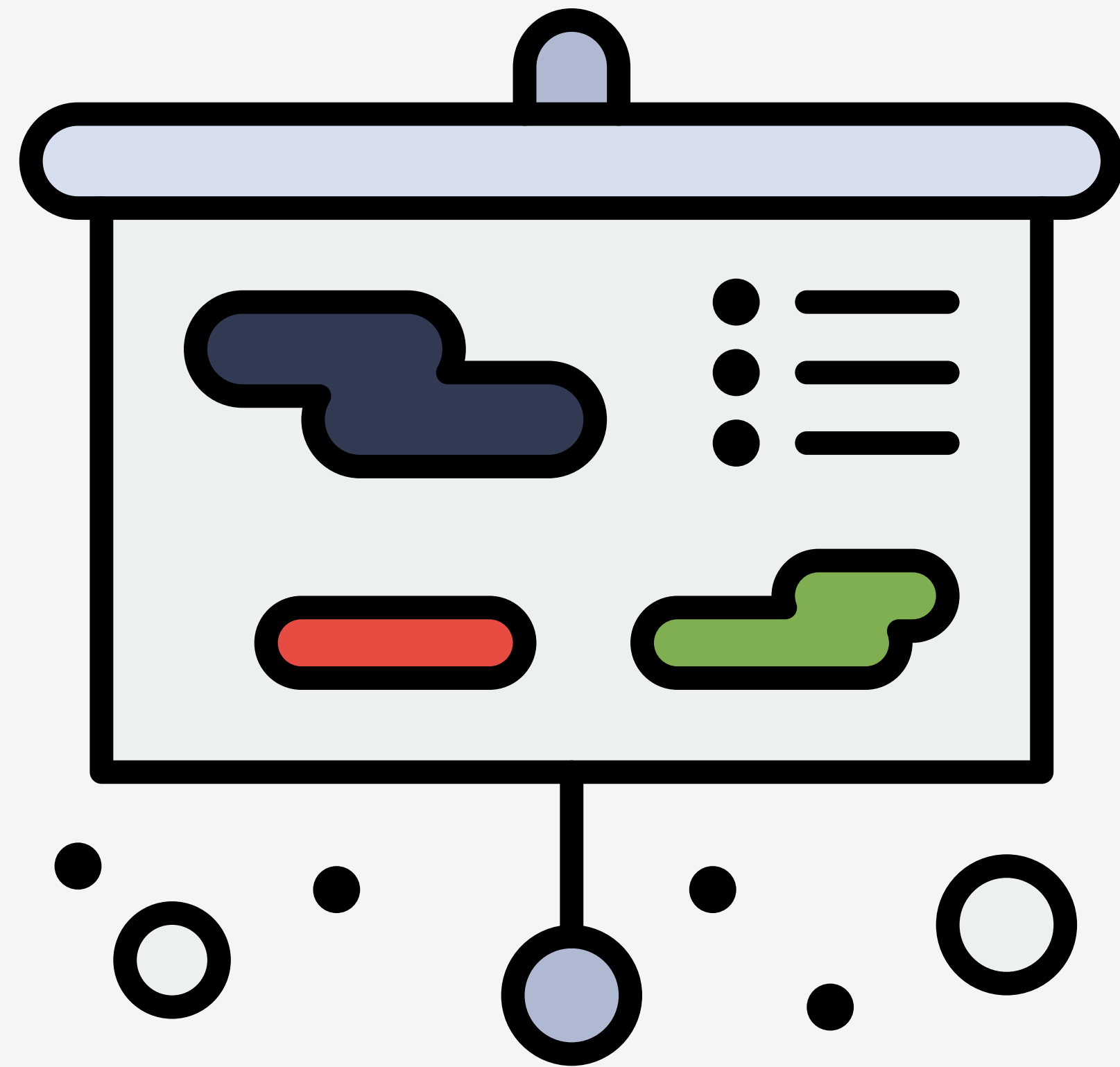

INTRODUCTION TO JAVASCRIPT

Lecture 20

TODAY'S TOPICS



- SassScript
- Participation: Sassy Cats

ANNOUNCEMENTS

- Course Surveys



QUESTIONS?

