

# Styling a Web Page with CSS

How CSS Is Used to Design Web Pages

# Viewing and Experiencing Web Pages

- Web page may be displayed differently
  - Kind of device, kind of screen
  - Design to accommodate users
- Experience web pages differently
  - Typical computer monitor



# Viewing and Experiencing Web Pages

- Web page may be displayed differently
  - Kind of device, kind of screen
  - Design to accommodate users
- Experience web pages differently
  - Typical computer monitor
  - Mobile device





# Viewing and Experiencing Web Pages

- Web page may be displayed differently
  - Kind of device, kind of screen
  - Design to accommodate users
- Experience web pages differently
  - Typical computer monitor
  - Mobile device
  - Huge display screen



Video monitor by Luis Villa del Camp/CC-by-2.0

# Web Page Accessibility

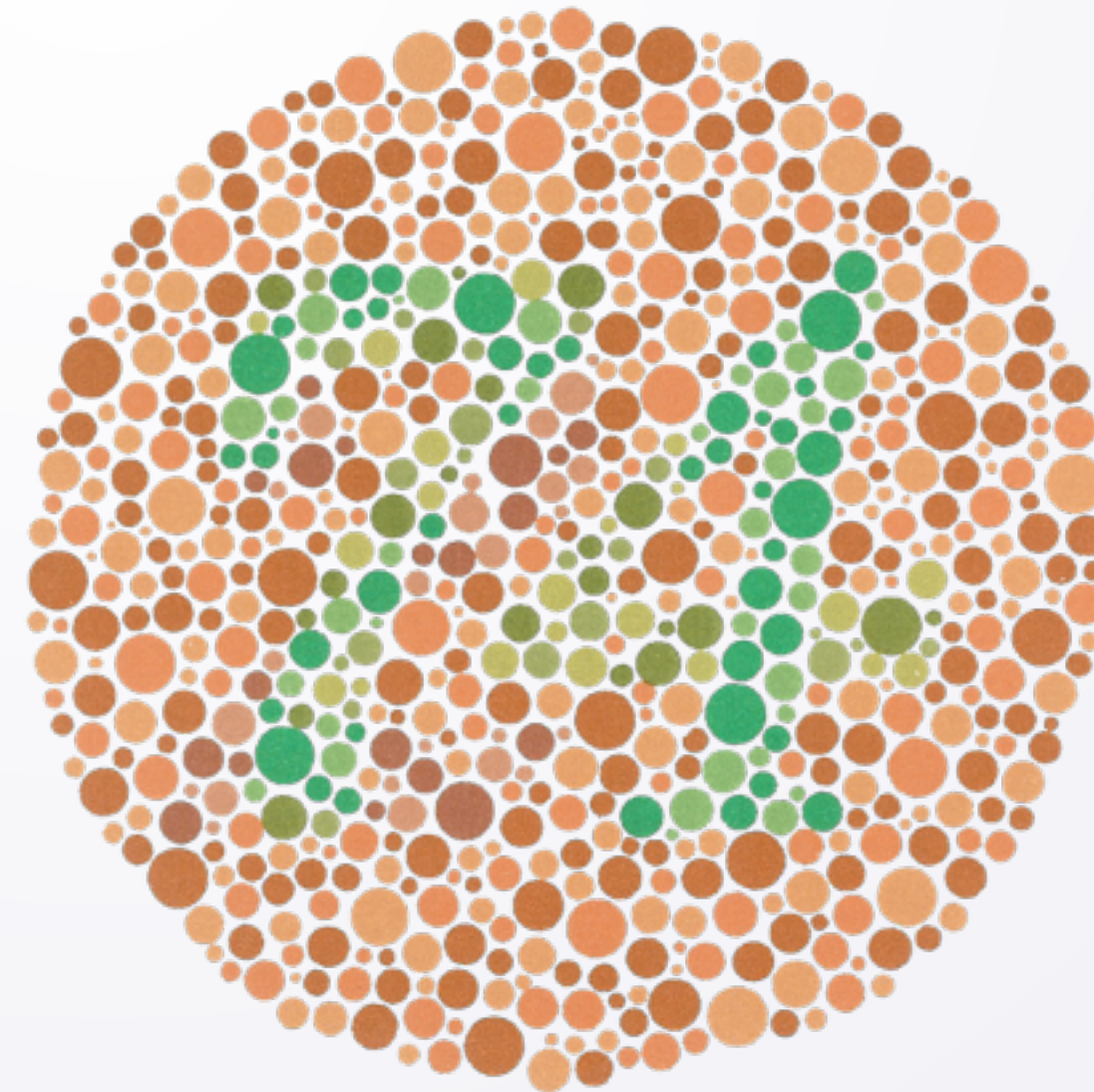
- Work to ensure web pages accessible
  - Remove barriers, help everyone





