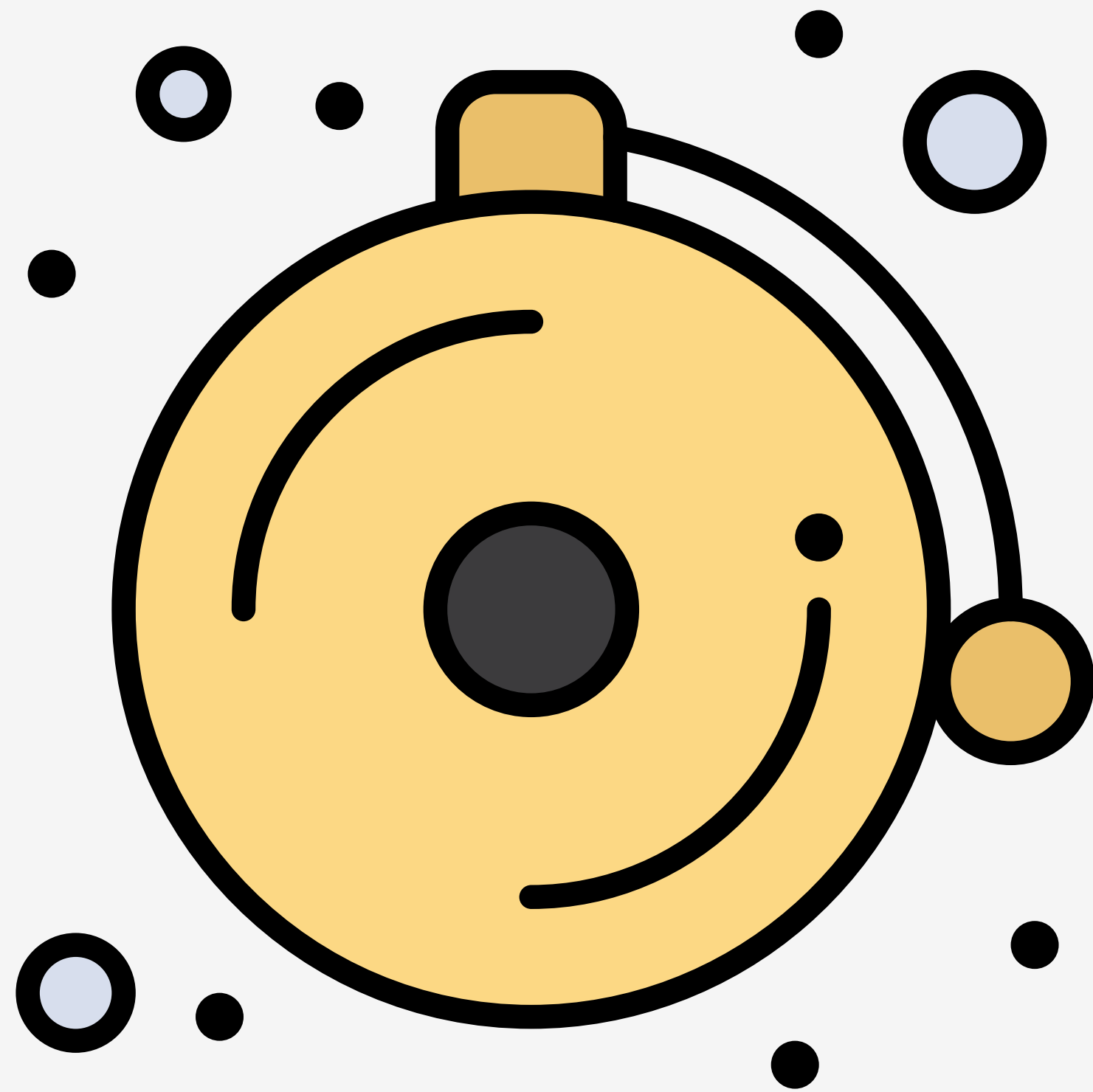
HANDS-ON

WEB PORTFOLIO - FINAL



- Create *YOUR* web portfolio
- Base on the designs created in *UX Design*
- Fully responsive
- Use of frameworks and libraries are allowed
- Push the completed site to GitHub Pages
- Submit the URL to the live site
- *DUE:* Thu. Apr. 16 @ 11:59 PM

NEXT TIME...



- Sass Mixins
- Sass @Extend
- Sass Parent Selector
- **Participation:** Mixin Mixup