SASS LISTS

SASS LISTS



Going Away

- Lists are a sequence of values
- Elements can be separated by commas or spaces
- Parentheses can be used, but not required
- The `nth()` function is used to access a single element in a list by their index
- The index `1` will return the first element, the index `-1` will return the last element

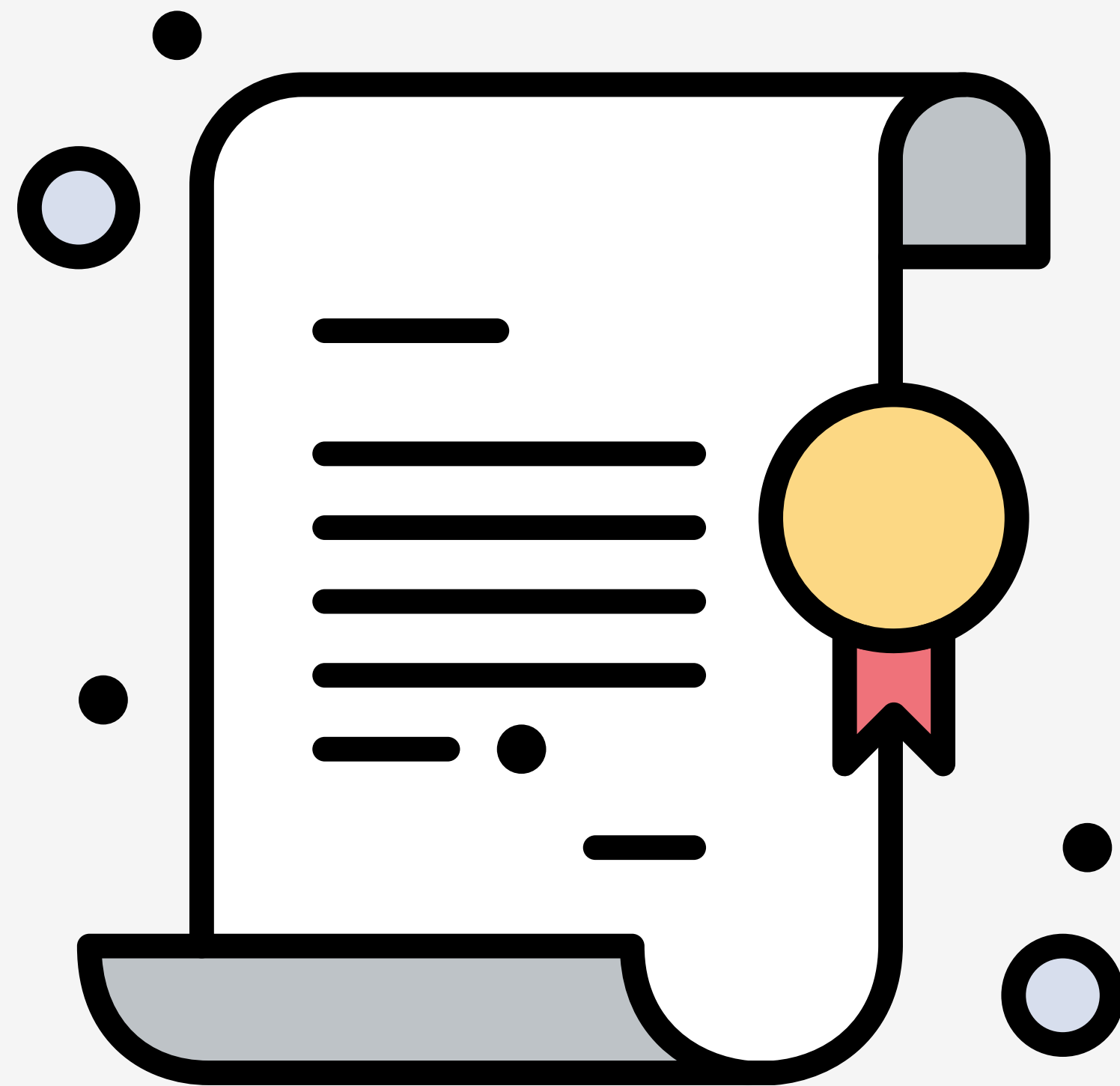
```
/* Sass */
$font-family: Helvetica, sans-serif;
$border-radius: 0px 20px 0px 20px;
$colors: (red, green, blue);

box {
  width: 100px;
  height: 100px;
  border: 1px solid #333;
  border-radius: $border-radius;
  background-color: nth($colors, 2);
  font-family: $font-family;
}
```

```
/* CSS */
box {
  width: 100px;
  height: 100px;
  border: 1px solid #333;
  border-radius: 0px 20px 0px 20px;
  background-color: green;
  font-family: Helvetica, sans-serif;
}
```