# Web Page Accessibility

- Work to ensure web pages accessible
  - Remove barriers, help everyone
  - Sight, hearing, color, clicking, more



# Web Page Accessibility

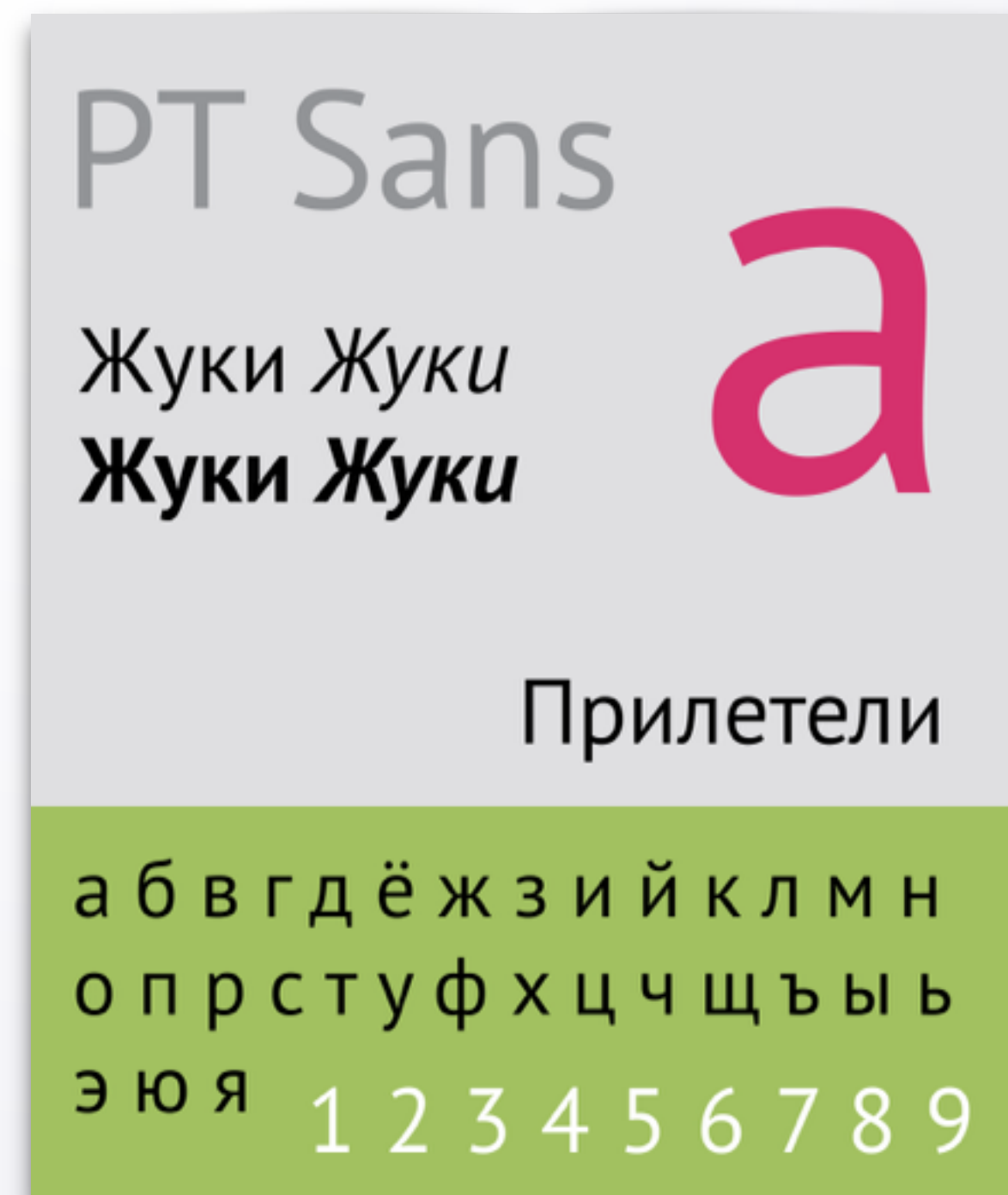
- Work to ensure web pages accessible
  - Remove barriers, help everyone
  - Sight, hearing, color, clicking, more
  - Example: help screen-reading





