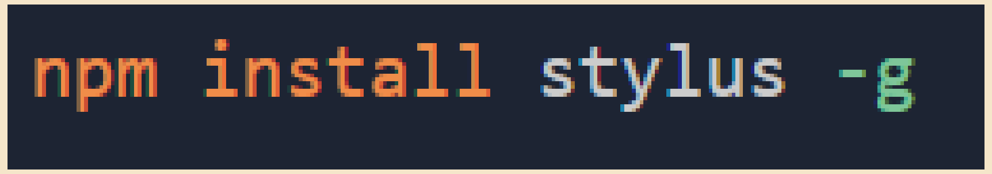
PREPROCESSOR

STYLUS

DANIEL FERNANDEZ & ERICK MONTES

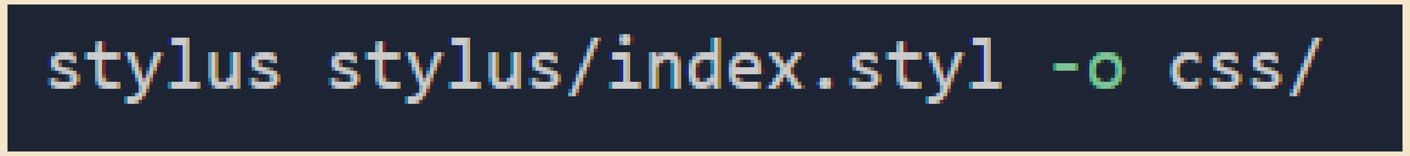
SETTING UP

- Install stylus



```
npm install stylus -g
```

- Compile into CSS files



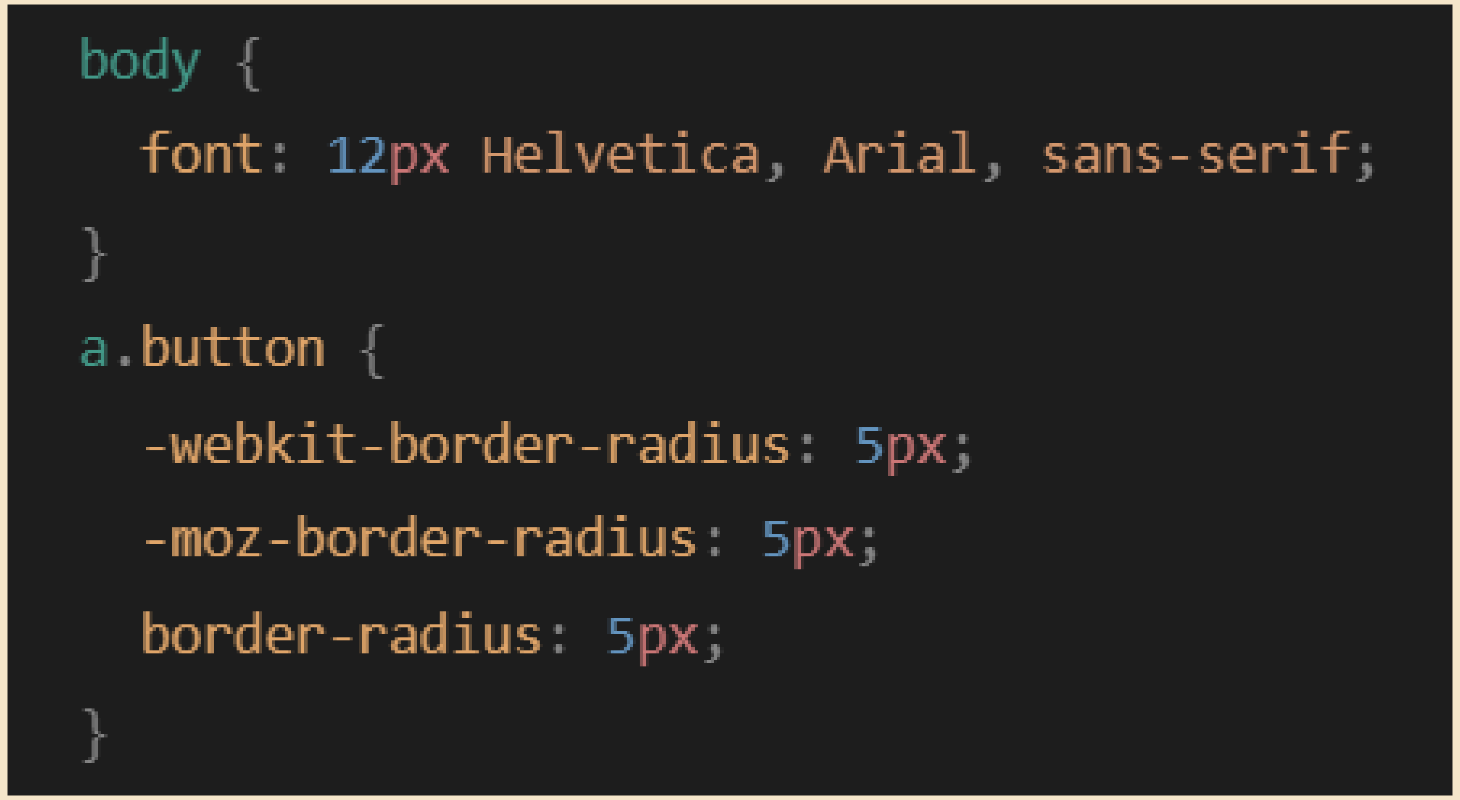
```
stylus stylus/index.styl -o css/
```

stylus file

output folder

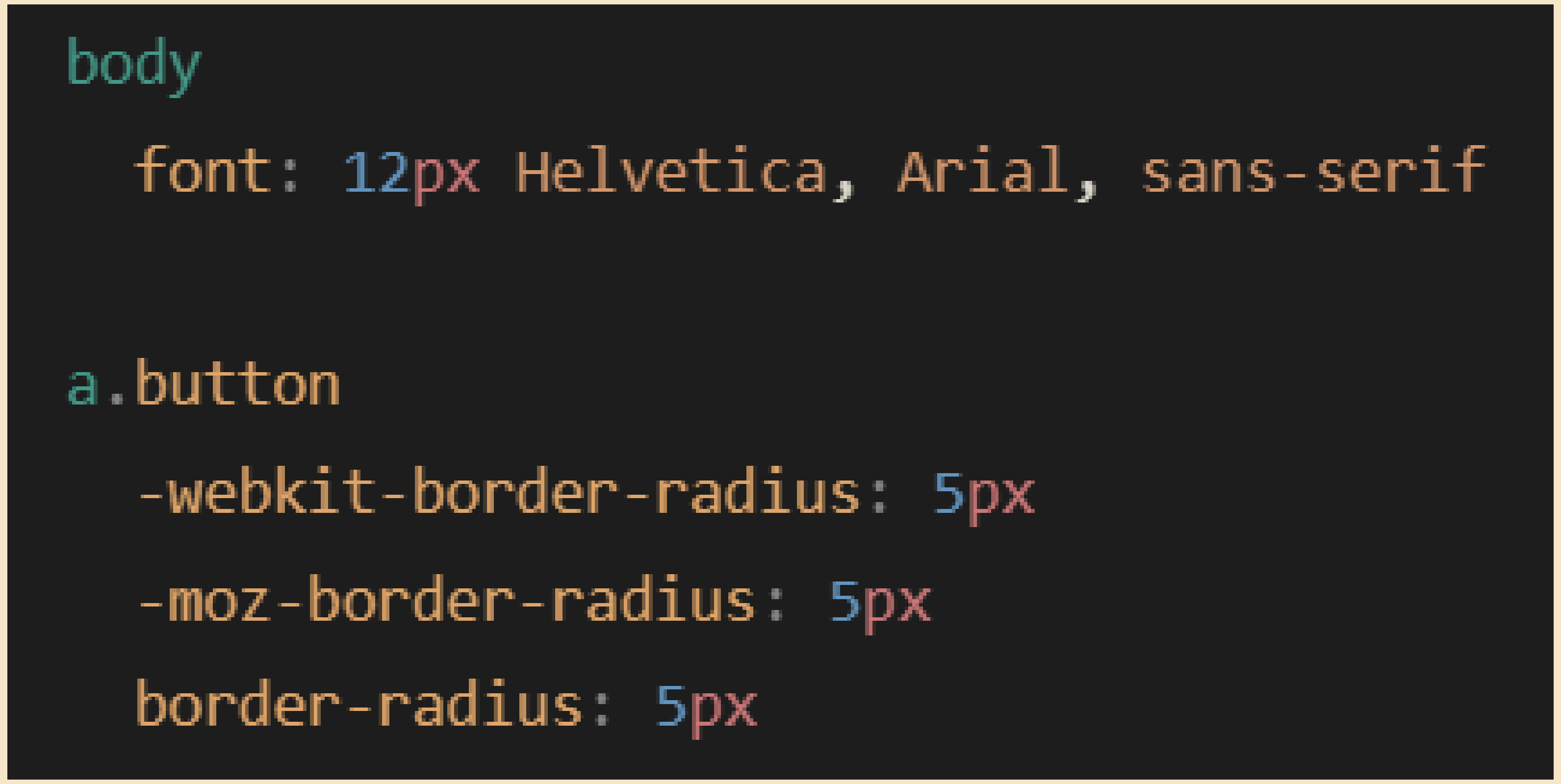
SINTAX

- You don't need keys to mark each css block, just use the indentation.
- Also you can omit the semi-colons
- You can use regular css



```
body {
  font: 12px Helvetica, Arial, sans-serif;
}
a.button {
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
  border-radius: 5px;
}
```