SASS CONTENT BLOCKS

SASS CONTENT BLOCK



- A **content block** is a block of style that is passed to a mixin
- A **content block** can be included in a mixin using **@content** rule

```
/* Sass */
@mixin hover {
  &:not([disabled]):hover {
    @content;
  }
}

.button {
  background-color: #007bff;

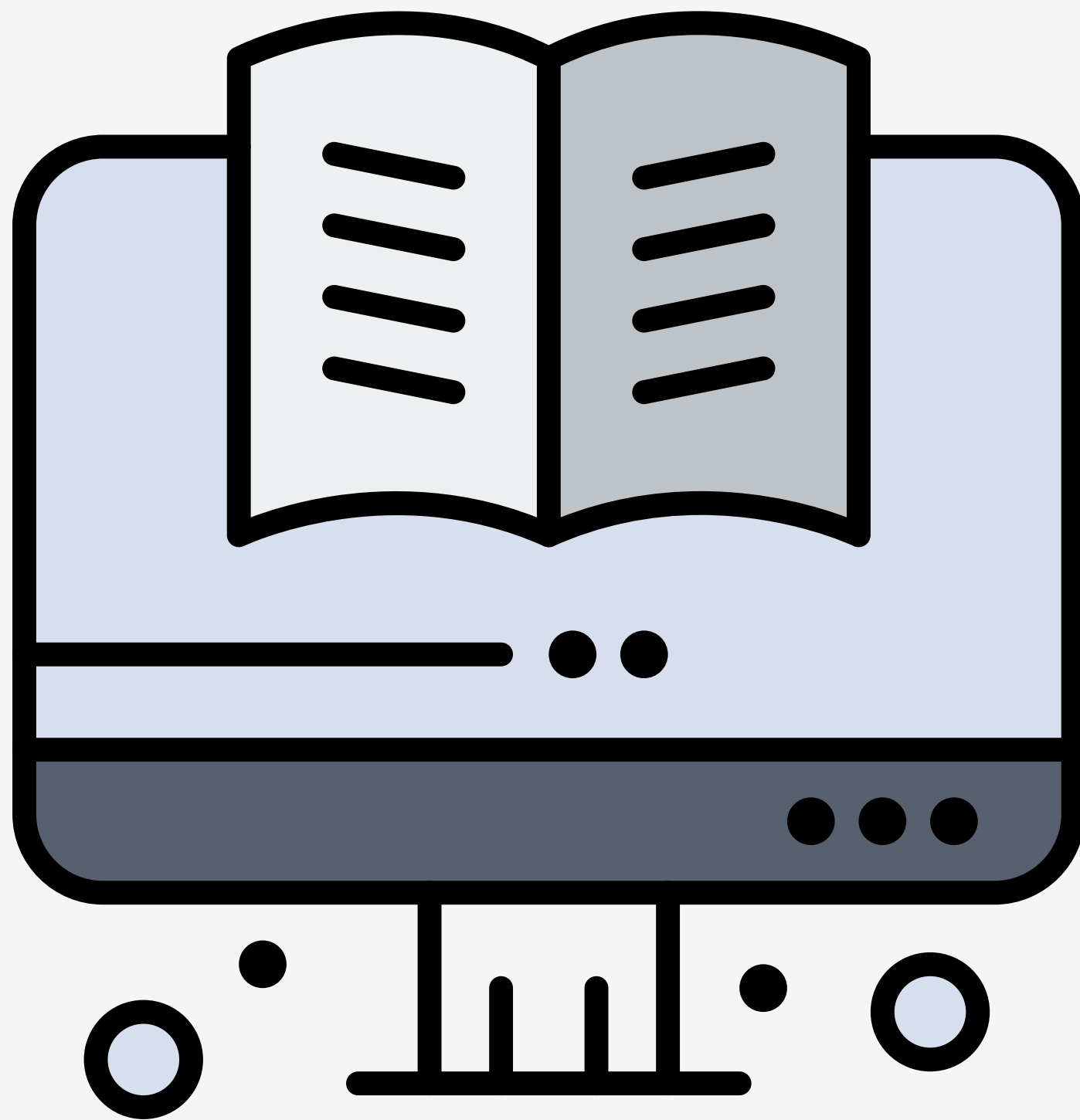
  @include hover {
    background-color: #0069d9;
    cursor: pointer;
  }
}
```

```
/* CSS */
.button {
  background-color: #007bff;
}

.button:not([disabled]):hover {
  background-color: #0069d9;
  cursor: pointer;
}
```

SASS @IF

SASS @IF



- The `@if` rule can be used to conditional evaluate blocks
- The `@if` rule expression will return `true` or `false`
- If the expression is `true` the block will be evaluated
- The `@else` rule can be added and will be evaluated if the `@if` expression is false
- The `@else if` rule can be used when more than one condition needs to be tested

```
/* Sass */
@mixin breakpoint ($size) {
  @if $size=='medium' {
    @media screen and (min-width: 640px) {
      @content;
    }
  }

  @else if $size=='large' {
    @media screen and (min-width: 1024px) {
      @content;
    }
  }

  @else {
    @error "Unknown size #{ $size }."
  }
}
```

```
/* Sass */
main {
  display: grid;
  grid-gap: 5px;
  grid-template-columns: 1fr;

  @include breakpoint('medium') {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
  }

  @include breakpoint('large') {
    display: grid;
    grid-template-columns: 200px 1fr 1fr 200px;
  }
}
```

