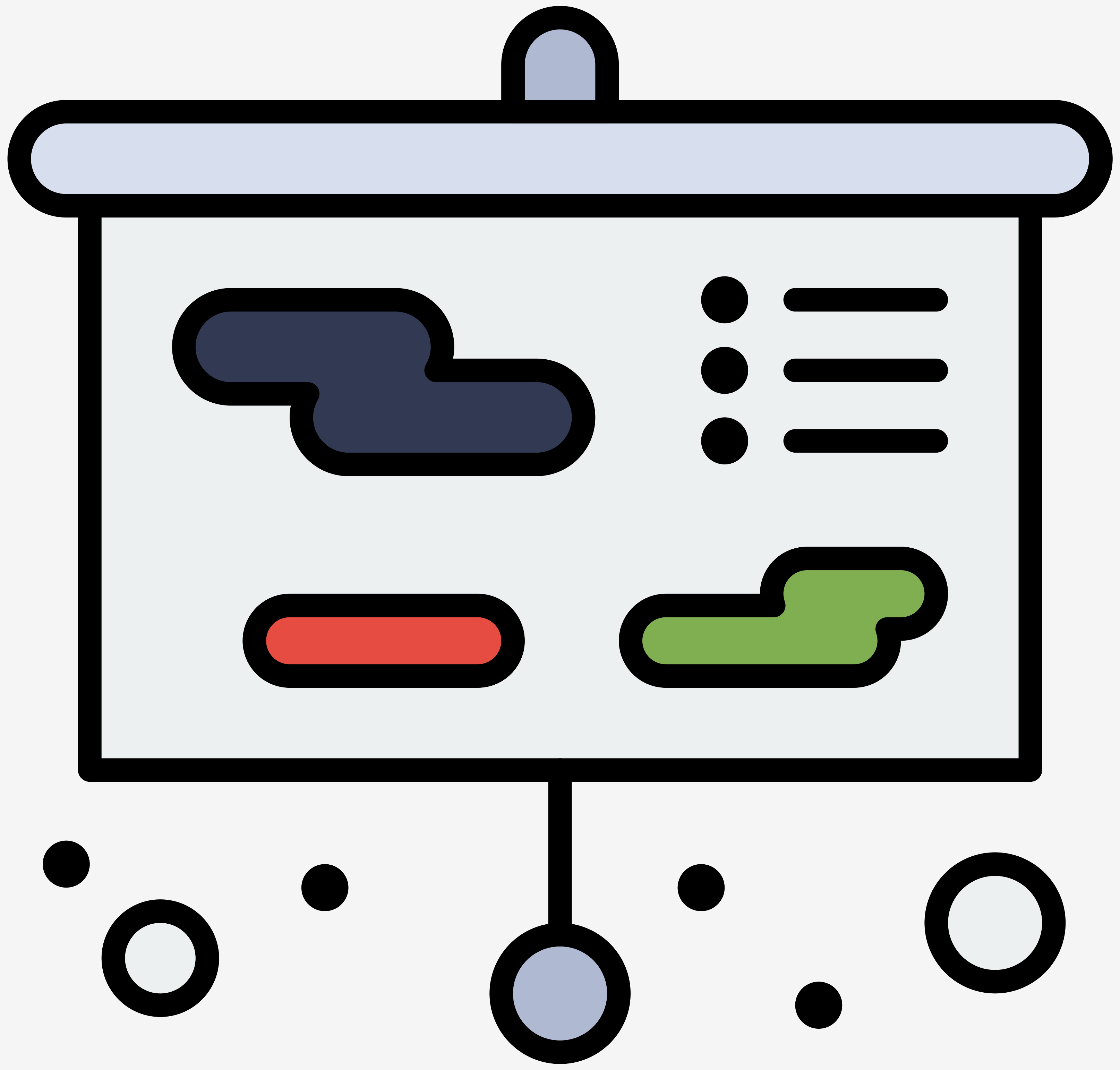
INTRODUCTION TO JAVASCRIPT

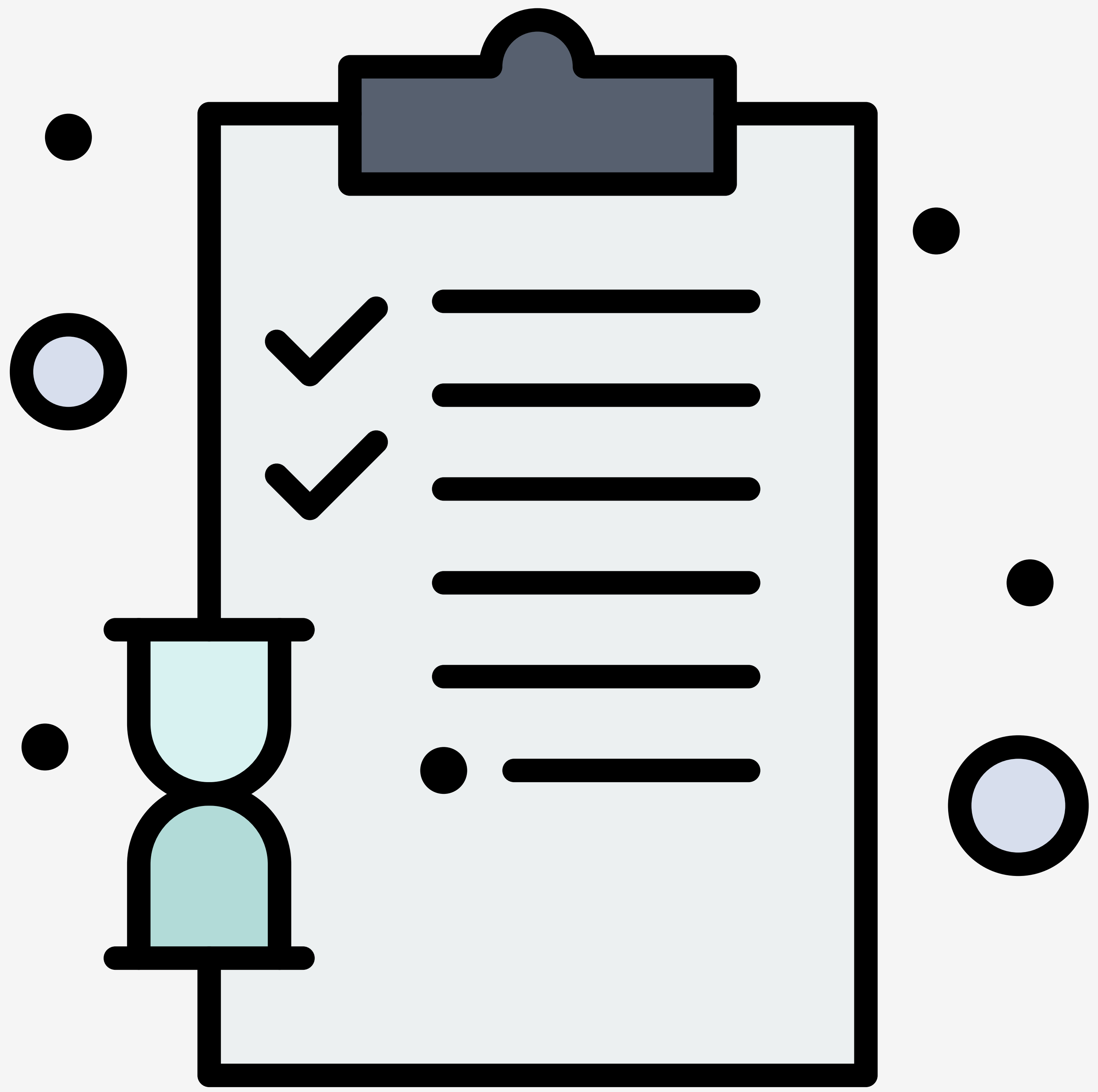
Lecture 18

TODAY'S TOPICS