CSS

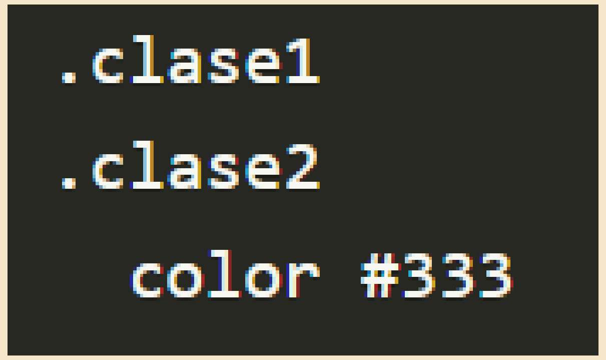


```
body
  font: 12px Helvetica, Arial, sans-serif

a.button
  -webkit-border-radius: 5px
  -moz-border-radius: 5px
  border-radius: 5px
```

Stylus

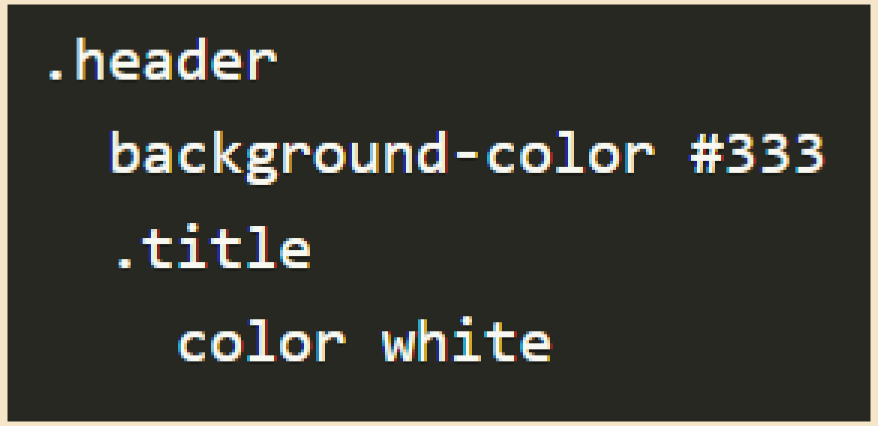
SELECTORS AND NESTING

- Multiple objects



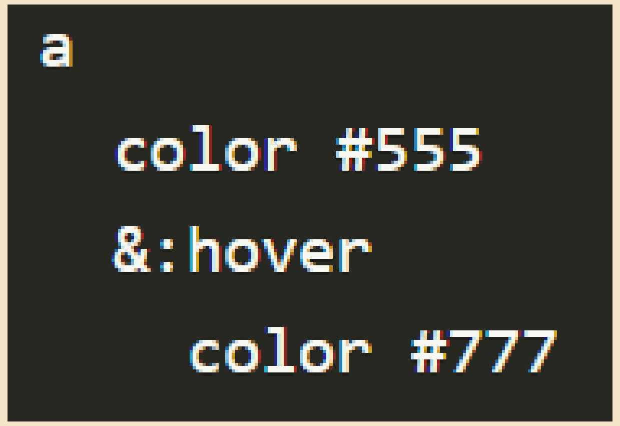
```
.clase1
.clase2
  color #333
```

- Nesting



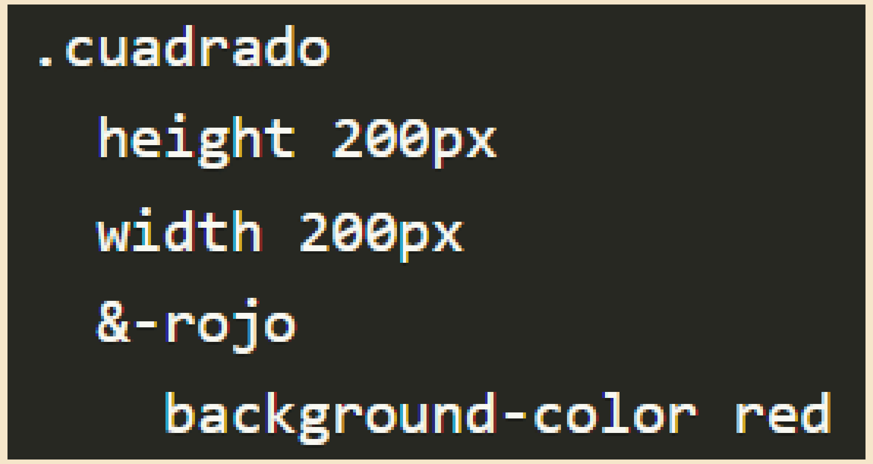
```
.header
  background-color #333
  .title
    color white
```

- Pseudo-class

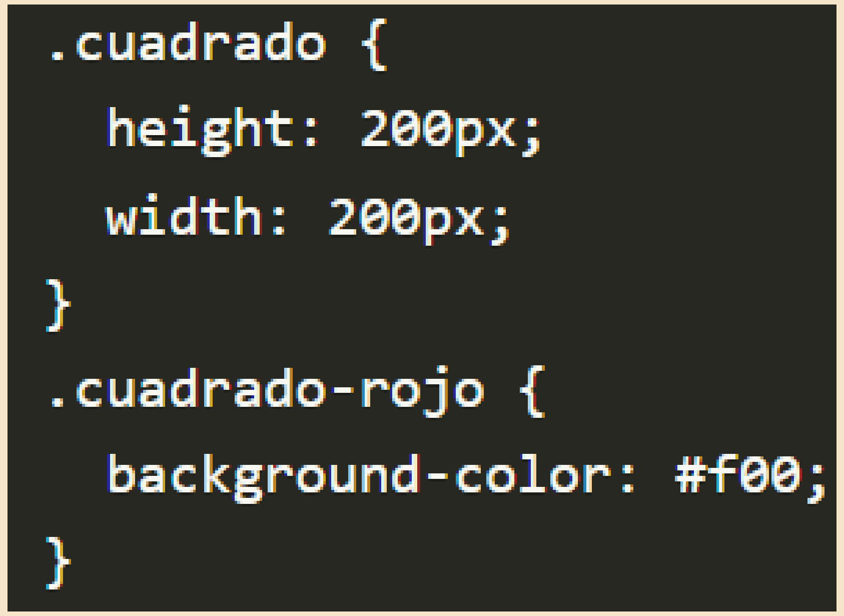


```
a
  color #555
  &:hover
    color #777
```

- New selectors



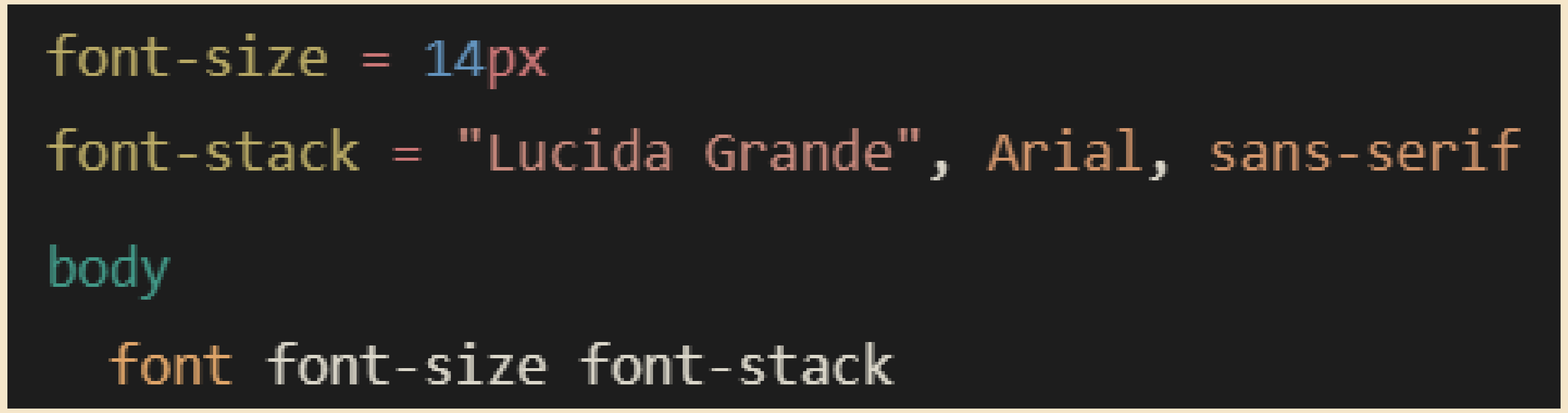
```
.cuadrado
  height 200px
  width 200px
  &-rojo
    background-color red
```



```
.cuadrado {
  height: 200px;
  width: 200px;
}
.cuadrado-rojo {
  background-color: #f00;
}
```

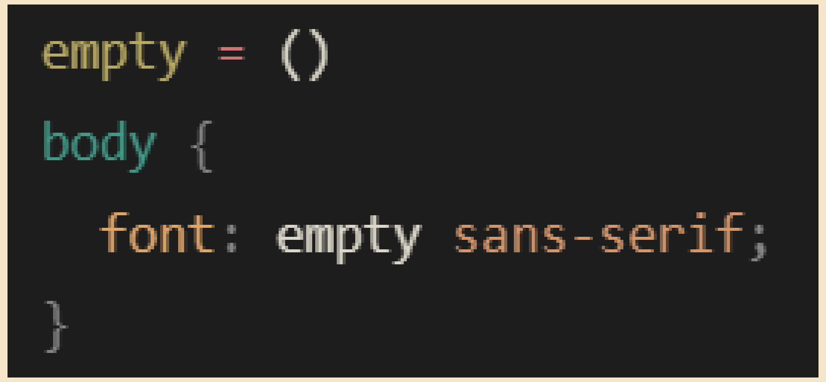
VARIABLES

- Don't need a symbol to declare



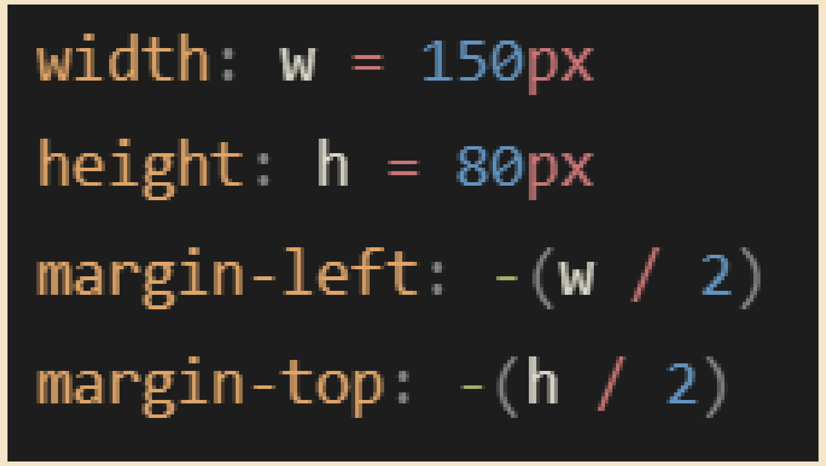
```
font-size = 14px
font-stack = "Lucida Grande", Arial, sans-serif

body
  font font-size font-stack
```