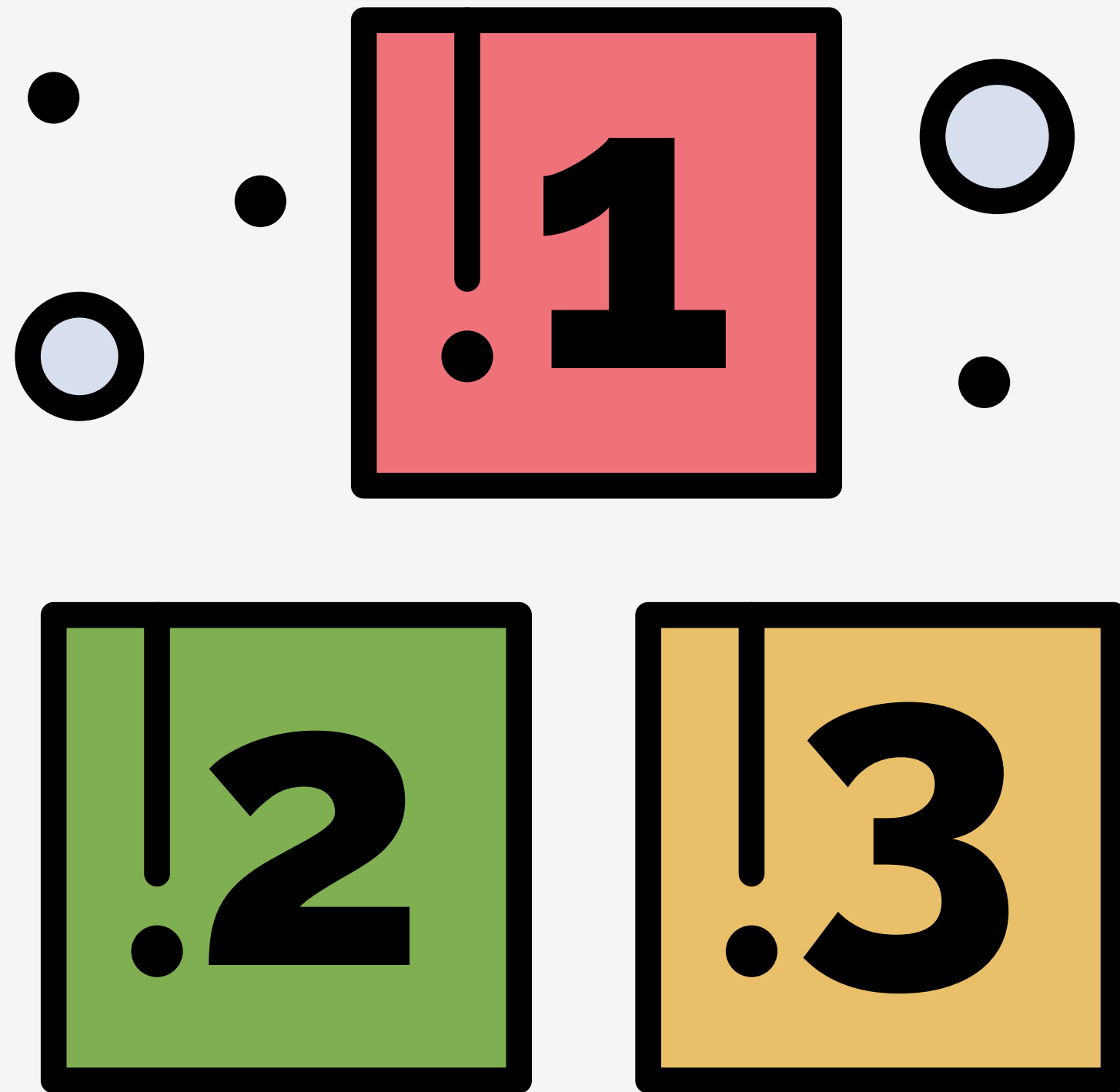
```
/* CSS */
main {
  display: grid;
  grid-gap: 5px;
  grid-template-columns: 1fr;
}

@media screen and (min-width: 640px) {
  main {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
  }
}

@media screen and (min-width: 1024px) {
  main {
    display: grid;
    grid-template-columns: 200px 1fr 1fr 200px;
  }
}
```