# Web Page Accessibility

- Work to ensure web pages accessible
  - Remove barriers, help everyone
  - Sight, hearing, color, clicking, more
  - Example: help screen-reading
  - Example: colors and fonts





# Web Page Accessibility

- Work to ensure web pages accessible
  - Remove barriers, help everyone
  - Sight, hearing, color, clicking, more
  - Example: help screen-reading
  - Example: colors and fonts
- Good user experience
  - Pages load quickly



# From HTML Toward CSS

- HTML, Hypertext Markup Language
  - Specifies document content



# From HTML Toward CSS

- HTML, Hypertext Markup Language
  - Specifies document content
  - Specifies some formatting
    - Header: `<h1>` `<h2>` `</h2>` `</h1>`

**h1 heading**

**h2 heading**

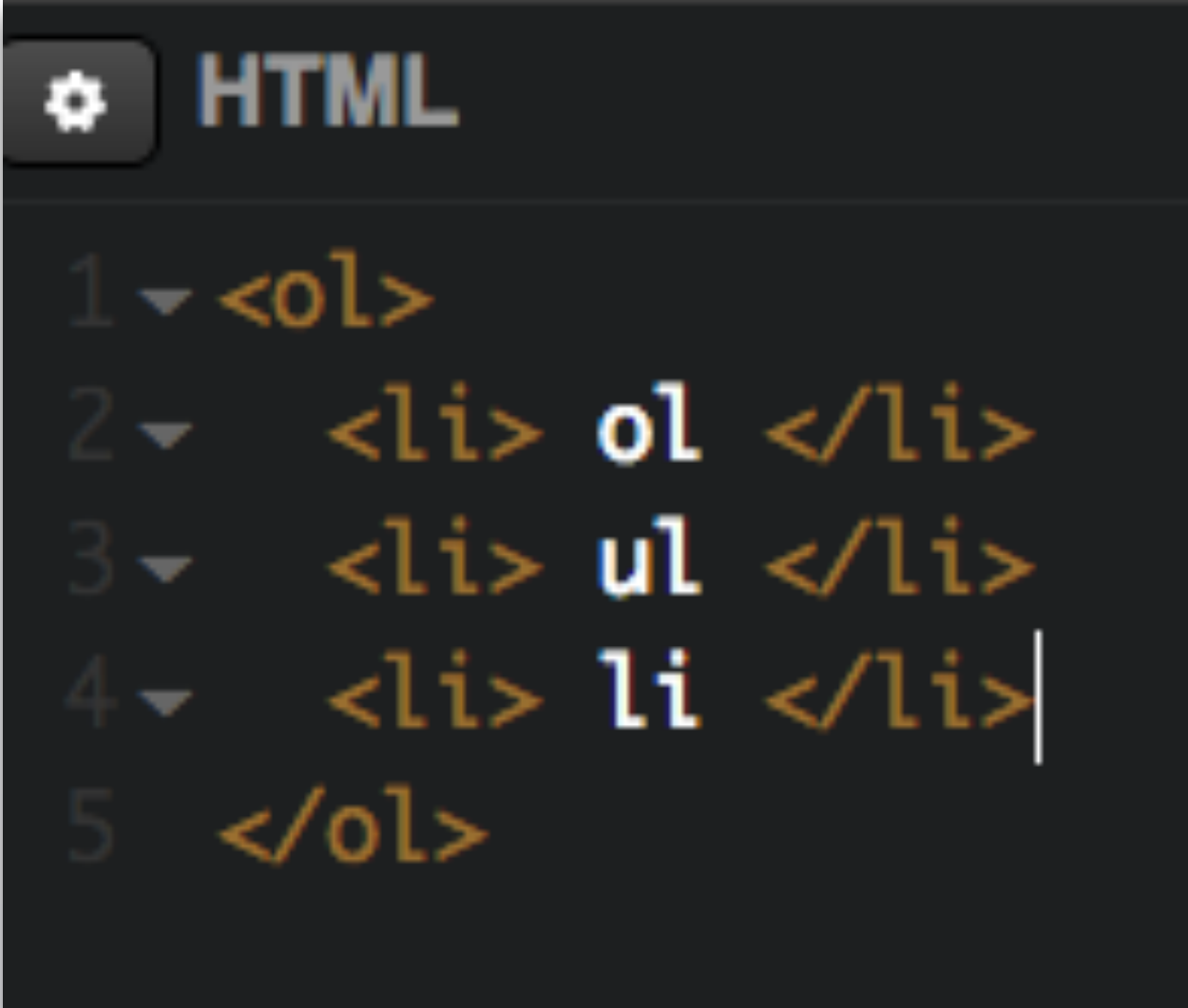