- Can declare a empty variable

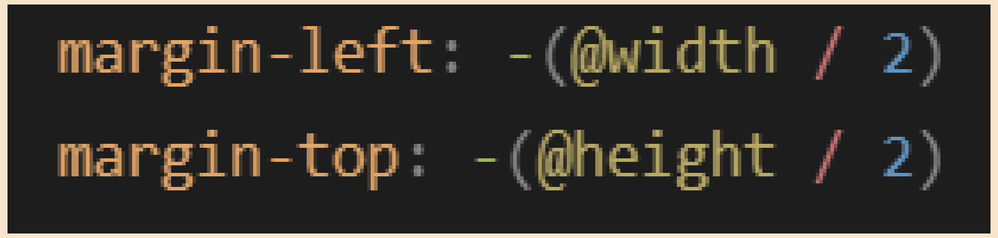


```
empty = ()
body {
  font: empty sans-serif;
}
```

- Can reference properties defined without assigning their values to variables



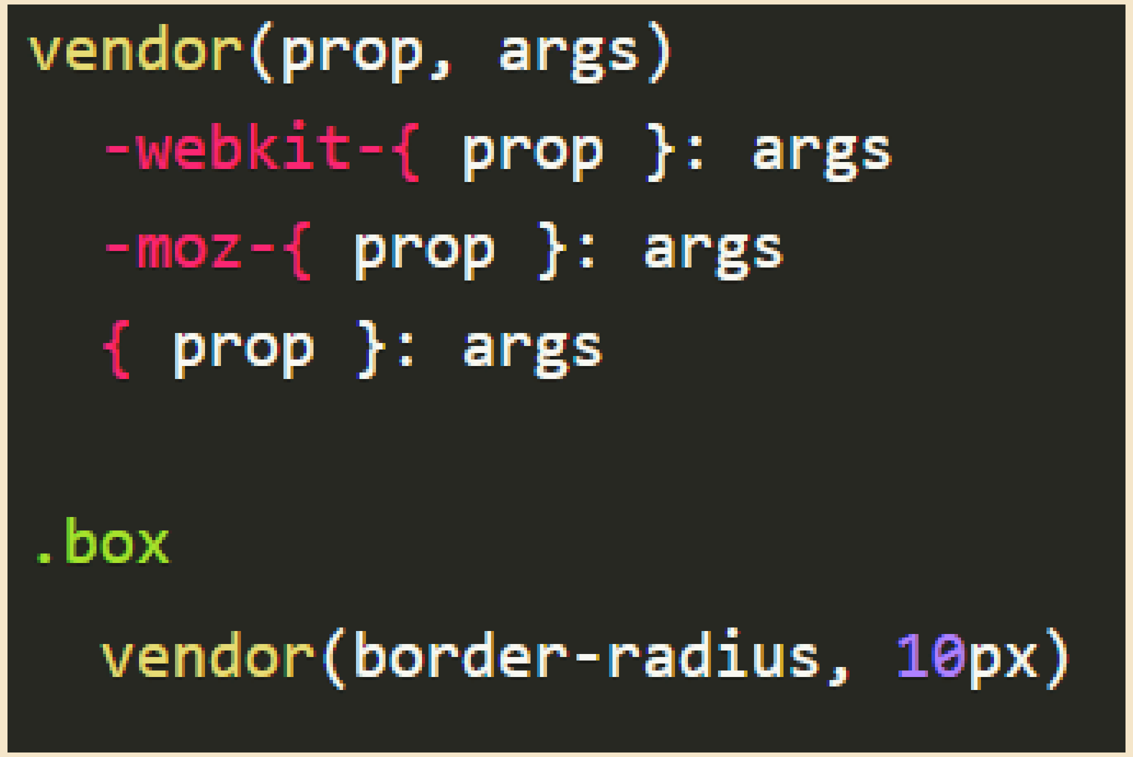
```
width: w = 150px
height: h = 80px
margin-left: -(w / 2)
margin-top: -(h / 2)
```



```
margin-left: -(@width / 2)
margin-top: -(@height / 2)
```

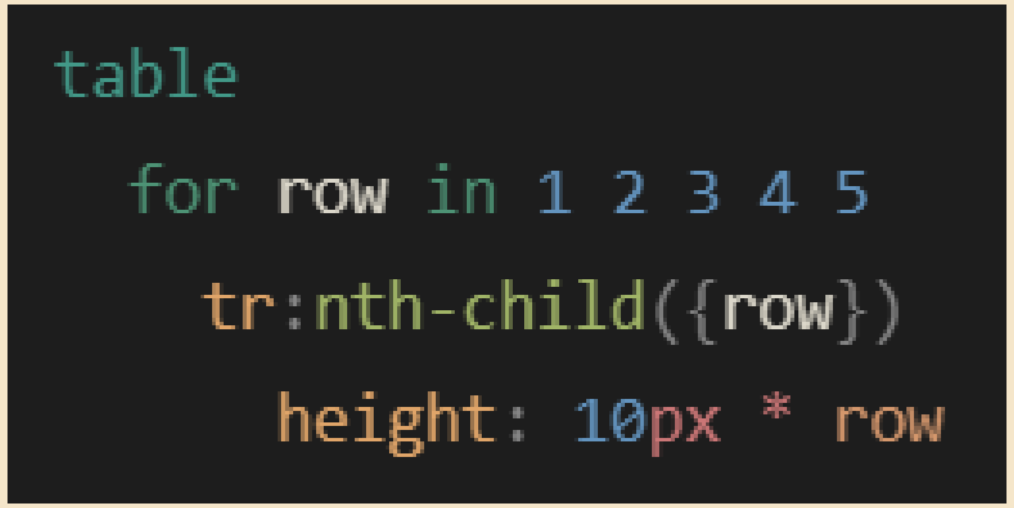
INTERPOLATION

- Expanding properties

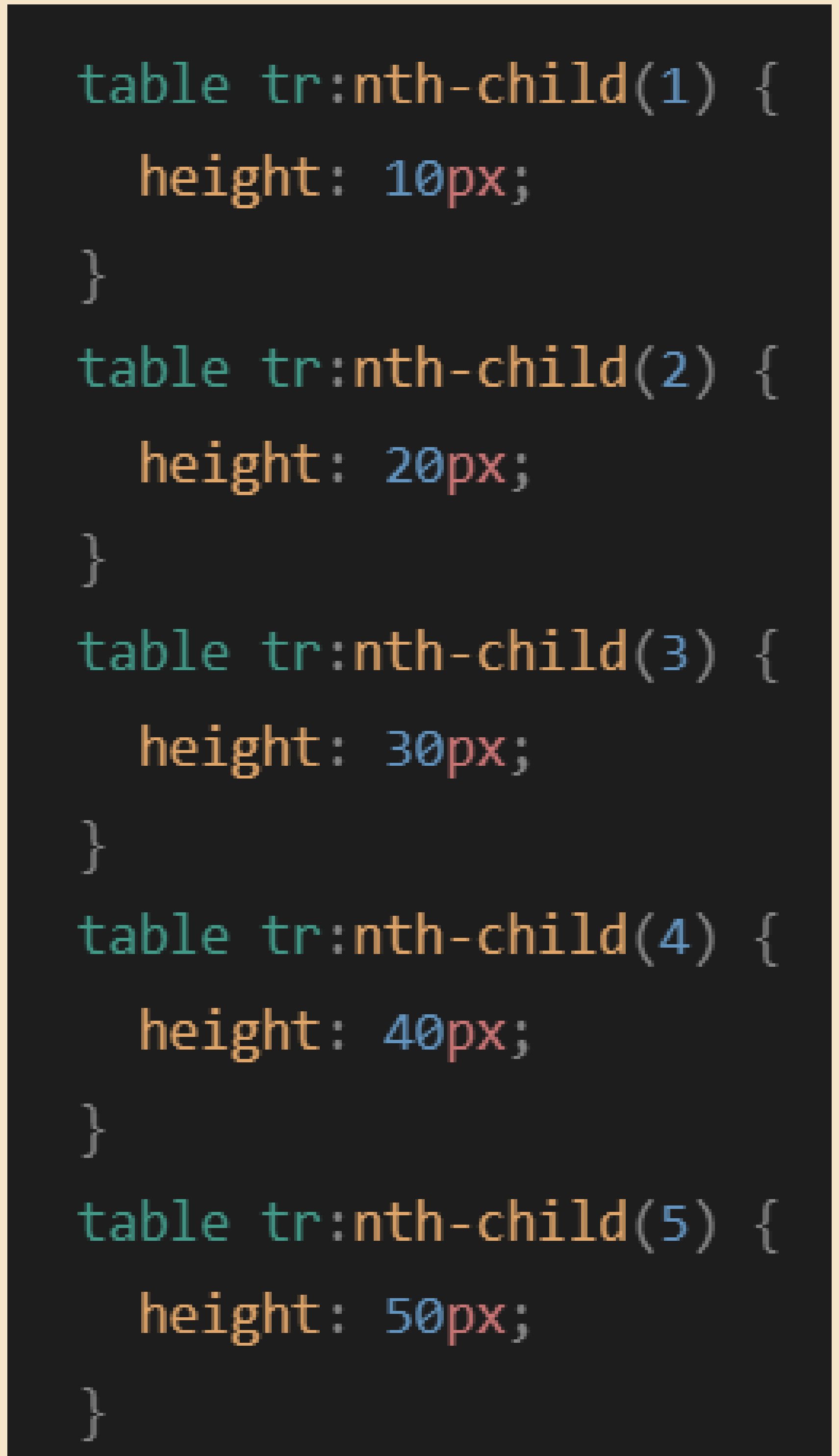


```
vendor(prop, args)
  -webkit-{ prop }: args
  -moz-{ prop }: args
  { prop }: args

.box
  vendor(border-radius, 10px)
```