- Introduction to Sass
- Exercise: Sassy Shapes

ANNOUNCEMENTS

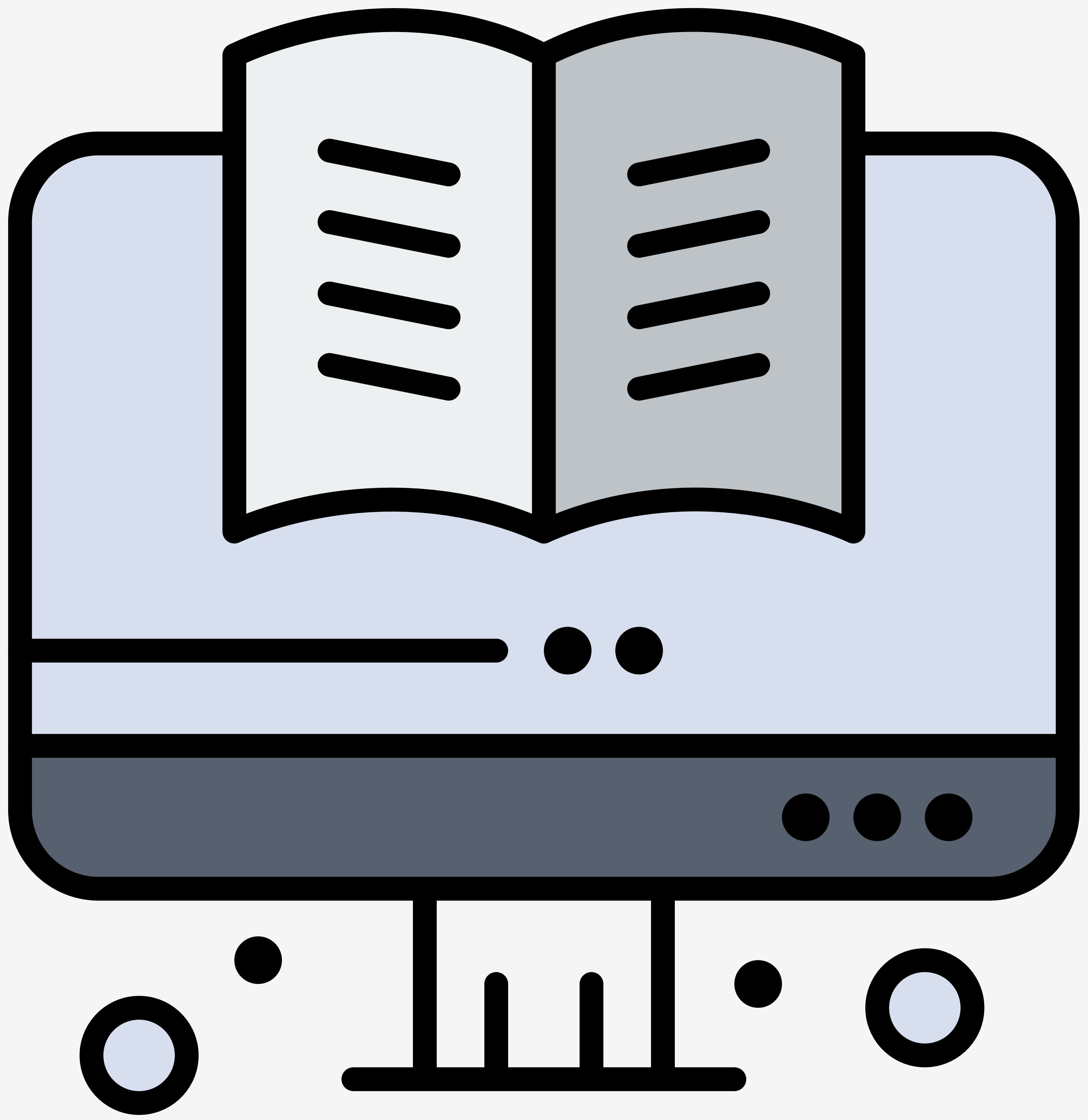


- Sign-in Sheet

QUESTIONS?