**h3 heading**

**h4 heading**



# From HTML Toward CSS

- HTML, Hypertext Markup Language
  - Specifies document content
  - Specifies some formatting
    - Header: `<h1> <h2> </h2> </h1>`
    - Lists: `<ol></ol>`, `<ul></ul>`




A screenshot of a code editor window titled "HTML" with a gear icon. The editor displays a list structure in HTML code. The code is as follows:

```
1 <ol>
2   <li> ol </li>
3   <li> ul </li>
4   <li> li </li>
5 </ol>
```

# From HTML Toward CSS

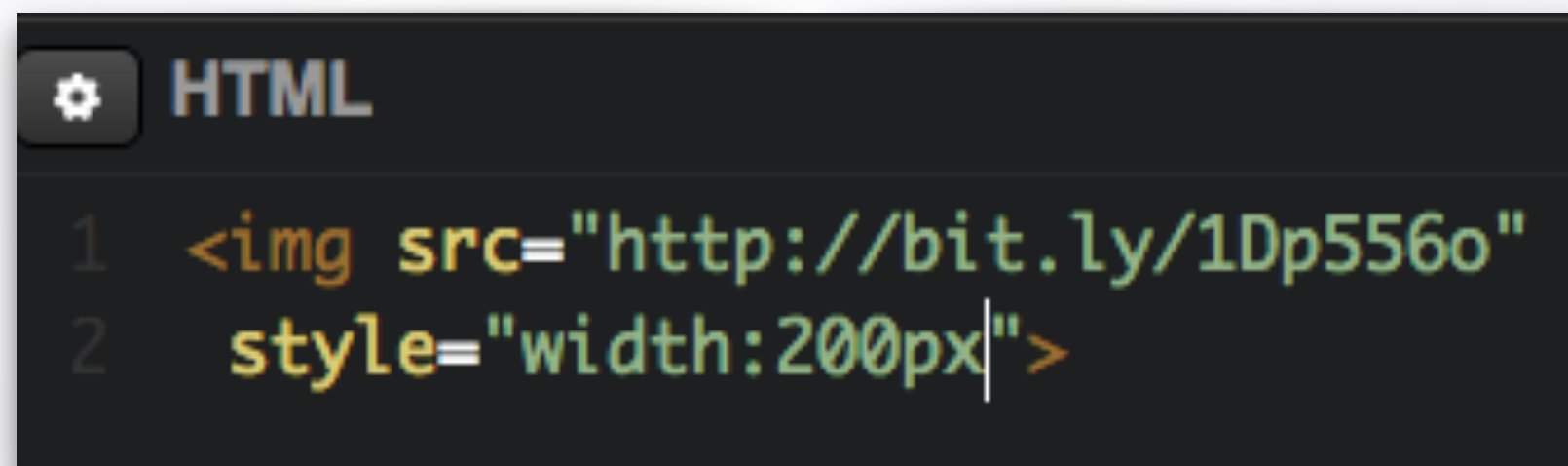
- HTML, Hypertext Markup Language
  - Specifies document content
  - Specifies some formatting
    - Header: `<h1> <h2> </h2> </h1>`
    - Lists: `<ol></ol>`, `<ul></ul>`
    - Tables: `<tr></tr> <th></th> <td></td>`
  - Displaying images
    - `width="200"`