- Iteration



```
table
  for row in 1 2 3 4 5
    tr:nth-child({row})
      height: 10px * row
```

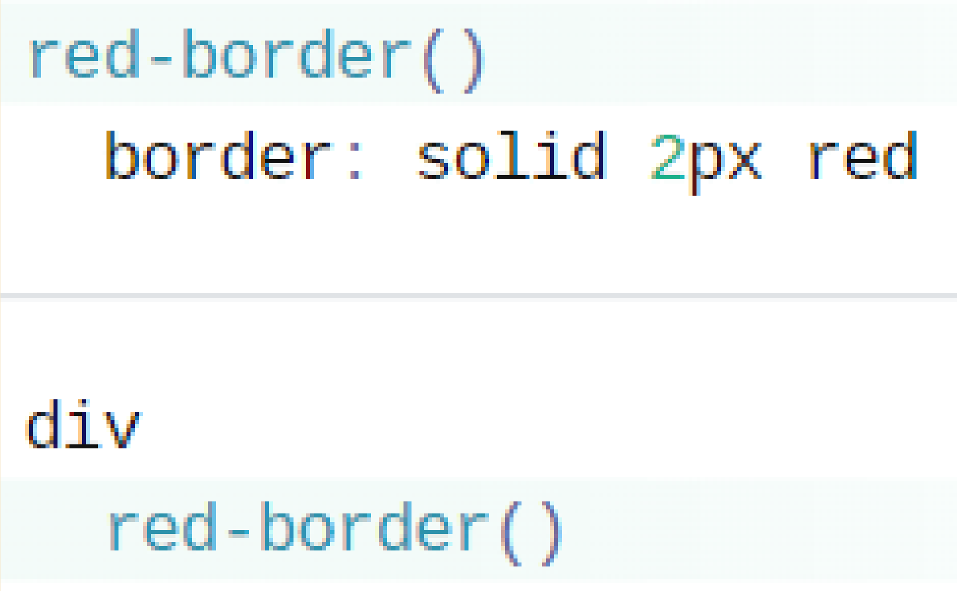


```
table tr:nth-child(1) {
  height: 10px;
}
table tr:nth-child(2) {
  height: 20px;
}
table tr:nth-child(3) {
  height: 30px;
}
table tr:nth-child(4) {
  height: 40px;
}
table tr:nth-child(5) {
  height: 50px;
}
```

MIXINS

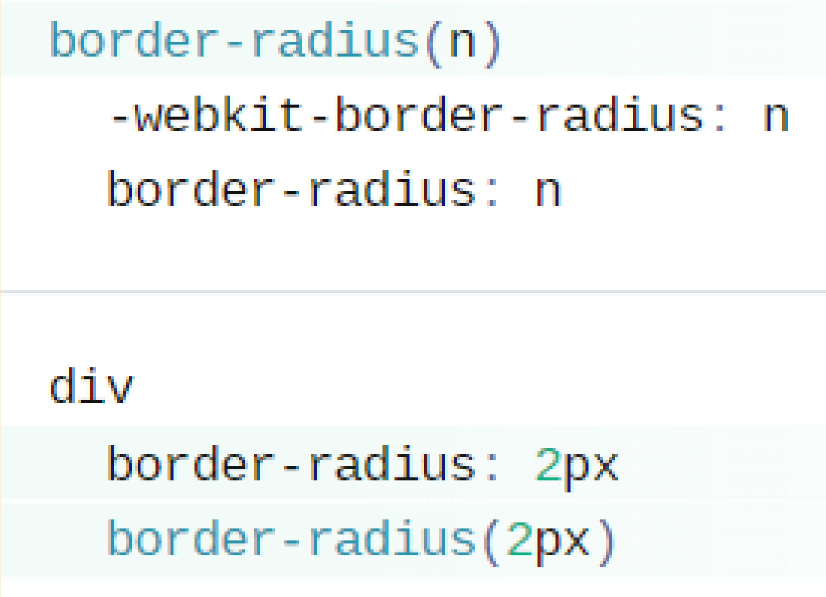
Component that can be reused. It also can receive arguments, have some by default or use rest params

- without argument



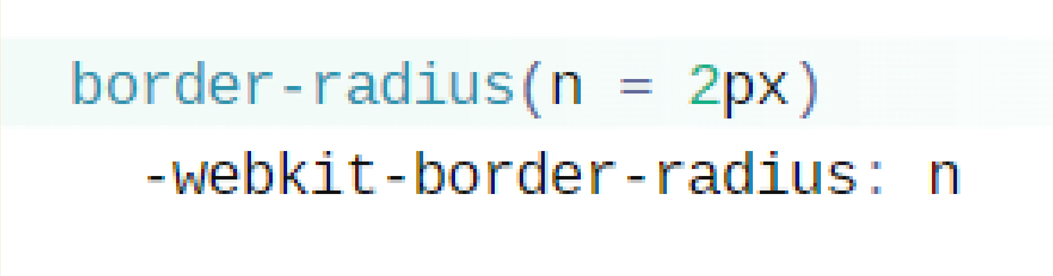
```
red-border()
  border: solid 2px red

div
  red-border()
```

- with argument



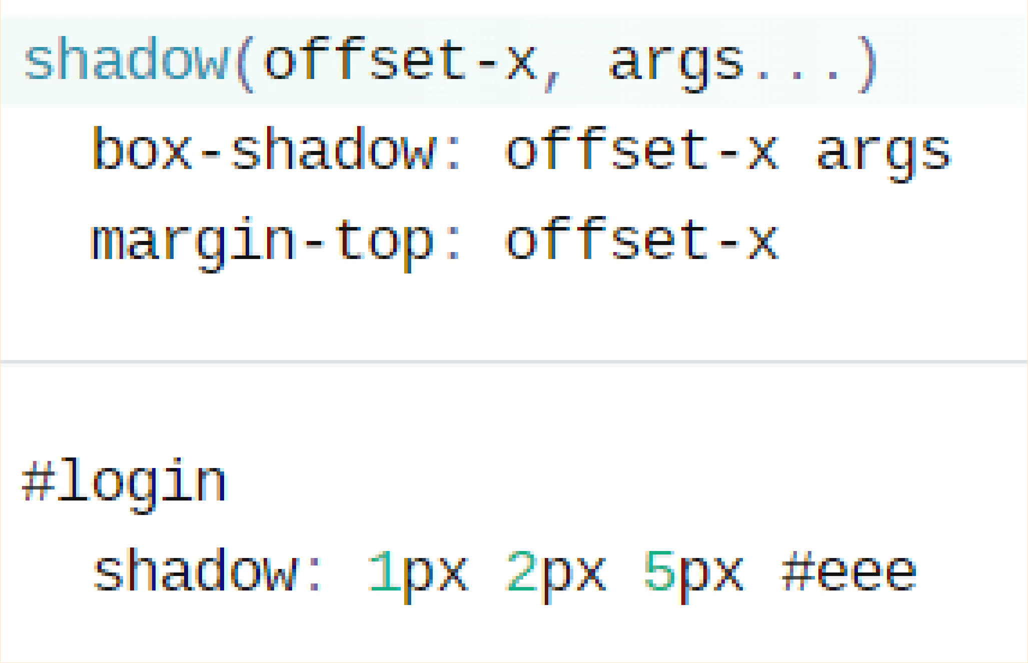
```
border-radius(n)
  -webkit-border-radius: n
  border-radius: n

div
  border-radius: 2px
  border-radius(2px)
```

- default argument



```
 border-radius(n = 2px)
   -webkit-border-radius: n
```

- rest params

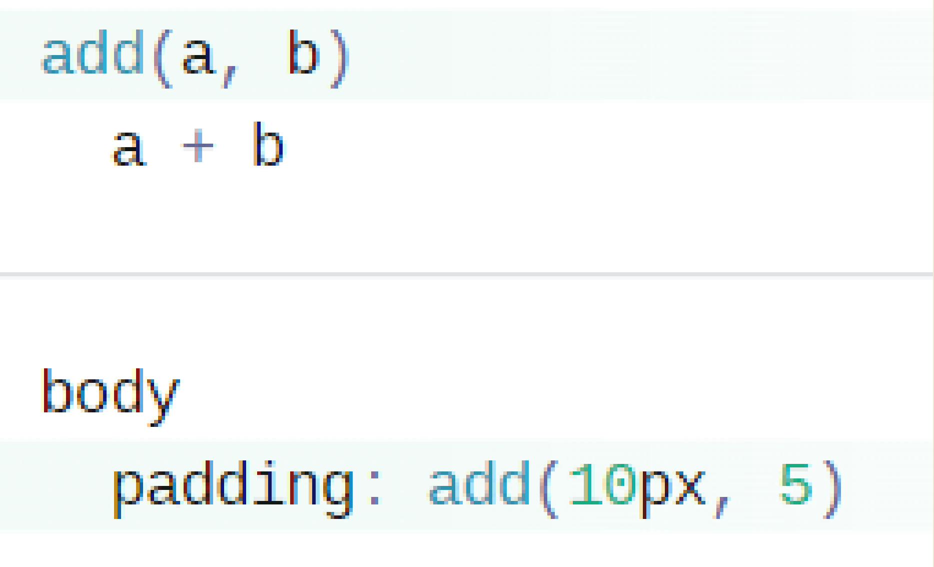


```
shadow(offset-x, args...)
  box-shadow: offset-x args
  margin-top: offset-x

#login
  shadow: 1px 2px 5px #eee
```

FUNCTIONS

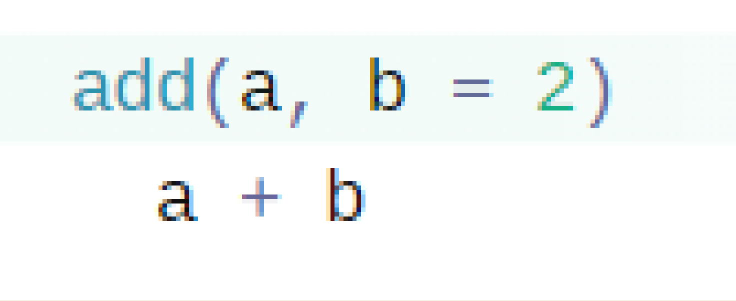
It is similar to mixins, However, functions return data, and mixins do not return data.