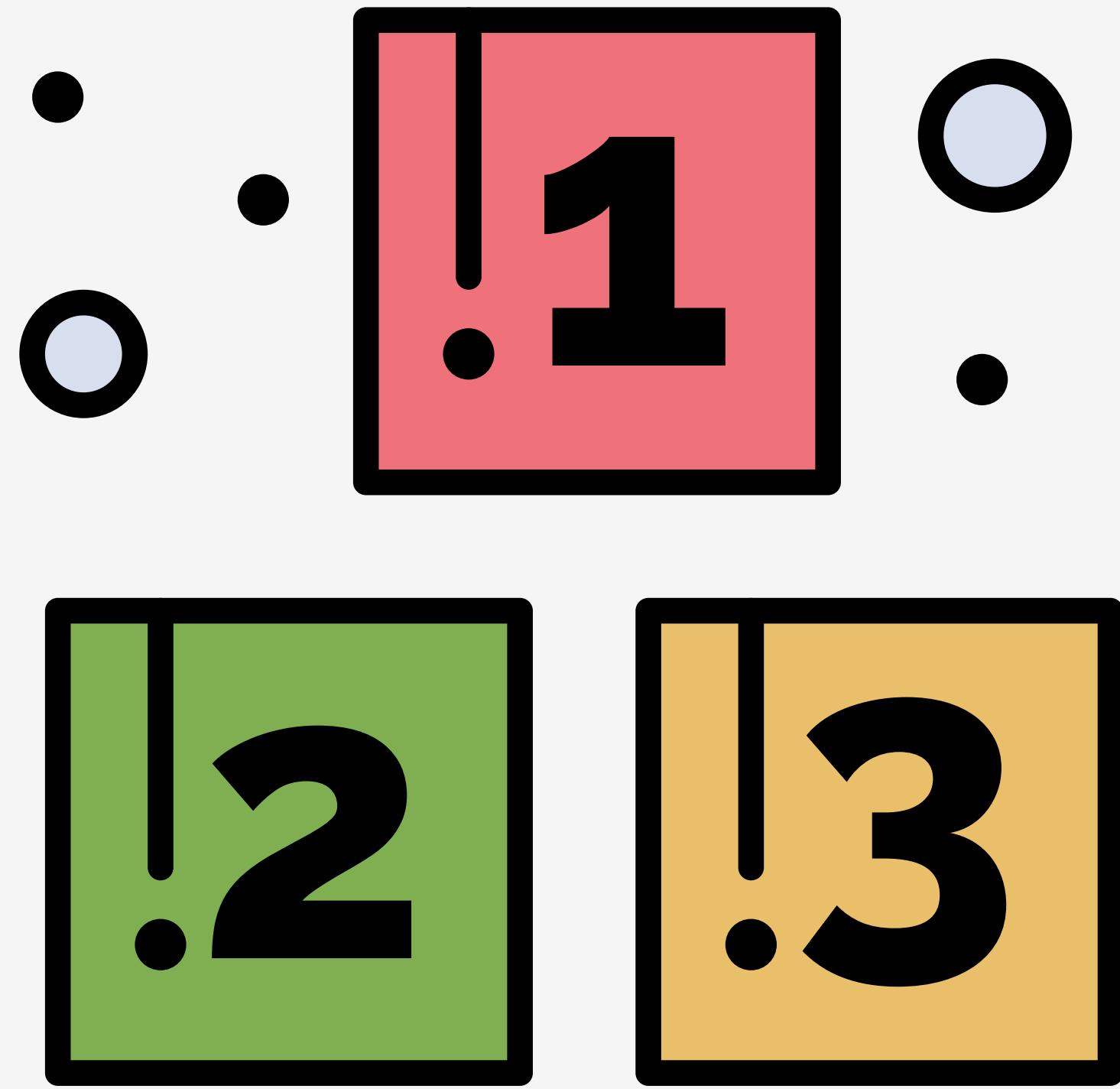
SASS @FOR

SASS @FOR



- The `@for` rule is used to count through a **range of numbers**
- The `@for` rule expression includes an **iterator** and a **range of numbers**
- The **range of numbers** can be connected using `to`, which excludes the final number, or `through`, which includes the final number
- For each iteration, the iterator is set to the current number in the range and the block is evaluated