A screenshot of a code editor window titled "HTML" with a gear icon. It shows two lines of code: line 1 is ``.

```
HTML
1 
```

# From HTML Toward CSS

- HTML, Hypertext Markup Language
  - Specifies document content
  - Specifies some formatting
    - Header: `<h1> <h2> </h2> </h1>`
    - Lists: `<ol></ol>`, `<ul></ul>`
    - Tables: `<tr></tr> <th></th> <td></td>`
  - Displaying images
    - `width="200"`
    - `style="width:200px;"`

A code editor window titled "HTML" with a gear icon. It contains two lines of code: line 1 is ``.

```
HTML
1 
```



# CSS

- Cascading Style Sheets
  - Specifies look and formatting
  - Separates content from presentation
    - How big is `<h1>`?
    - Color for `<h1>`?

```
1 <h1>Header</h1>
```



# CSS

- Cascading Style Sheets
  - Specifies look and formatting
  - Separates content from presentation
    - How big is `<h1>`?
    - Color for `<h1>`?
    - Mobile or Desktop ?
- Design that scales
  - 1000 pages on website



# Styling a Web Page with CSS

## CSS Basics

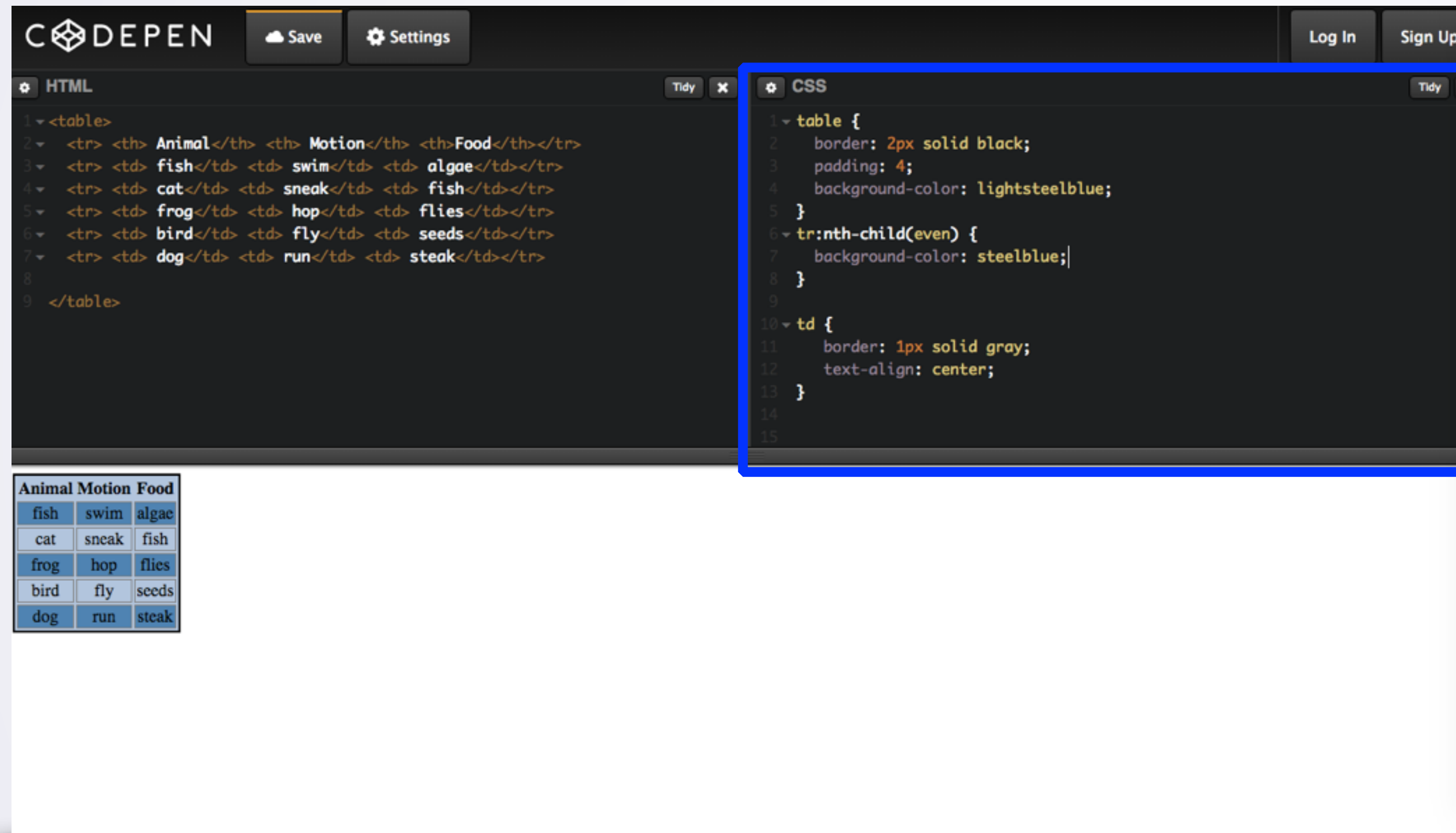


# CSS

- Last time: Why CSS?
  - Reusability
  - Maintainability
- This time: How to write your own