- Return values



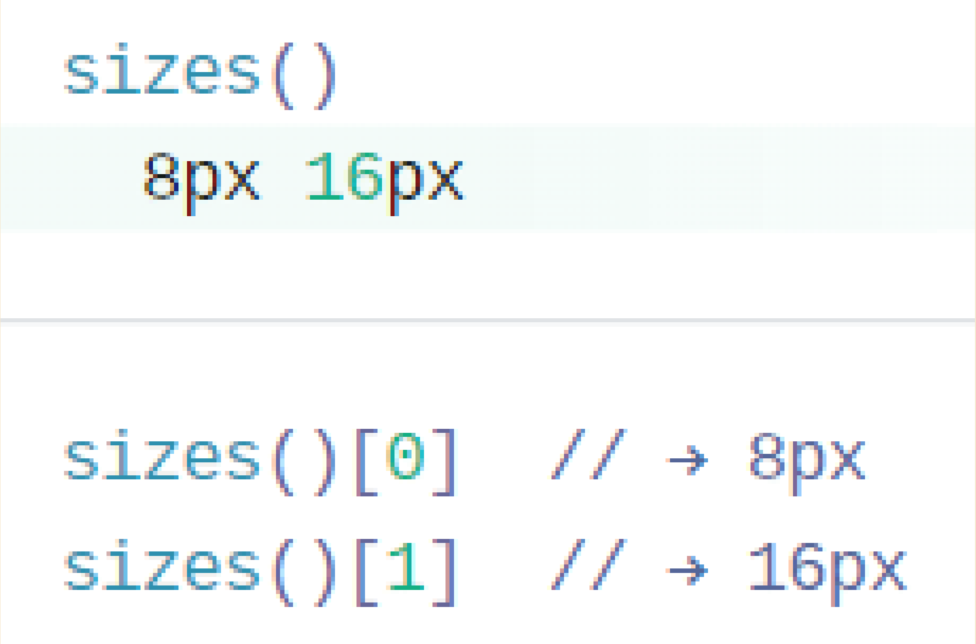
```
add(a, b)
  a + b


body
  padding: add(10px, 5)
```

- default argument



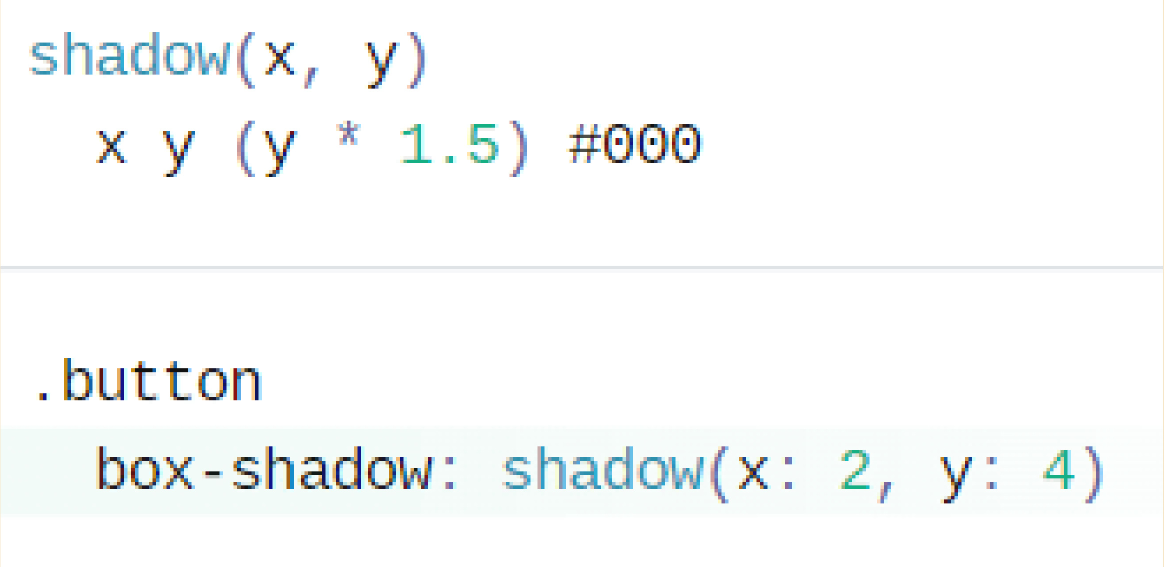
```
add(a, b = 2)
  a + b
```

- Multiple values



```
sizes()
  8px 16px


sizes()[0]  // → 8px
sizes()[1]  // → 16px
```

- named parameters



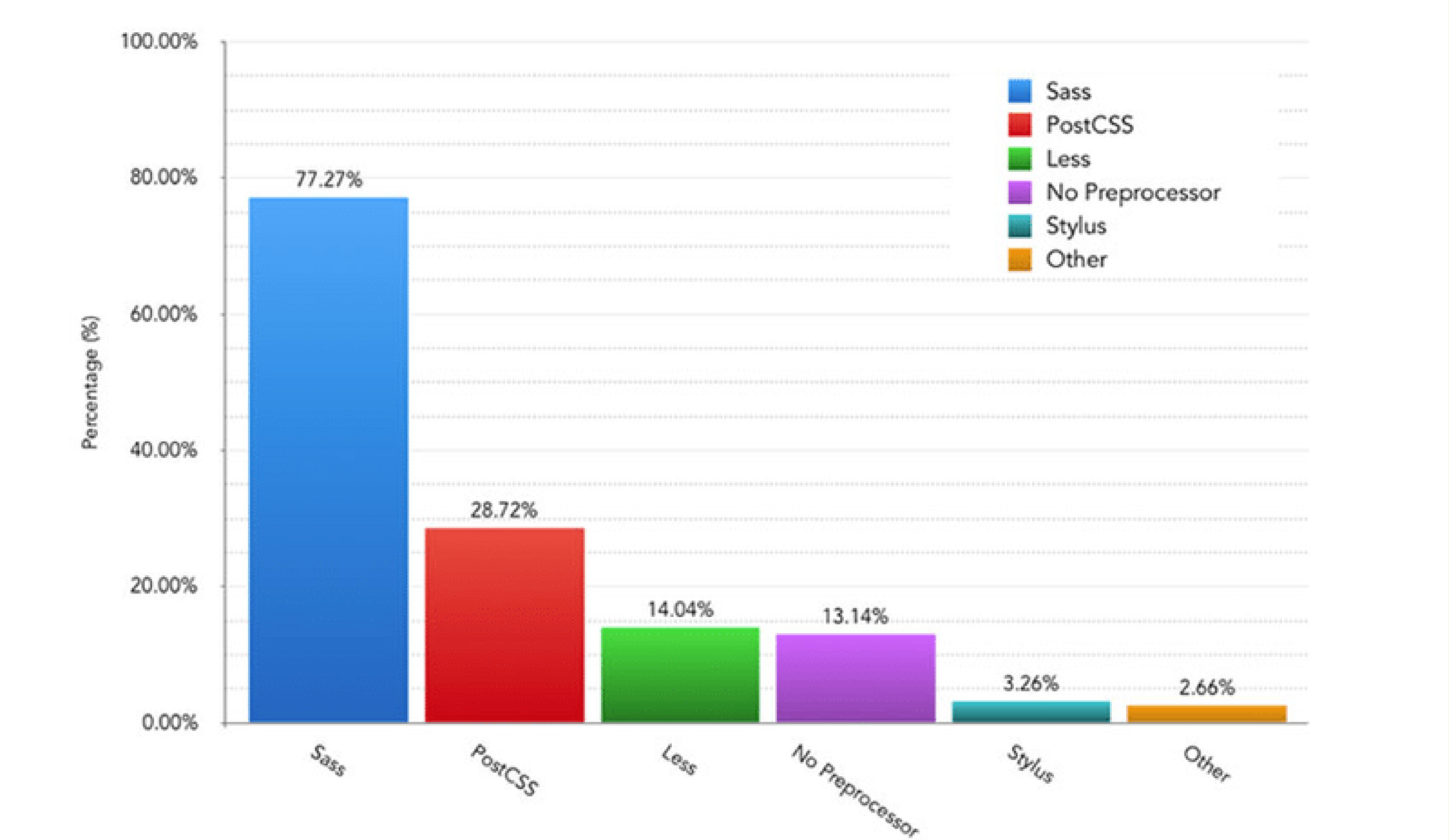
```
shadow(x, y)
  x y (y * 1.5) #000


.button
  box-shadow: shadow(x: 2, y: 4)
```

ADVANTAGES

- Simple syntax: It has a very concise and simple syntax that is easy to read and maintain.
- Flexibility: Stylus is adaptable, it enables the use of plugins
- Performance: It's known for its good performance, this is because it's written in JavaScript and uses a more simple and direct syntax.
- Lightweight: Stylus has a small size and is less complex than other preprocessors, making it easier to use and set up.
- Stylus can use raw CSS code, this makes easy to convert a file of CSS to stylus

DISADVANTAGES

- Not so popular: Stylus is less popular which means there may be fewer resources and community available to help with specific issues.