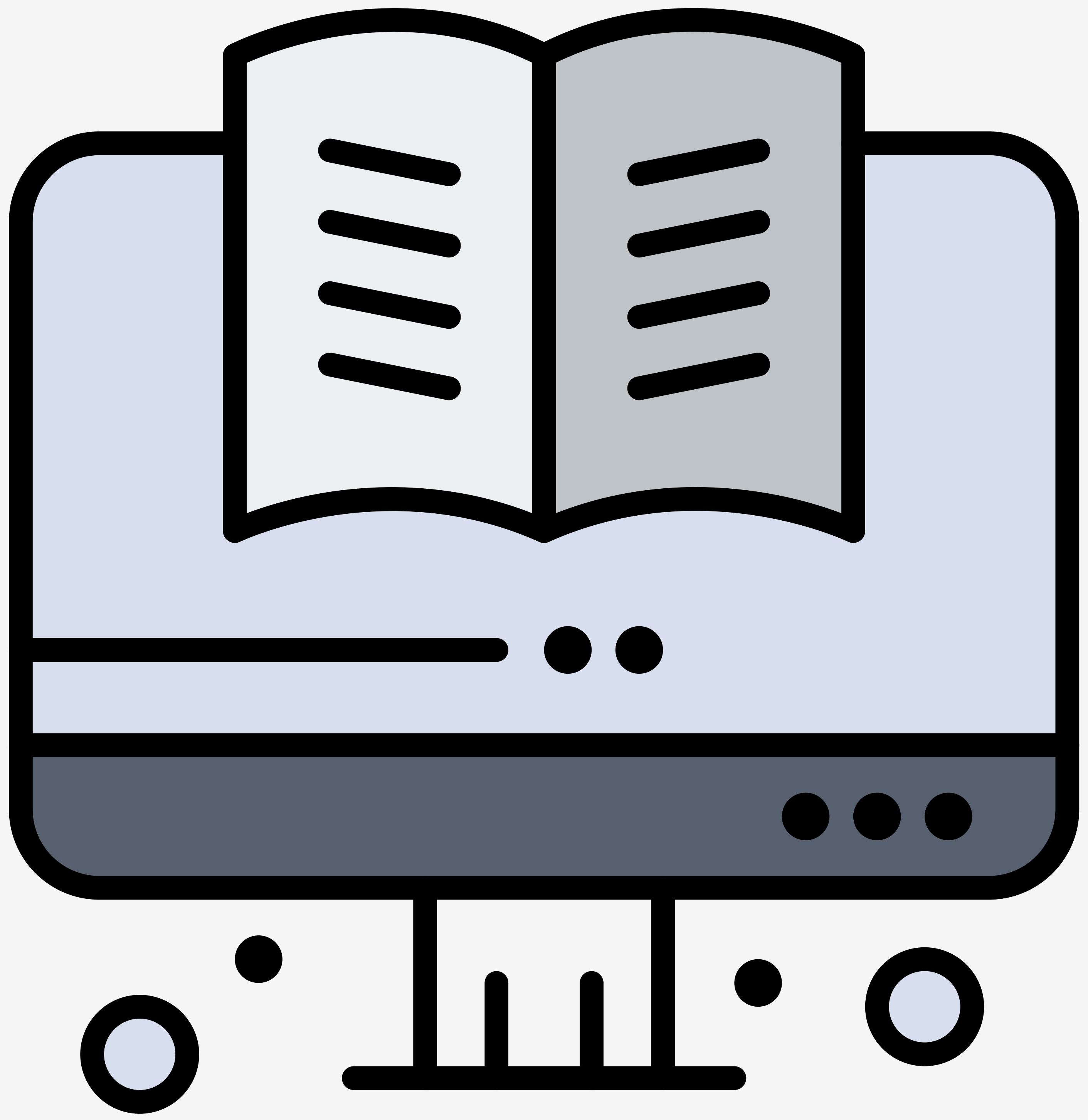
SASS

SASS



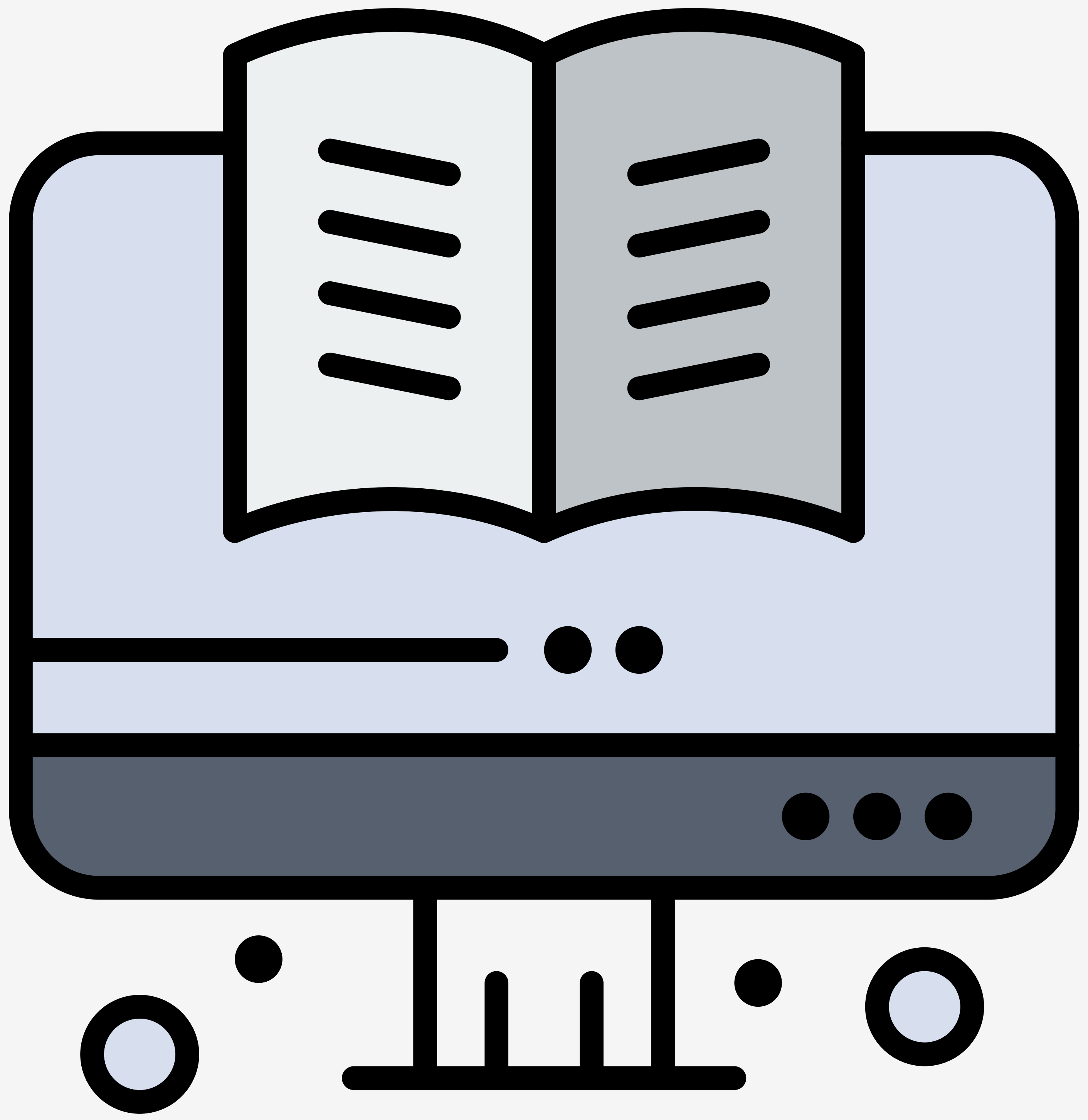
- Syntactically Awesome Stylesheets
- CSS Preprocess
- Sass introduced new concepts to CSS:
  - Variables
  - Nesting
  - Conditional Statements
  - Loops
  - Functions
  - Mixins