SASS INTERPOLATION



- **Interpolation** is used to embed the results of an expression into CSS
- **Interpolation** syntax is `#{ }`
- **Interpolation** is often used to create selectors, property names and custom property values

```
/* Sass */
// loops 10 times
@for $i from 1 through 10 {
  .box:nth-child(#{ $i }) {
    background-color: lighten(#000, $i * 10%);
  }
}
```

```
/* CSS */
.box:nth-child(1) {
  background-color: #1a1a1a;
}

.box:nth-child(2) {
  background-color: #333333;
}

...

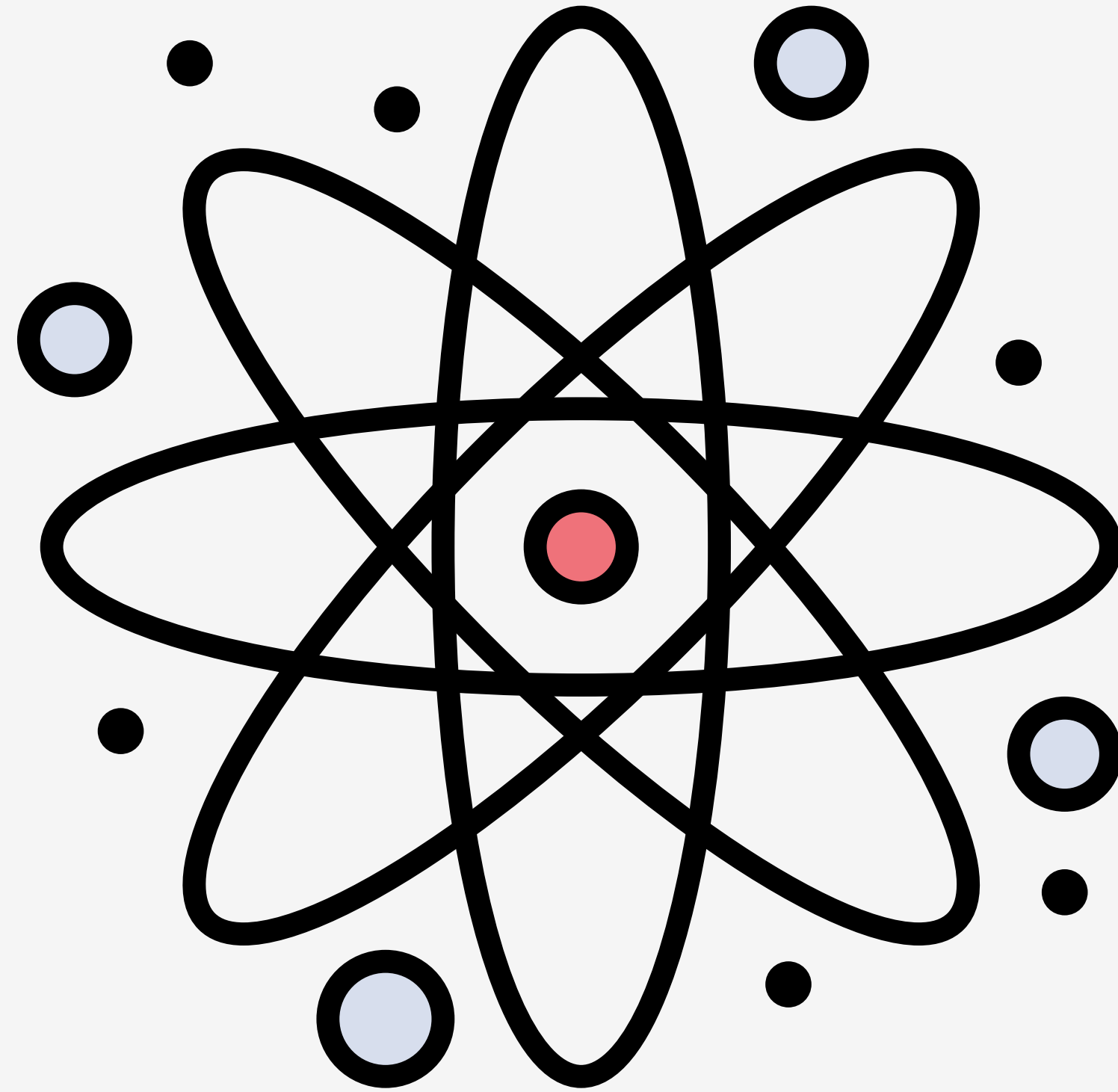
.box:nth-child(10) {
  background-color: white;
}
```

```
/* Sass */  
// loops 9 times  
@for $i from 1 to 10 {  
  .box:nth-child(#{ $i }) {  
    background-color: lighten(#000, $i * 10%);  
  }  
}
```

```
/* CSS */  
.box:nth-child(1) {  
  background-color: #1a1a1a;  
}  
  
.box:nth-child(2) {  
  background-color: #333333;  
}  
  
...  
  
.box:nth-child(9) {  
  background-color: #e6e6e6;  
}
```