DISADVANTAGES

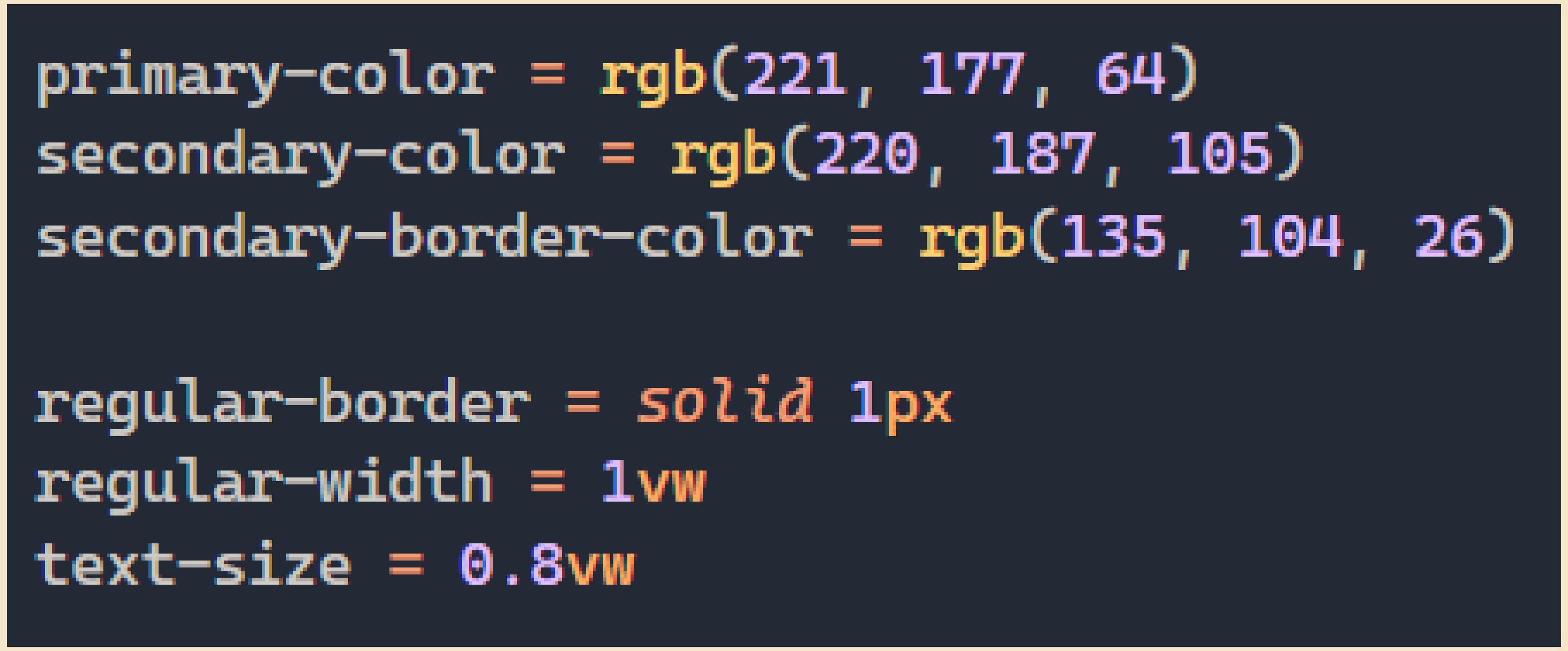
- Longer learning curve and increased complexity: Althought the sintax is simplier, it can take time to get used to the new features and concepts. You must have a good knowledge of the syntax and concepts in order to take full advantage of its potential.

- Longer compile time: When using the Stylus to write your CSS code, you'll need to use a compiler to convert your Stylus code to CSS before you can use it on your website. This can increase build time if your project is large.

YUGIOH BOARD

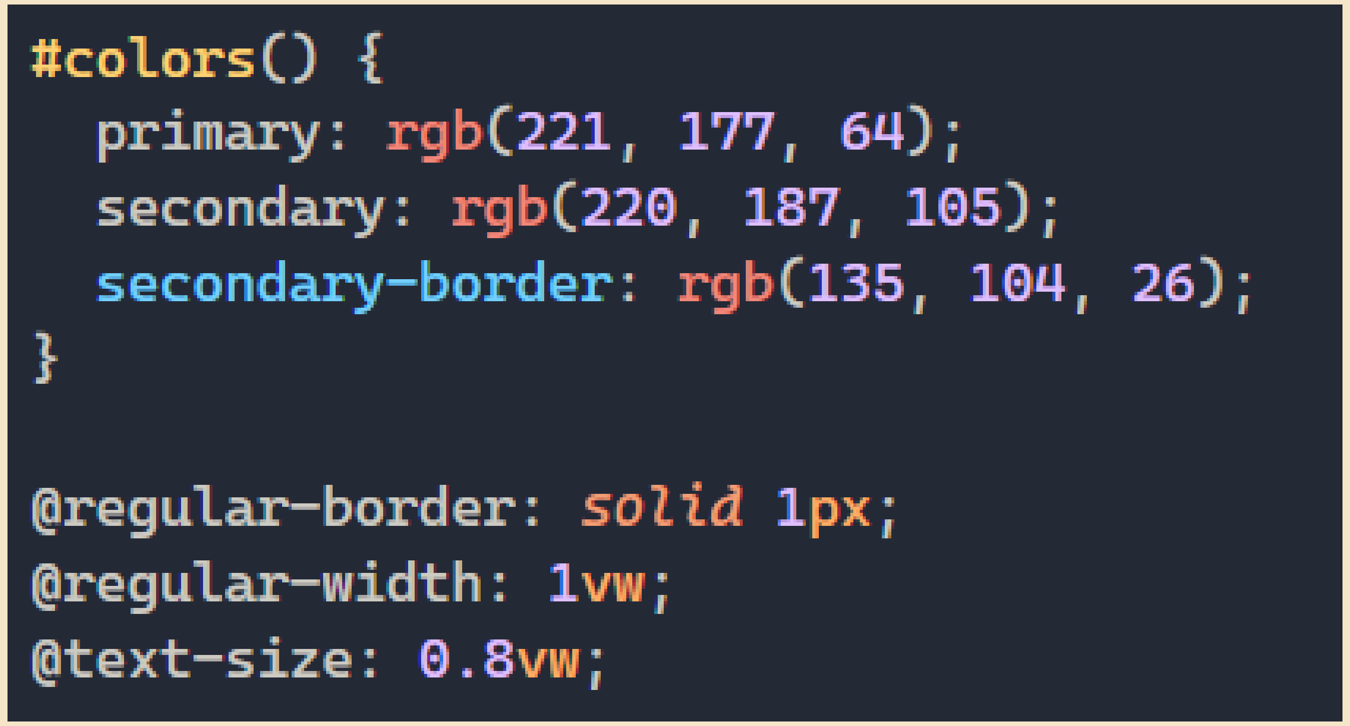
Variables

Stylus



```
primary-color = rgb(221, 177, 64)
secondary-color = rgb(220, 187, 105)
secondary-border-color = rgb(135, 104, 26)

regular-border = solid 1px
regular-width = 1vw
text-size = 0.8vw
```