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

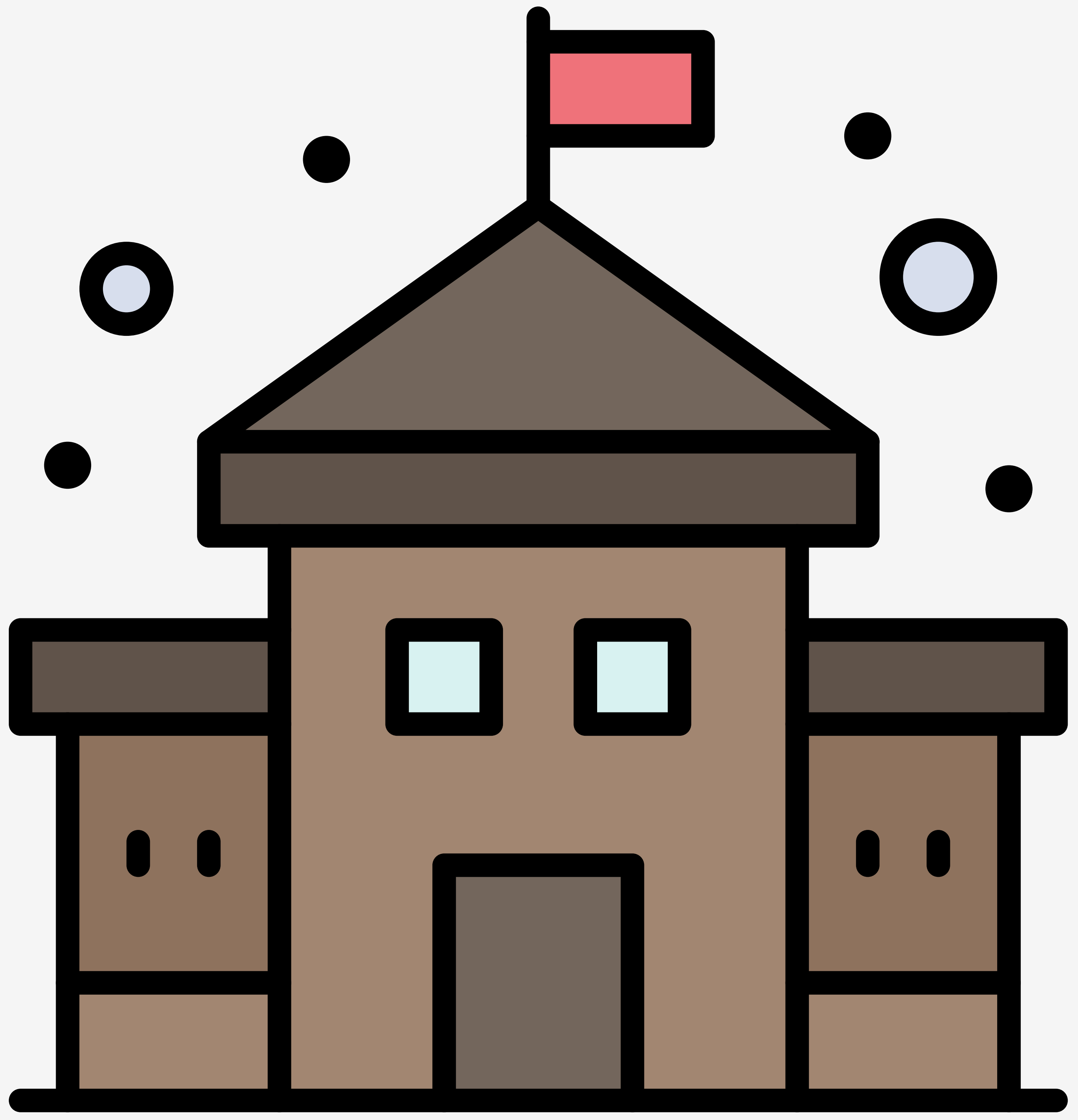
SASS COMPILING



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

SASS VARIABLES

SASS VARIABLES



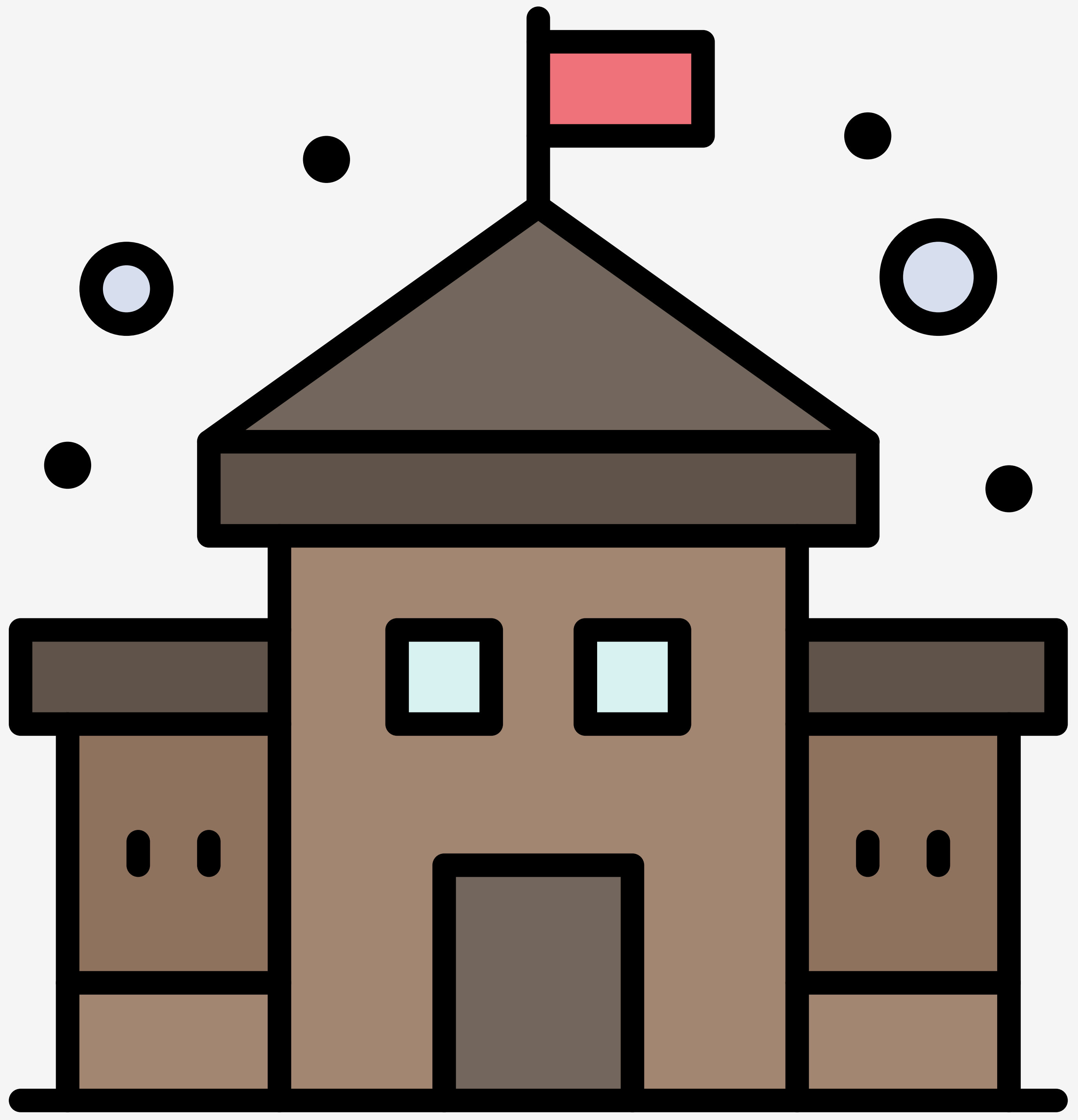
- Sass variables are used hold values
- Sass variable names begin with a $
- A colon (:) separate the name from the value
- Sass Variable can be used anywhere in the code

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