SASS @EACH

SASS @EACH



- The `@each` rule will iterate over each element in a `list` or `map`
- For each element, the block will be evaluated
- The block will have access to the key and value of each element

```
/* Sass */
$colors: red, blue, green;

%box {
  display: inline-block;
  width: 100px;
  height: 100px;
}

@each $color in $colors {
  .box-#{ $color } {
    @extend %box;
    background-color: $color;
  }
}
```

```
/* CSS */
.box-red, .box-blue, .box-green, .box-orange {
  display: inline-block;
  width: 100px;
  height: 100px;
}

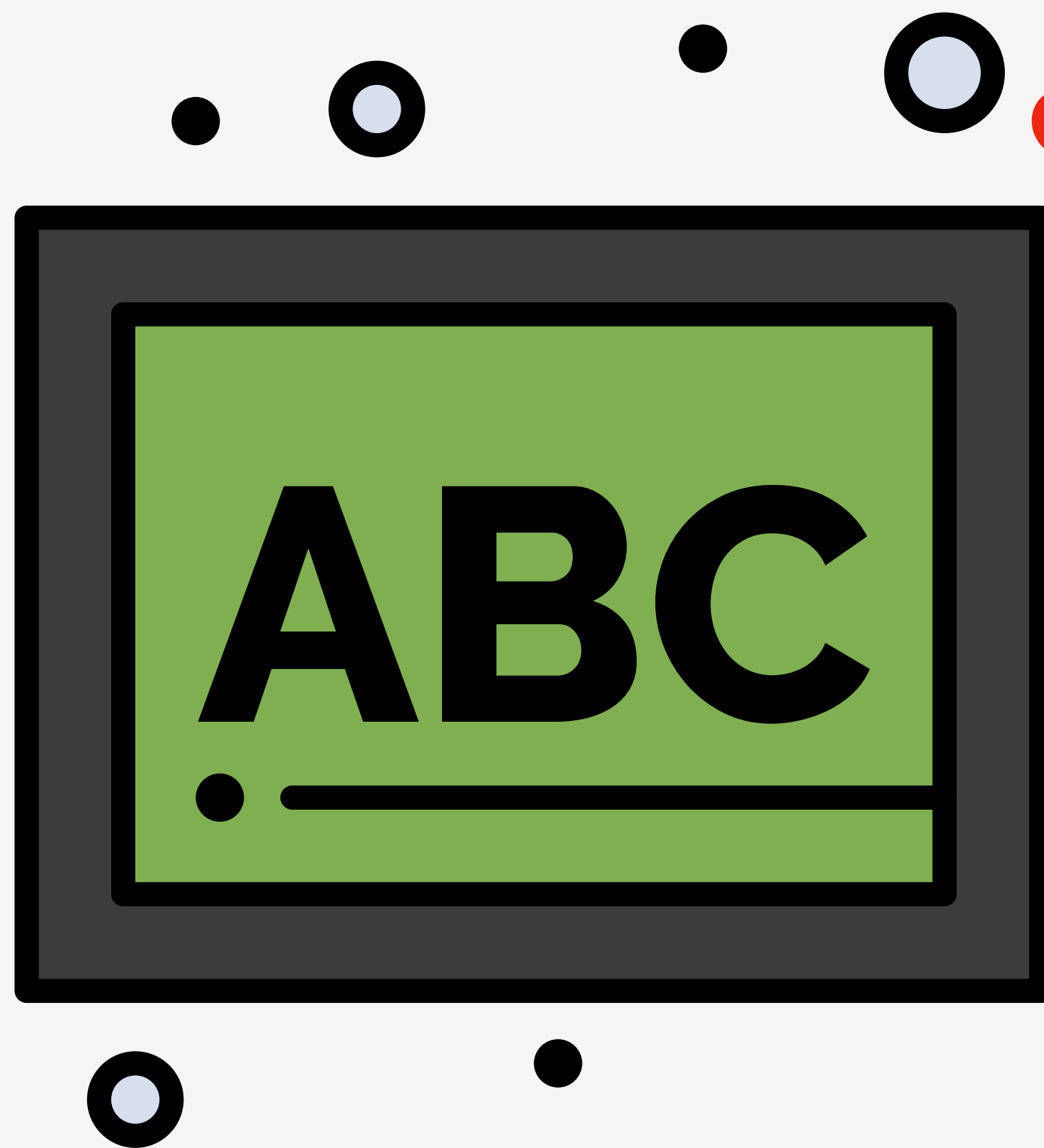
.box-red {
  background-color: red;
}

.box-blue {
  background-color: blue;
}

.box-green {
  background-color: green;
}
```

SASS MAPS

SASS MAP



Going Away

- **Maps** are a collection of key / value pairs
- Keys and values are separated by colon (**:**) and each pair is separated by a comma (**,**)
- Maps **MUST** be written with parentheses
- The **map-get()** function is used to retrieve a value from a Map


```
/* Sass */
$grey: (
  "light": #eee,
  "medium": #999,
  "dark": #666
);

.box {
  background-color: map-get($grey, "light");
  border: 2px solid map-get($grey, "medium");
  color: map-get($grey, "dark");
}
```

```
/* CSS */
.box {
  background-color: #eee;
  border: 2px solid #999;
  color: #666;
}
```

```
/* Sass */
$buttons: (
  "default": #6c757d,
  "cool": #007bff,
  "hot": #dc3545,
);

@each $button, $color in $buttons {
  .button-#{ $button } {
    background-color: $color;
    border-color: $color;
  }
}
```