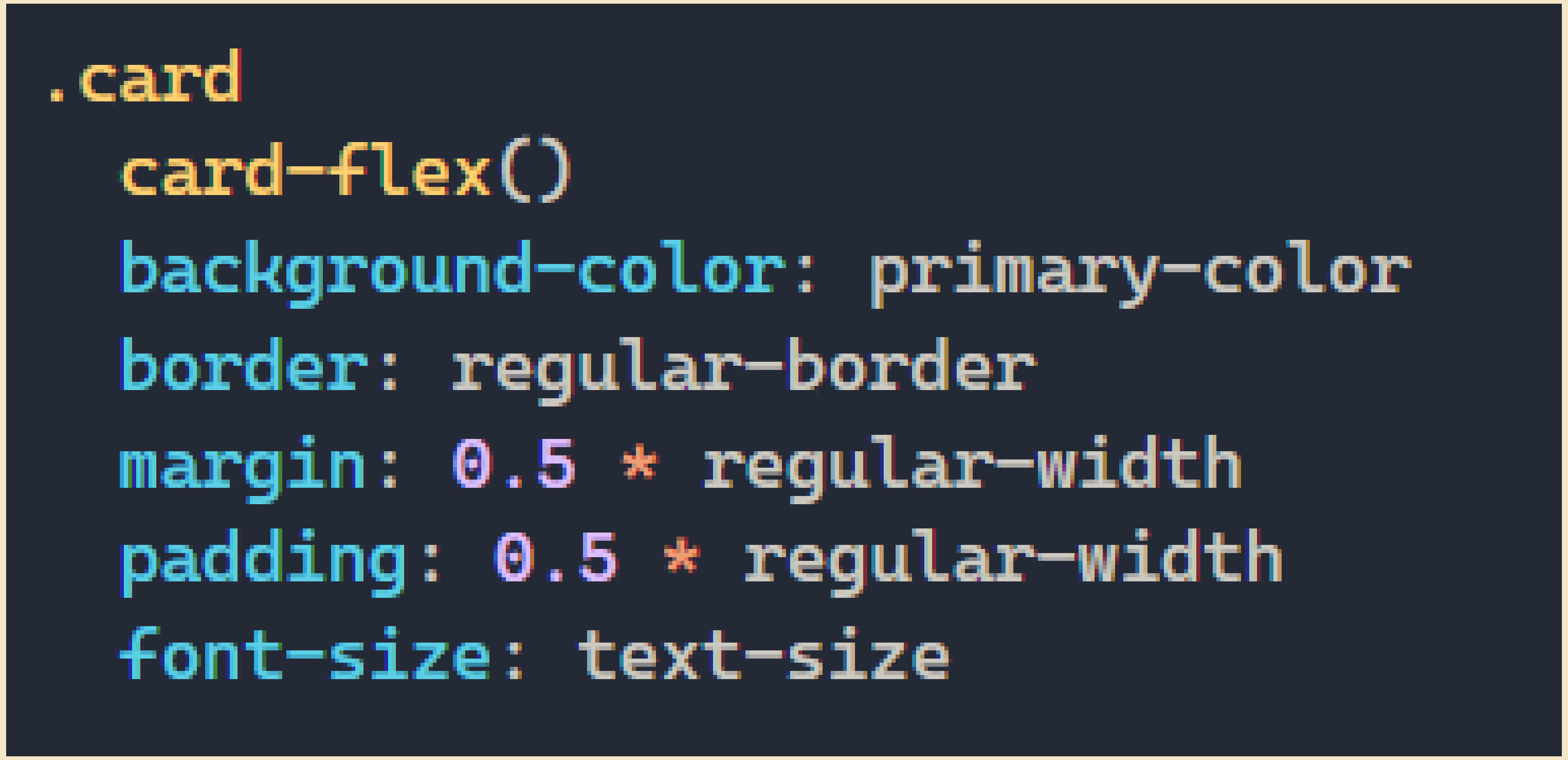
Less



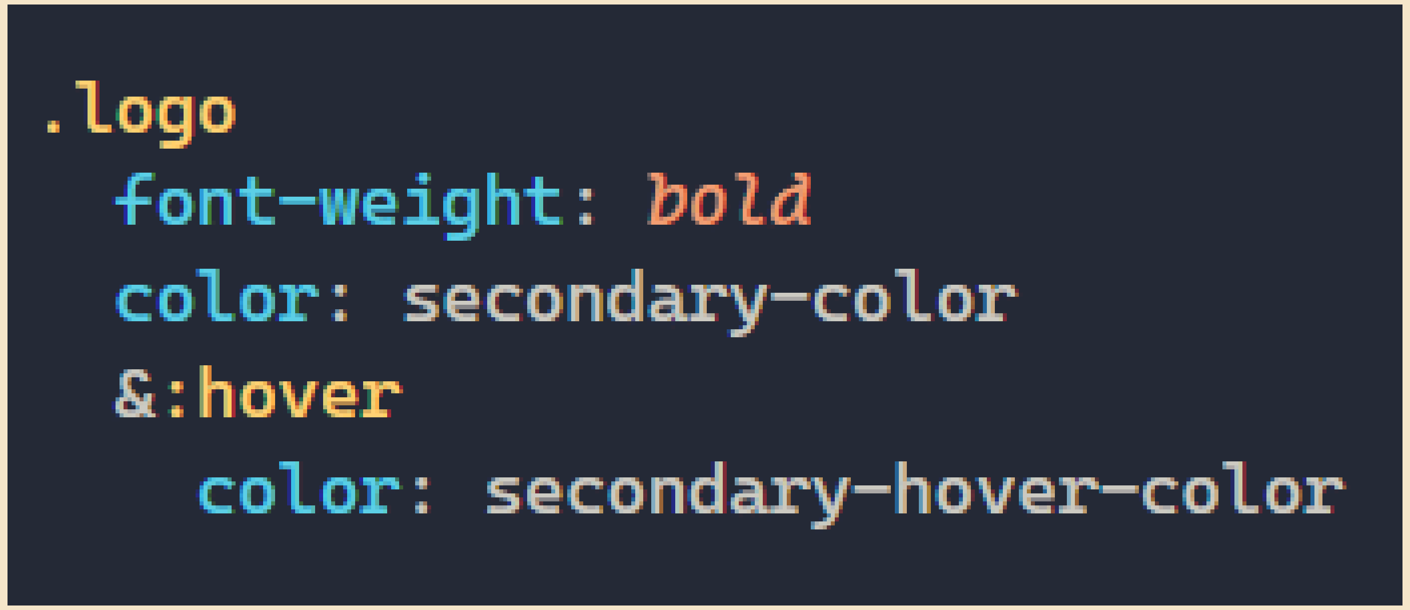
```
#colors() {
  primary: rgb(221, 177, 64);
  secondary: rgb(220, 187, 105);
  secondary-border: rgb(135, 104, 26);
}

@regular-border: solid 1px;
@regular-width: 1vw;
@text-size: 0.8vw;
```

YUGIOH BOARD

Selectors



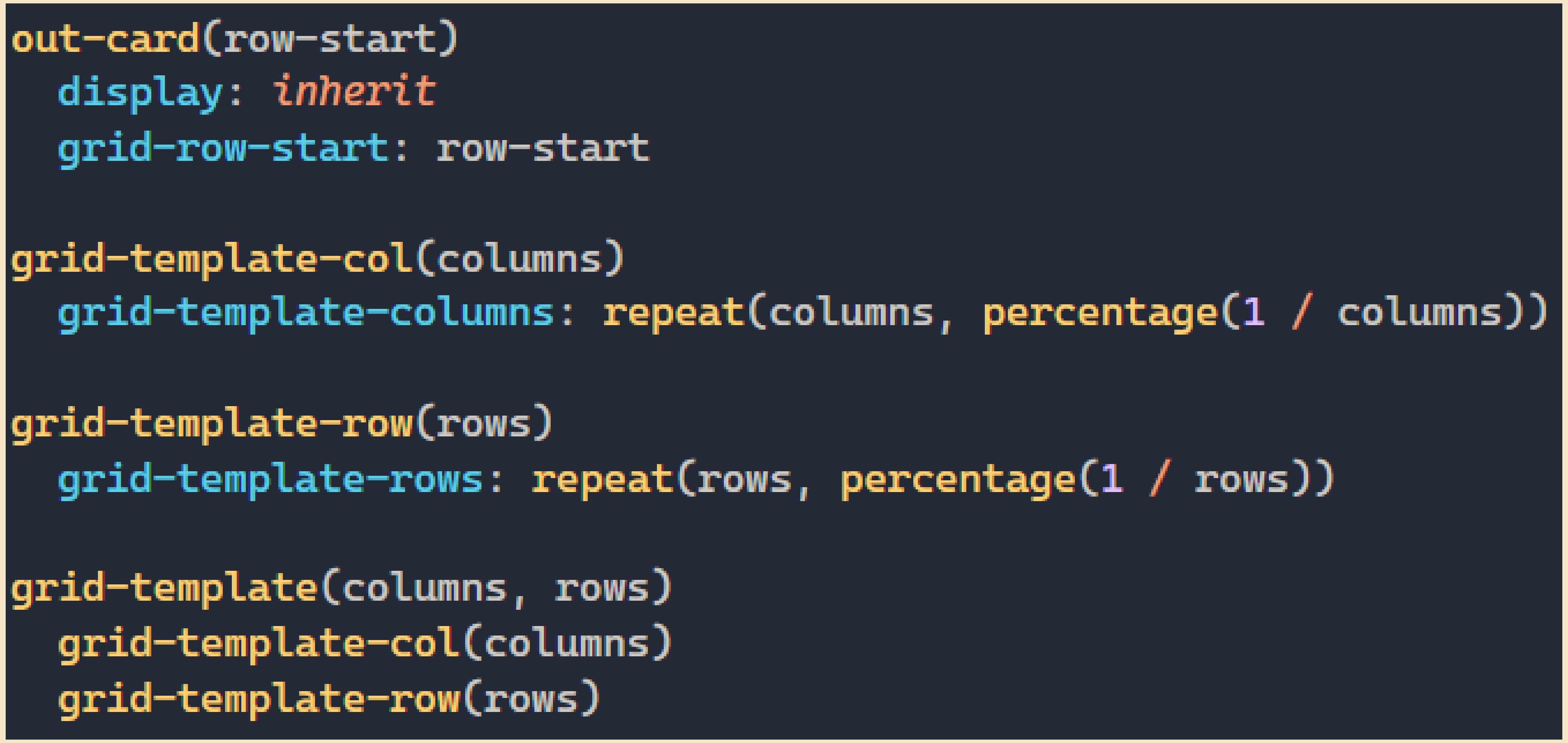
```
.card
  card-flex()
  background-color: primary-color
  border: regular-border
  margin: 0.5 * regular-width
  padding: 0.5 * regular-width
  font-size: text-size
```



```
.logo
  font-weight: bold
  color: secondary-color
  &:hover
    color: secondary-hover-color
```