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

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

CSS VARIABLES



- CSS variables are also used hold values
- CSS variables are set using the custom property notation
- CSS variables are access using the `var()` function

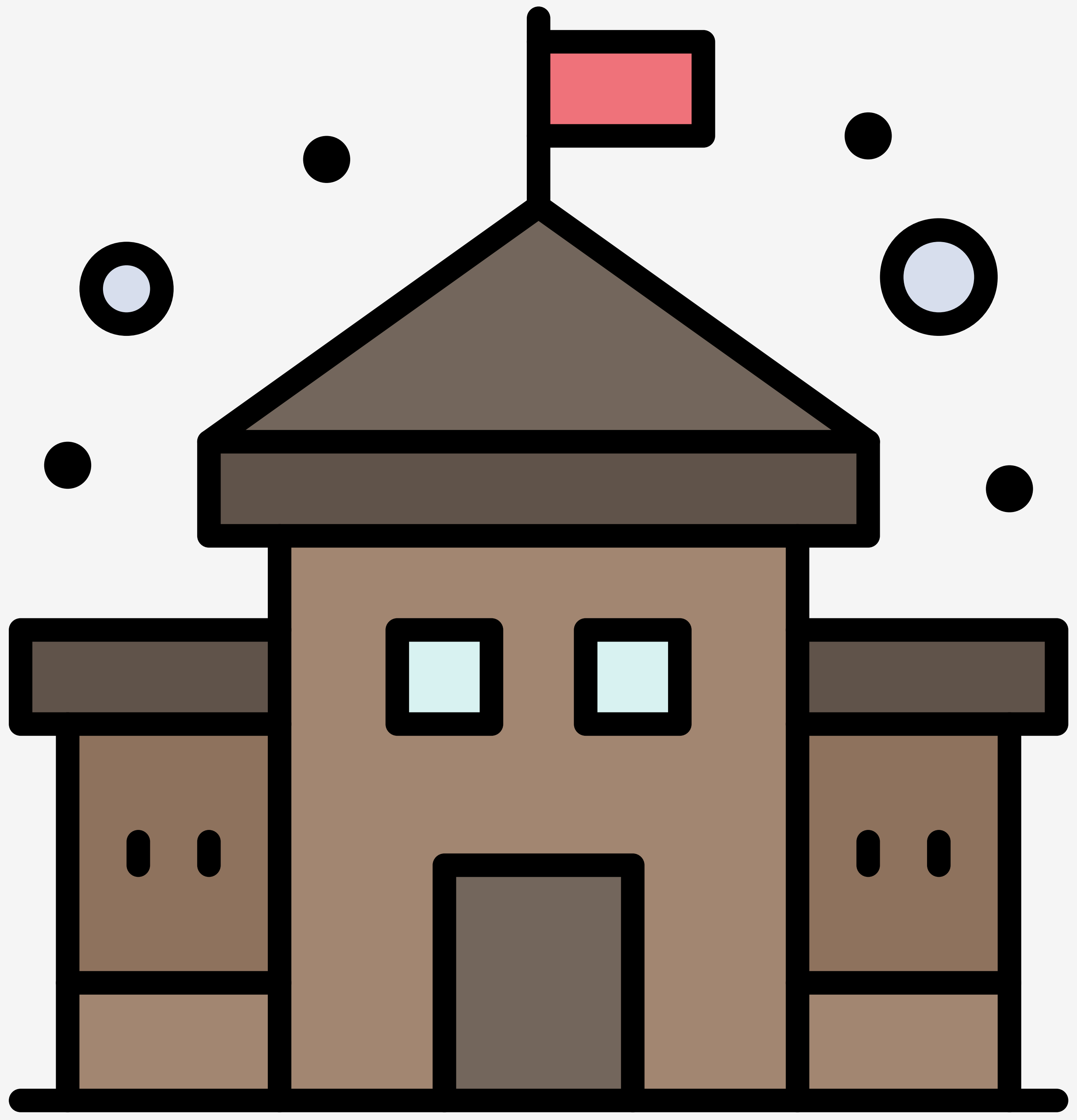
```scss
/* Sass */
$offwhite: #EEE8D6;
$darkblue: #022933;

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