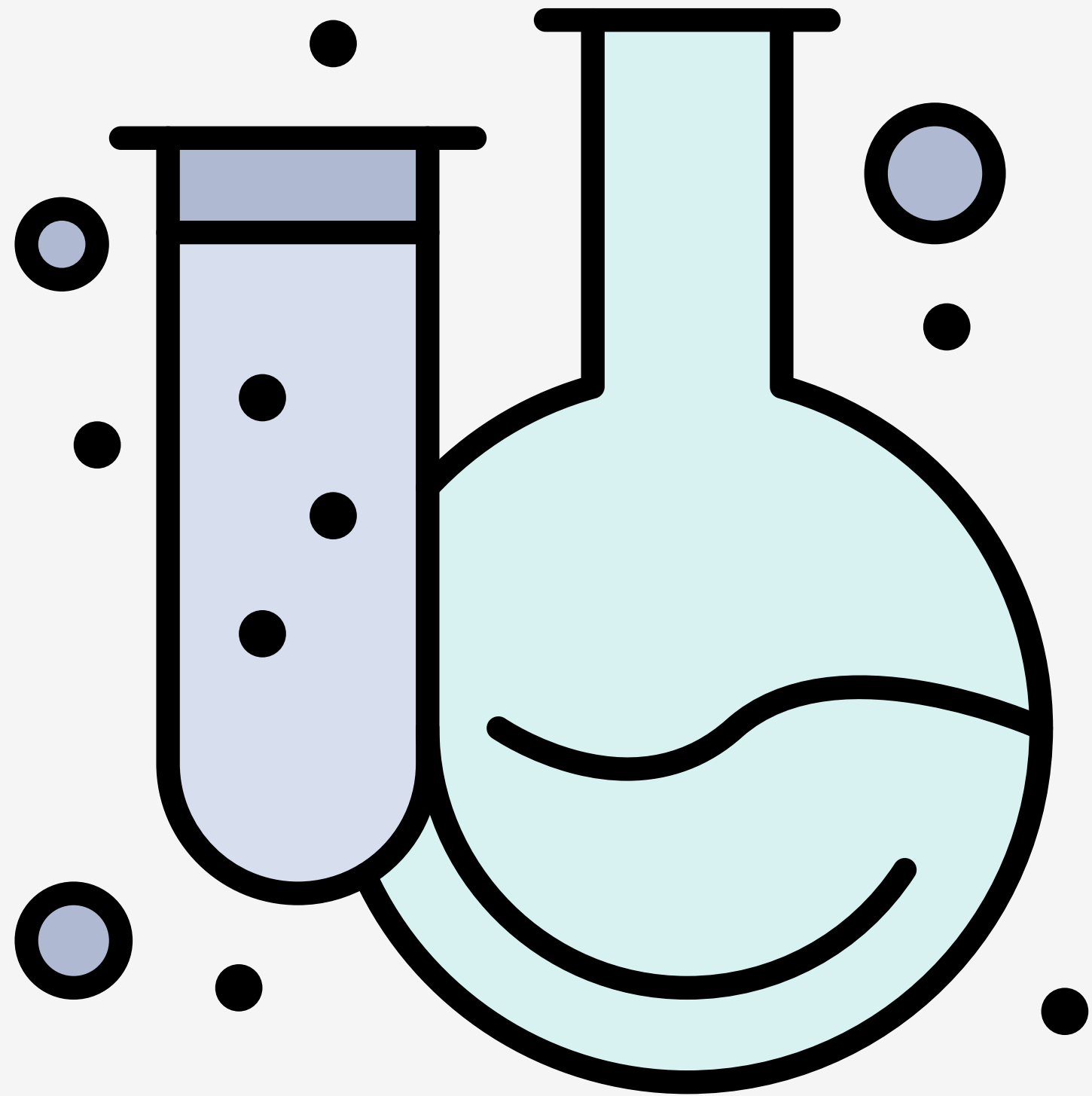
```
/* CSS */
.button-default {
  background-color: #6c757d;
  border-color: #6c757d;
}

.button-cool {
  background-color: #007bff;
  border-color: #007bff;
}

.button-hot {
  background-color: #dc3545;
  border-color: #dc3545;
}
```

PRACTICE

SASSY CATS



- *FORK THE PEN!*
- Use Sass to create the CSS styles to make cats different sizes and colors
- Use a `@for` rule to create the sizes classes.
- Create a list and use the `@each` rule to create the colors
- Submit the URL to your pen
- *DUE:* Thu. Nov. 20 @ 11:59 PM

NEXT TIME...

- Hands-on: Icon Maker

