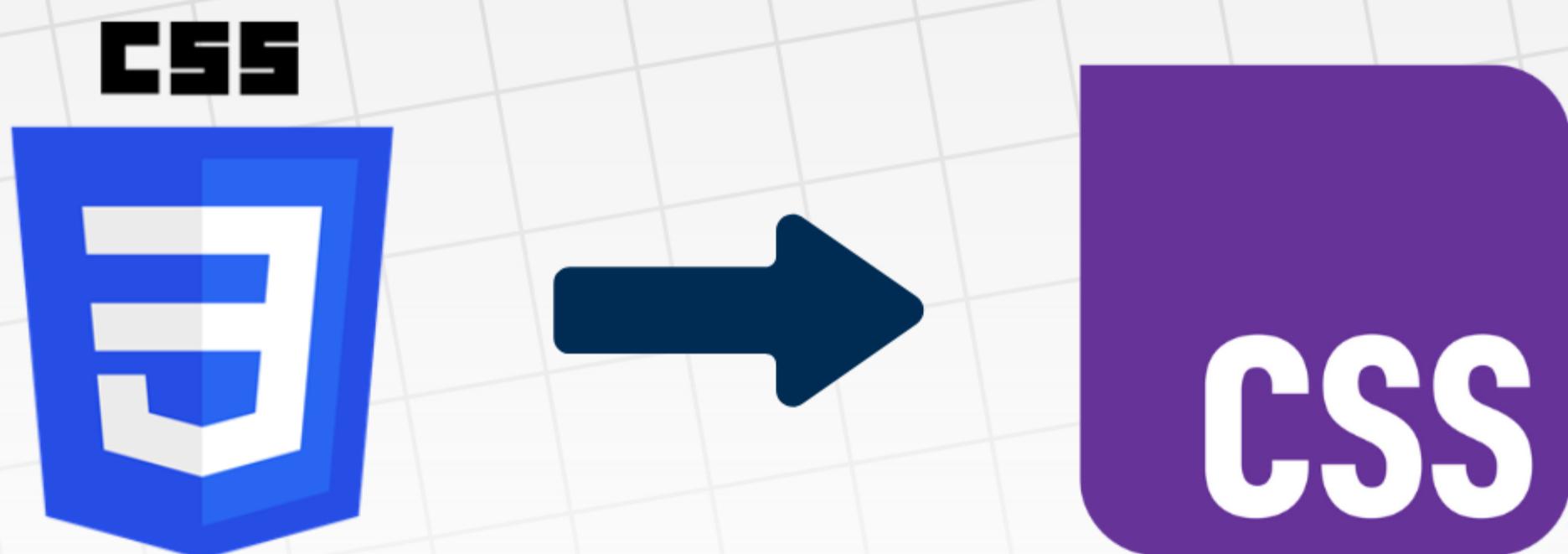


Amir Ben Abdallah



# 7 NEW CSS FEATURES YOU NEED TO KNOW

DISCOVER THE FUTURE OF WEB DESIGN



Amir Ben Abdallah



# LIGHT-DARK MODE: DYNAMIC THEMING FOR ALL USERS

Switching between light and dark themes has never been simpler! Use the **light-dark()** function to effortlessly adapt your design to user preferences.



```
body {  
  background-color: light-dark(white, black);  
  color: light-dark(black, white);  
}
```

2

## NEW CSS PSEUDO-CLASSES : :USER-VALID & :USER-INVALID

Provide real-time feedback to users without overwhelming them with initial warnings. These **pseudo-classes** activate after user interaction, making forms more intuitive and user-friendly.



```
input:user-valid {  
border-color: green;  
}
```

```
input:user-invalid {  
border-color: red;  
}
```



## EFFORTLESS ANIMATIONS WITH THE INTERPOLATE-SIZE PROPERTY

Smoothly animate properties like height or padding without complex setups. The **interpolate-size** property simplifies transitions, making your designs more dynamic and polished.



```
panel {  
  interpolate-size: height 0.5s ease;  
}  
  
panel.open {  
  height: auto;  
}
```

Amir Ben Abdallah



4

# CENTER ELEMENTS THE EASY WAY WITH ALIGN-CONTENT

Centering elements in block layouts just got easier!

The **align-content** property reduces the need for Flexbox or Grid, making your CSS cleaner and more intuitive.



```
my-element {  
  display: block;  
  align-content: center;  
}
```

5

# LEVEL UP YOUR CSS WITH THE `@PROPERTY` AT-RULE

The `@property` at-rule brings TypeScript-like functionality to CSS, allowing you to define syntax, inheritance, and default values for dynamic, type-safe styling.



```
@property --rotation {  
    syntax: '<angle>';  
    inherits: false;  
    initial-value: 0deg;  
}  
  
div {  
    transform: rotate(var(--rotation));  
}
```

## 6

# MASTER SMOOTH TRANSITIONS WITH @STARTING-STYLE

The **@starting-style** rule is a game-changer for defining initial element states, ensuring smooth transitions and eliminating the Flash of Unstyled Content (FOUC).



```
@starting-style {  
  .modal {  
    opacity: 0;  
    visibility: hidden;  
  }  
}  
.  
.modal {  
  opacity: 1;  
  visibility: visible;  
  transition: opacity 0.3s ease, visibility 0.3s ease;  
}
```

7

# SIMPLIFY COMPLEX LAYOUTS WITH ADVANCED CSS MATH

CSS now supports powerful math functions like **round()**, **mod**, and **rem()**, making complex designs easier than ever. Simplify your calculations and create dynamic layouts with less effort.



```
.box {  
  margin: round(2.5px); /* Rounds to 3px */  
}  
.stripe:nth-child(odd) {  
  left: calc(var(--index) * 50px mod 200px);  
}  
.circle {  
  width: rem(10px, 3px); /* Outputs 1px */  
}
```

# I HOPE YOU FOUND THIS HELPFUL !

Let's learn together! Share your insights in the comments



Amir Ben Abdallah