```css
/* CSS */
:root {
  --offwhite: #EEE8D6;
  --darkblue: #022933;
}

body {
  color: var(--offwhite);
  background-color: var(--darkblue);
}
```

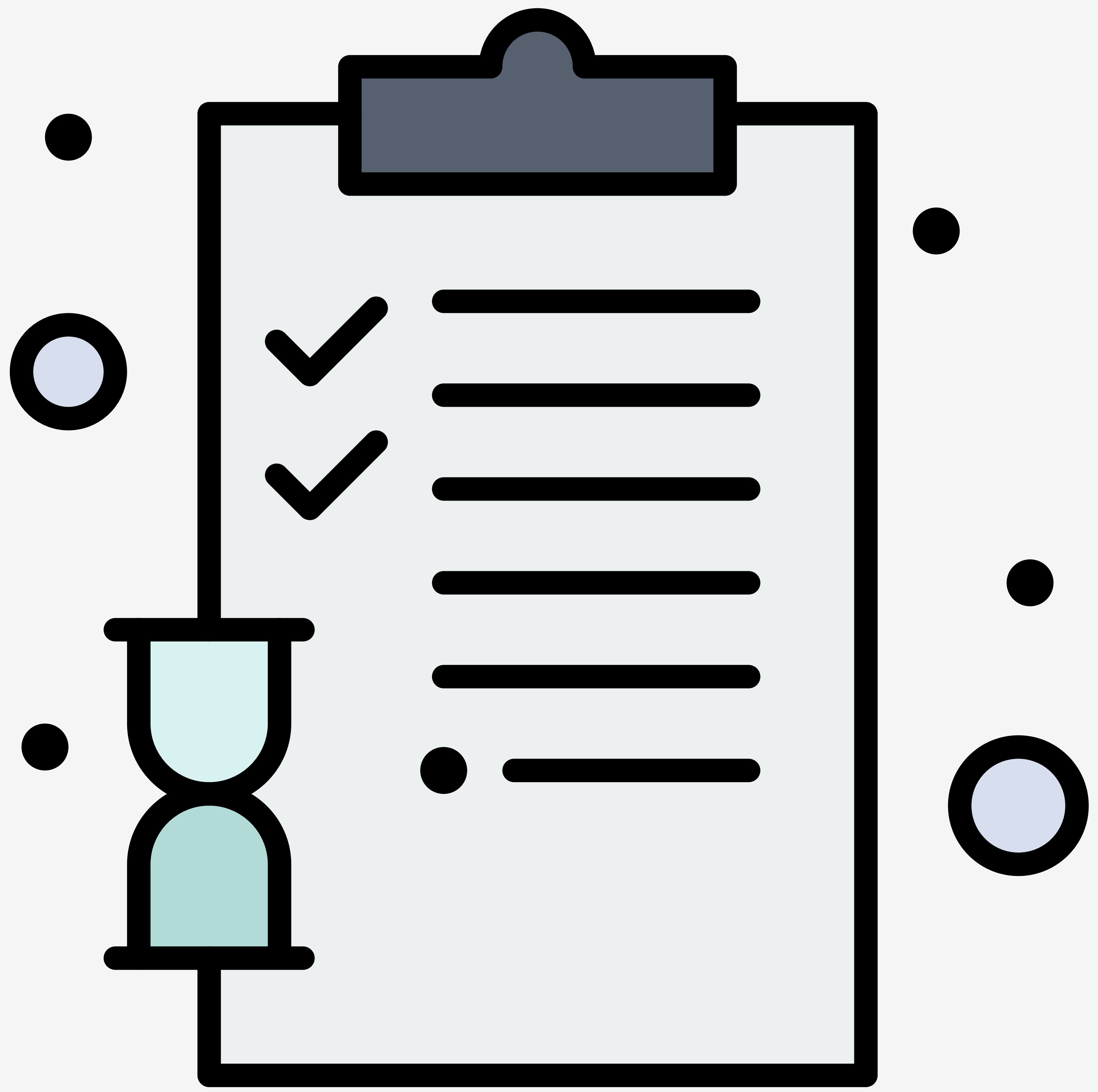
SASS VARIABLES VS CSS VARIABLES



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

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

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

  p {
    margin: 0;
    font-size: 1.5rem;
  }
}
```