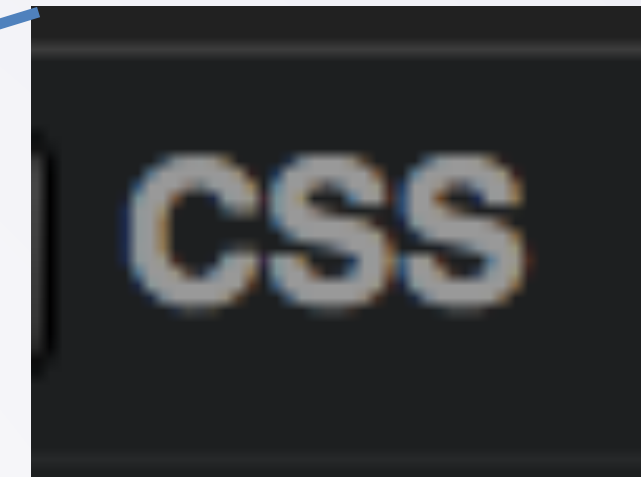
# Where to Write CSS

- CodePen: CSS Pane



# Where to Write CSS

- CodePen: CSS Pane



The screenshot shows the CodePen editor interface. At the top, there's a header with the CodePen logo, 'Save' and 'Settings' buttons, and 'Log In' and 'Sign Up' buttons. Below the header, there are two main panes: 'HTML' on the left and 'CSS' on the right. The 'HTML' pane contains a table structure with headers 'Animal', 'Motion', and 'Food', and five rows of data. The 'CSS' pane contains CSS code that styles the table, including a table border, padding, background color, and specific styles for even-numbered rows and table cells. Below the code panes, a preview of the rendered HTML table is shown, displaying the data with light blue borders and alternating row colors.