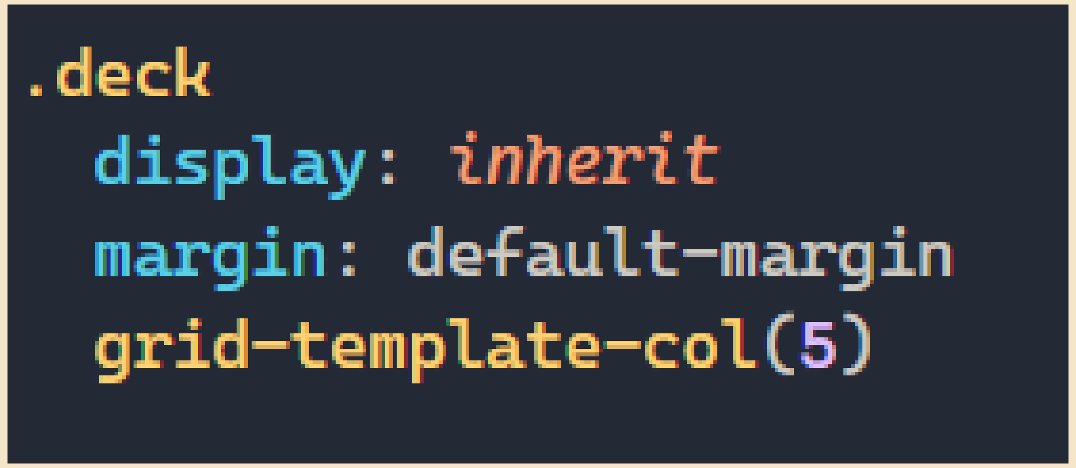
YUGIOH BOARD

Mixins and functions

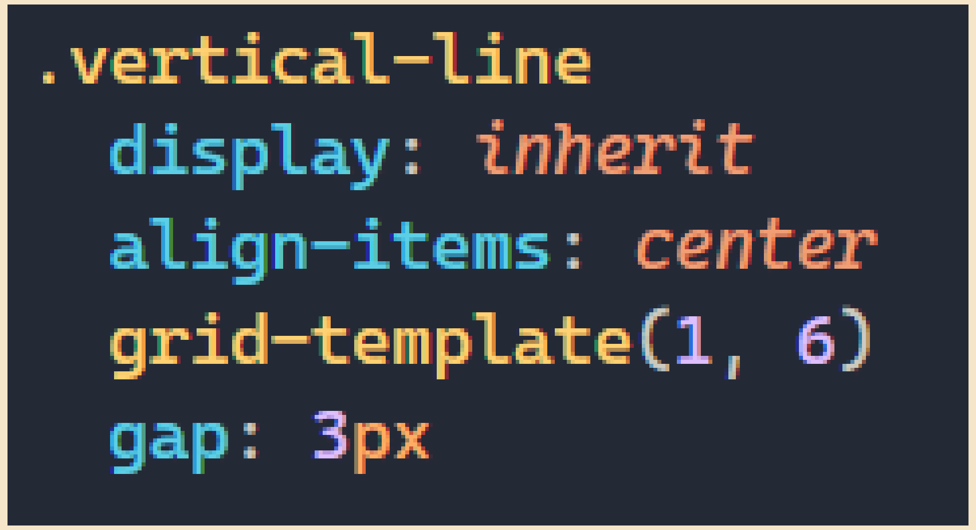


```
out-card(row-start)
  display: inherit
  grid-row-start: row-start

grid-template-col(columns)
  grid-template-columns: repeat(columns, percentage(1 / columns))

grid-template-row(rows)
  grid-template-rows: repeat(rows, percentage(1 / rows))

grid-template(columns, rows)
  grid-template-col(columns)
  grid-template-row(rows)
```

Selectors



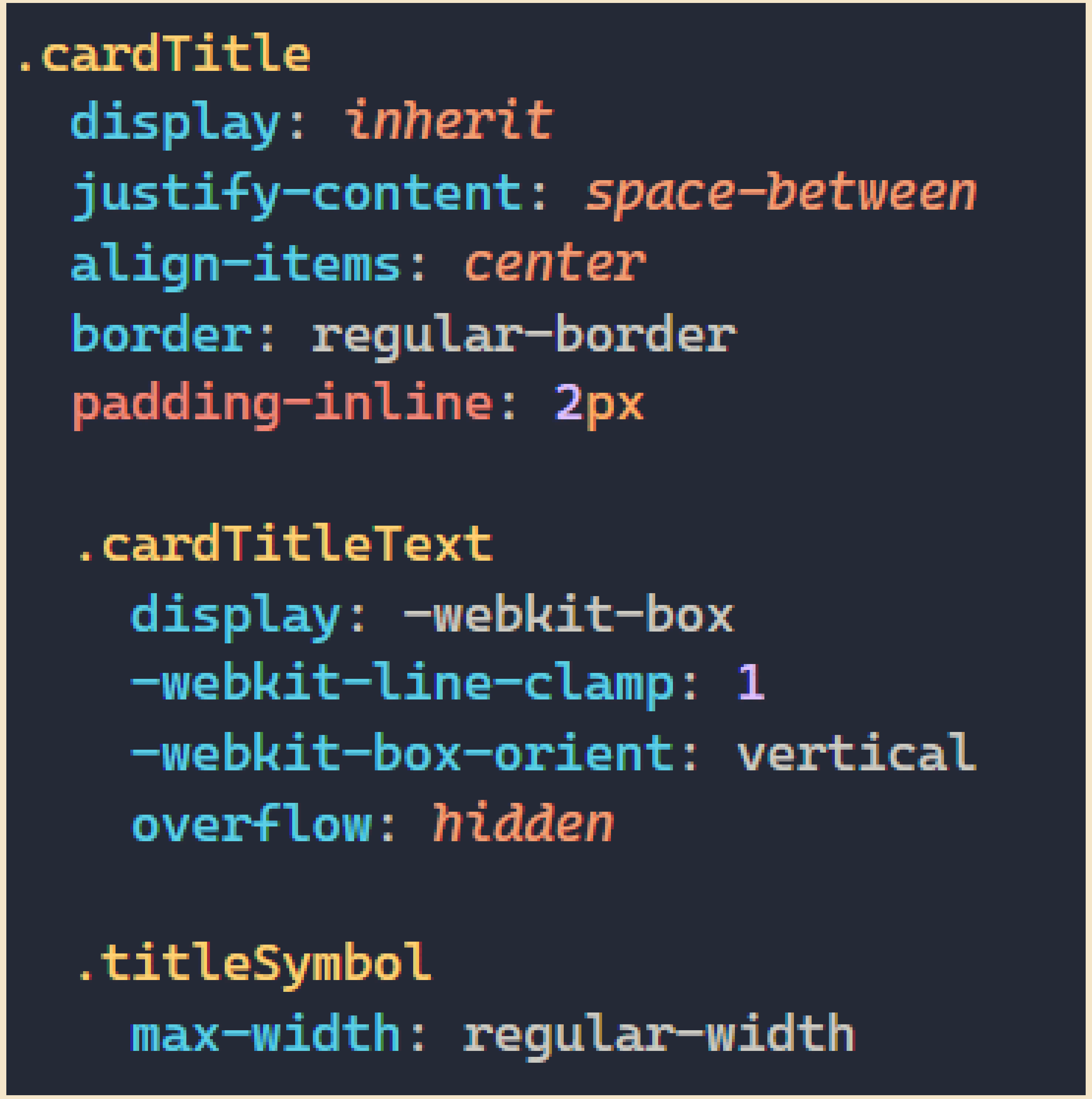
```
.deck
  display: inherit
  margin: default-margin
  grid-template-col(5)
```



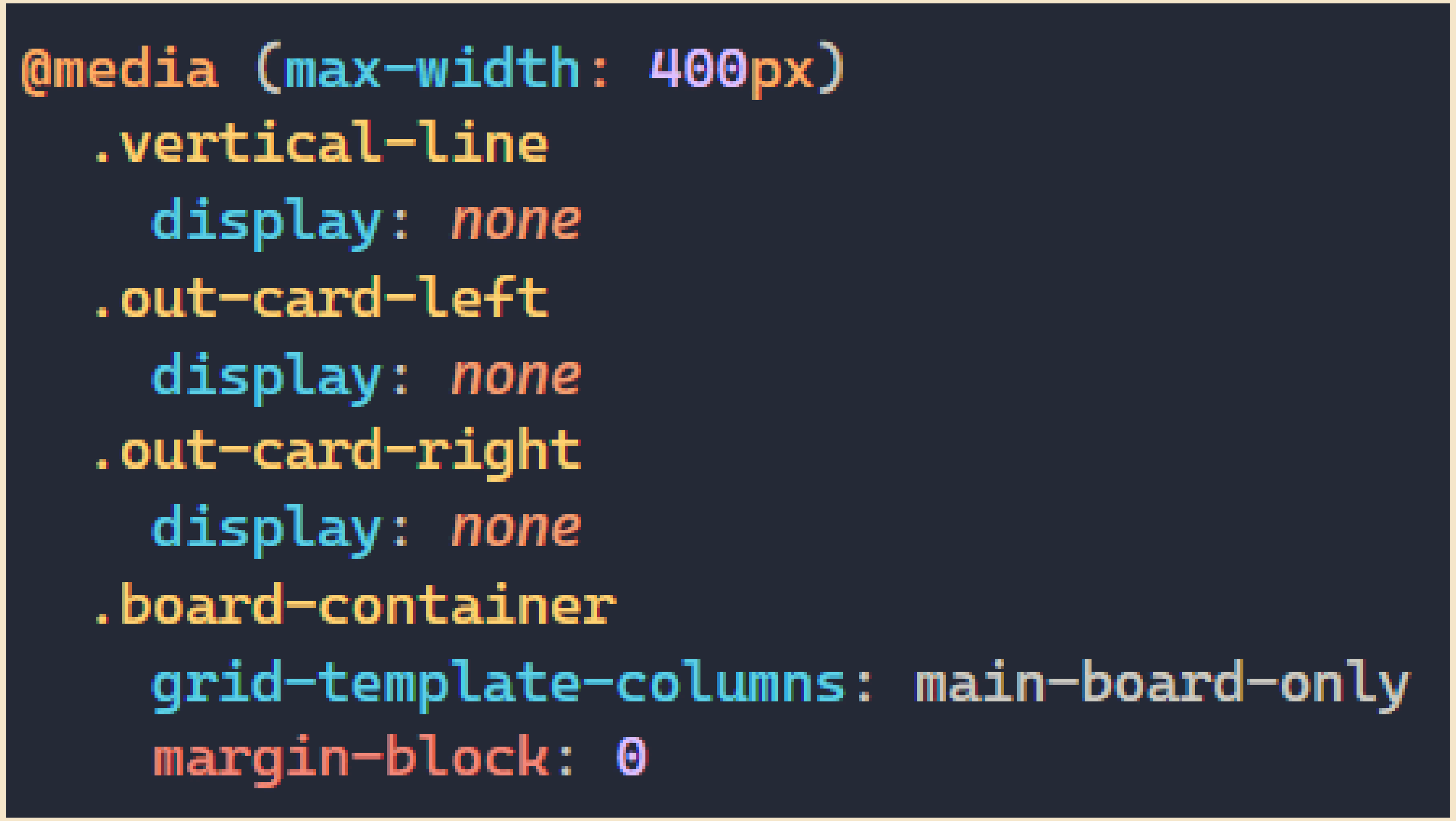
```
.vertical-line
  display: inherit
  align-items: center
  grid-template(1, 6)
  gap: 3px
```

YUGIOH BOARD

Nesting



```
.cardTitle
  display: inherit
  justify-content: space-between
  align-items: center
  border: regular-border
  padding-inline: 2px

  .cardTitleText
    display: -webkit-box
    -webkit-line-clamp: 1
    -webkit-box-orient: vertical
    overflow: hidden

  .titleSymbol
    max-width: regular-width
```



```
@media (max-width: 400px)
  .vertical-line
    display: none
  .out-card-left
    display: none
  .out-card-right
    display: none
  .board-container
    grid-template-columns: main-board-only
    margin-block: 0
```