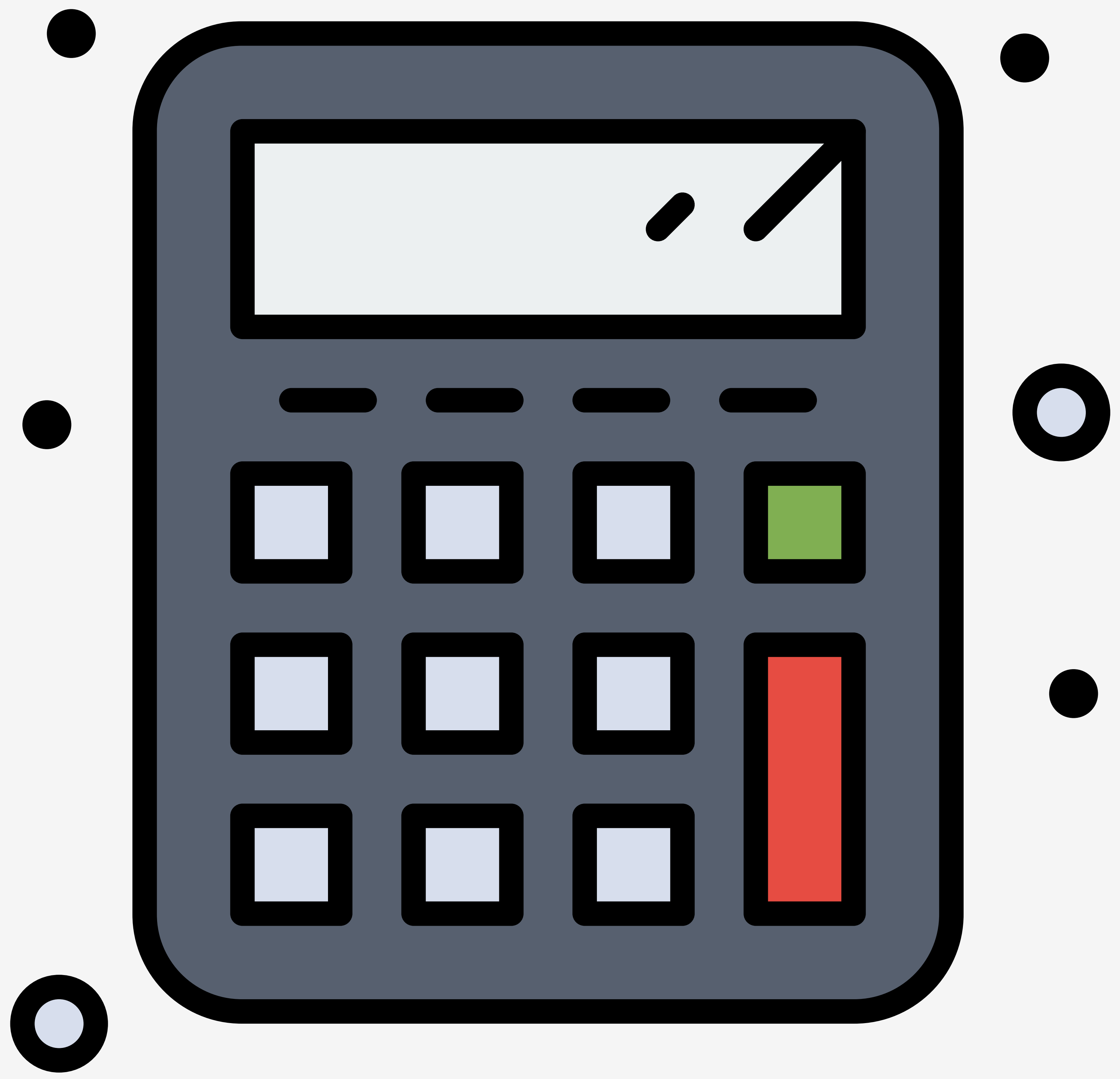
```css
/* CSS */
ul {
  list-style: none;
}

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

ul p {
  margin: 0;
  font-size: 1.5rem;
}
```