```
HTML
1 <table>
2   <tr> <th> Animal</th> <th> Motion</th> <th> Food</th></tr>
3   <tr> <td> fish</td> <td> swim</td> <td> algae</td></tr>
4   <tr> <td> cat</td> <td> sneak</td> <td> fish</td></tr>
5   <tr> <td> frog</td> <td> hop</td> <td> flies</td></tr>
6   <tr> <td> bird</td> <td> fly</td> <td> seeds</td></tr>
7   <tr> <td> dog</td> <td> run</td> <td> steak</td></tr>
8
9 </table>
```

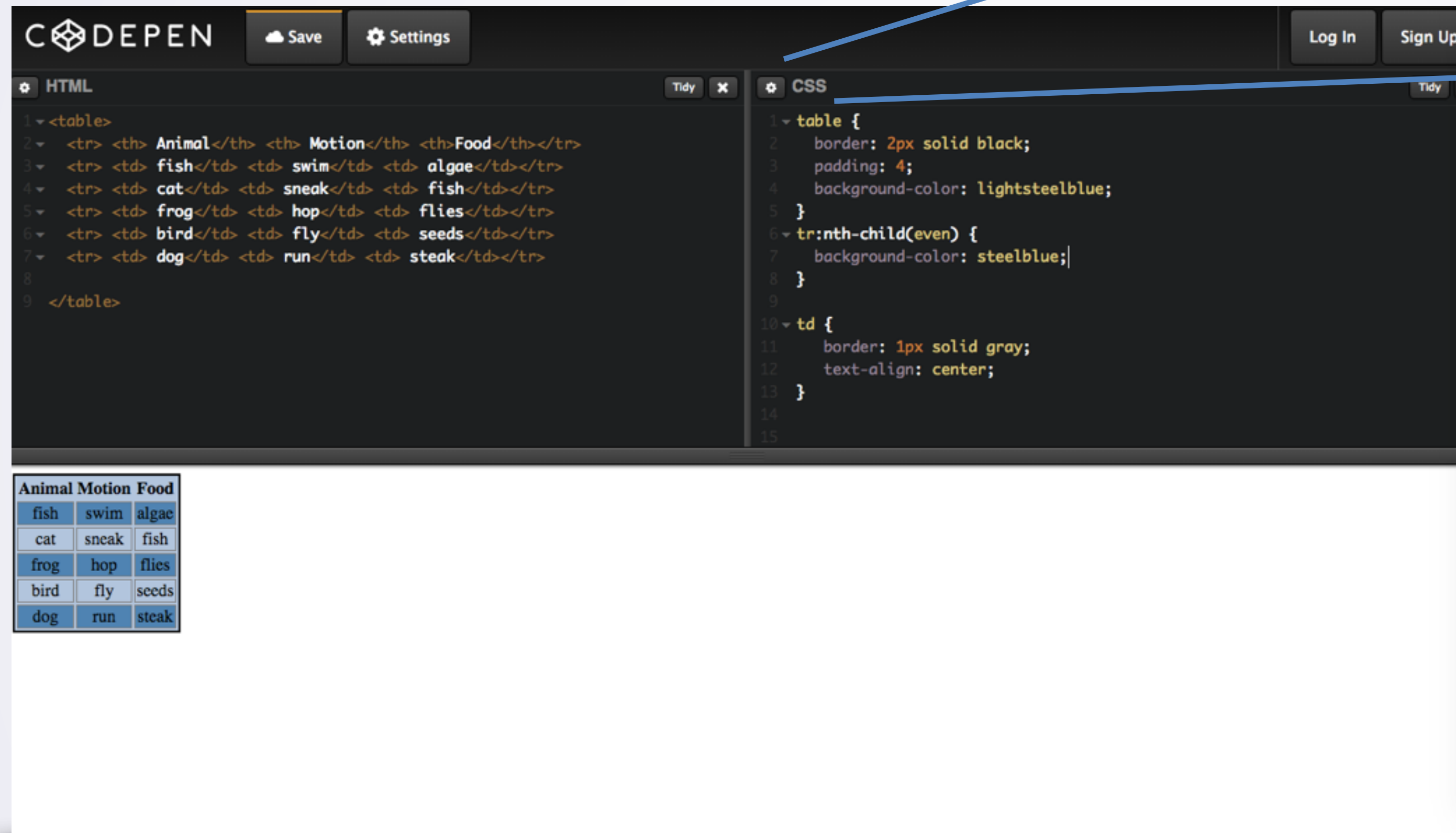
```
CSS
1 table {
2   border: 2px solid black;
3   padding: 4;
4   background-color: lightsteelblue;
5 }
6 tr:nth-child(even) {
7   background-color: steelblue;
8 }
9
10 td {
11   border: 1px solid gray;
12   text-align: center;
13 }
14
15
```