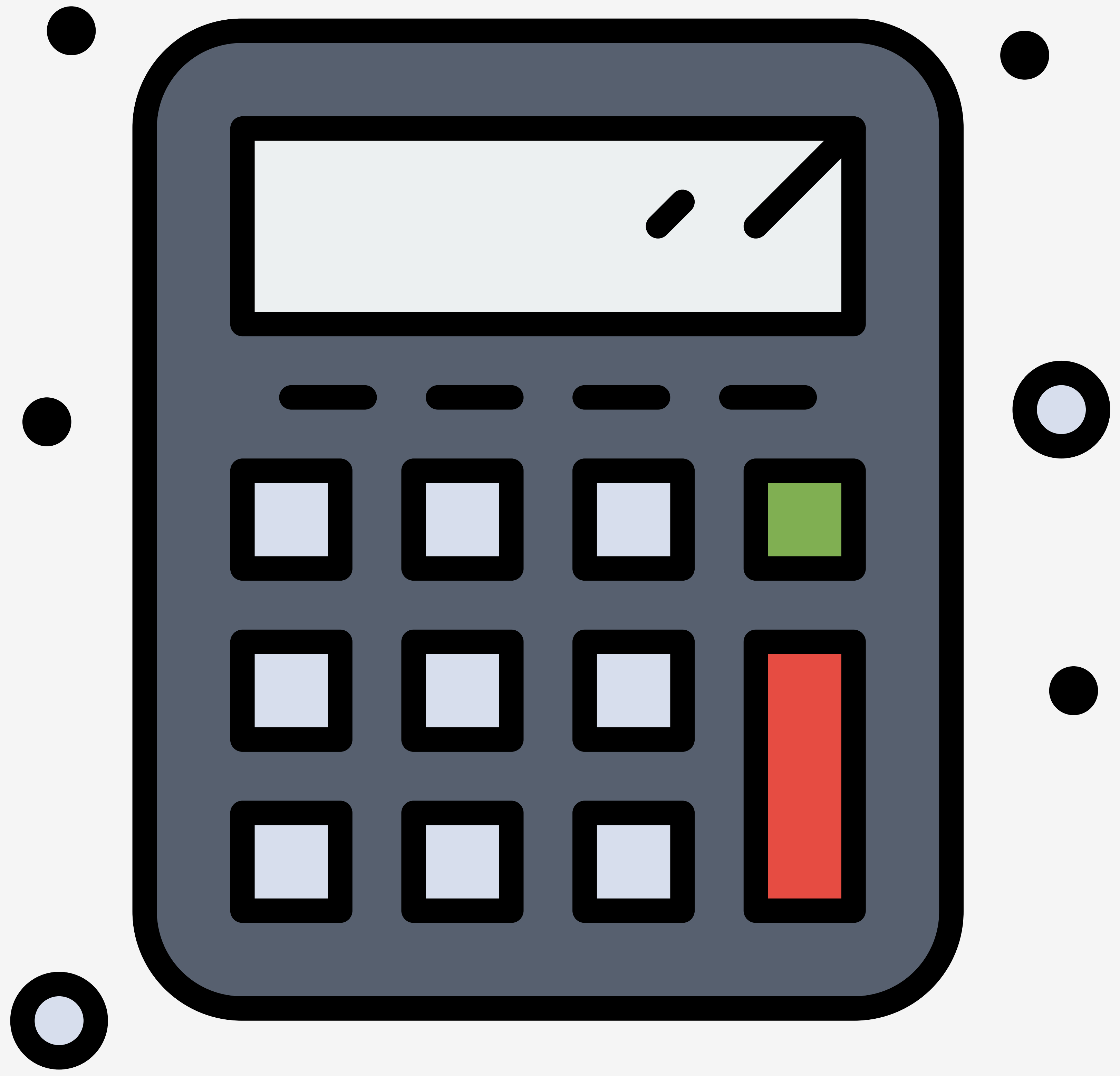
SASS PARTIALS

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.

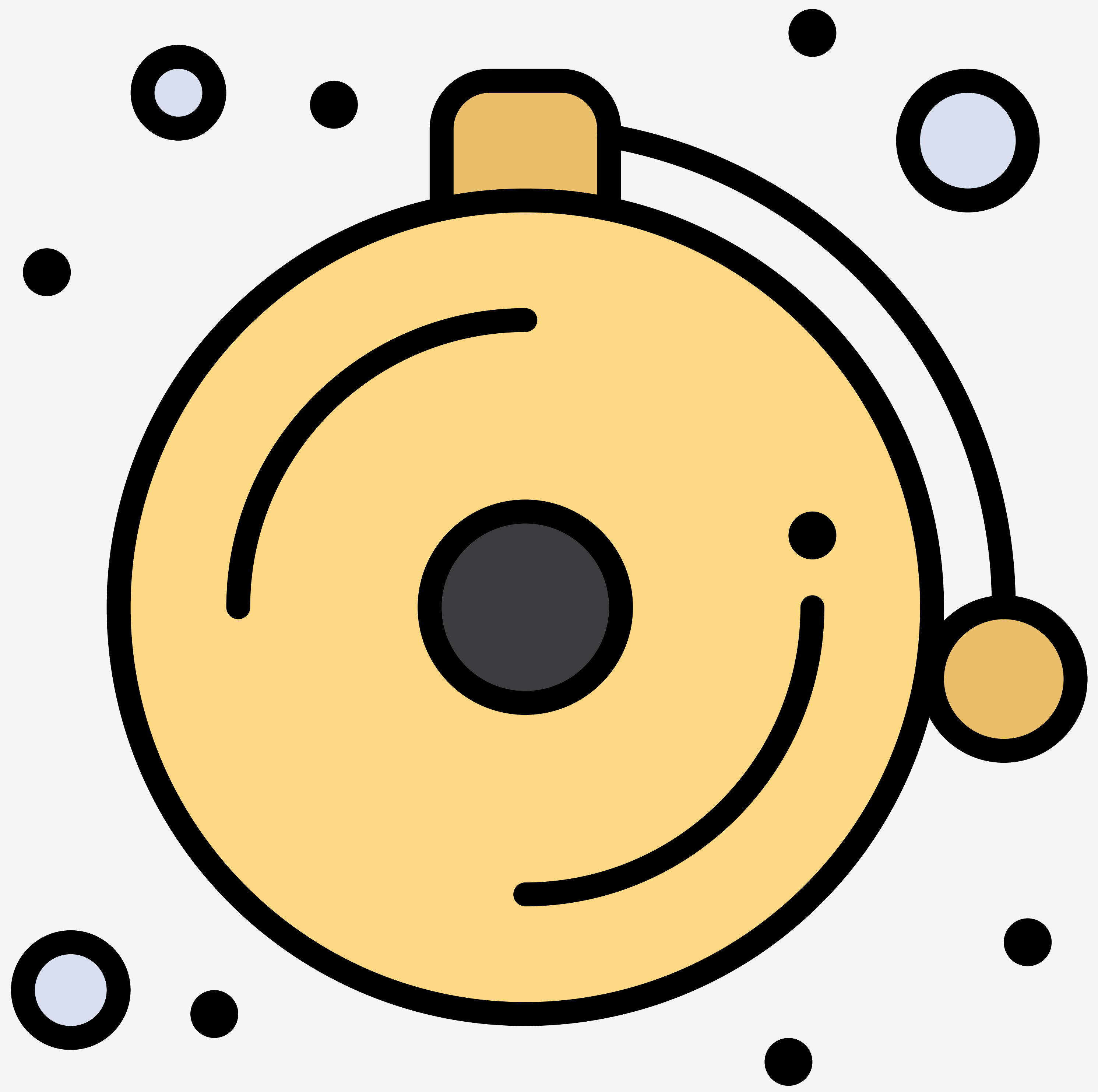
INCLUDING PARTIALS



- 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 will soon be the preferred way to include partials

HANDS-ON

NEXT TIME...



- Sass Mixins
- Sass @extend
- Sass Parent Selector
- Review: Boxes
- Exercise: Sassy Shapes