Animal	Motion	Food
fish	swim	algae
cat	sneak	fish
frog	hop	flies
bird	fly	seeds
dog	run	steak



# Where to Write CSS

- CodePen: CSS Pane



- From scratch: `<style>` or `<link>`

# CSS: Change Styles



The screenshot shows the CodePen interface with the HTML editor on the left and the CSS editor on the right. The HTML editor contains the following code:

```
1 <h1>My Favorite Foods</h1>
2 I like a lot of delicious food,
  some of my favorites are
3 <ul>
4   <li> Chocolate
5   <li> Cherries
6   <li> Ice Cream
```

The CSS editor is empty. The preview window at the bottom shows the rendered HTML: a heading "My Favorite Foods" and a paragraph "I like a lot of delicious food, some of my favorites are" followed by a bulleted list of "Chocolate", "Cherries", and "Ice Cream".



The screenshot shows the CodePen interface with the HTML editor on the left and the CSS editor on the right. The HTML editor contains the same code as the previous screenshot. The CSS editor now contains the following code:

```
1 h1 {
2   text-align: center;
3   color: blue;
4 }
```

The preview window at the bottom shows the rendered HTML with the heading "My Favorite Foods" now centered and colored blue. The paragraph and bulleted list remain unchanged.

- Change styles of elements
- Example: `<h1>` blue and centered

# CSS Syntax

```
h1 {  
    text-align: center;  
    color: blue;  
}
```

- Let's look at the CSS in detail



# CSS Syntax

```
h1 {  
    text-align: center;  
    color: blue;  
}
```

Selector: what element to format

- Let's look at the CSS in detail

# CSS Syntax

```
h1 {  
  text-align: center;  
  color: blue;  
}
```

Curly braces around formatting for that element

- Let's look at the CSS in detail

# CSS Syntax


```
h1 {  
    text-align: center;  
    color: blue;  
}
```

Property to change

- Let's look at the CSS in detail

# CSS Syntax

```
h1 {  
  text-align: center;  
  color: blue;  
}
```

A blue circle highlights the colon in the 'text-align: center;' line. A horizontal blue line extends from the circle to the right, ending under the word 'Colon'.

- Let's look at the CSS in detail



# CSS Syntax

```
h1 {  
  text-align: center;  
  color: blue;  
}
```

Value

- Let's look at the CSS in detail

# CSS Syntax

```
h1 {  
    text-align: center;  
    color: blue;  
}
```

Semicolon

- Let's look at the CSS in detail

# CSS Syntax

```
h1 {  
    text-align: center;  
    color: blue;  
}
```

- Let's look at the CSS in detail
- Many properties

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

# CSS: Select Some Elements

```
li {  
  color: green;  
}
```



# CSS: Select Some Elements

```
li {  
    color: green;  
}
```

- Makes **all** <li>s green
- What if we want some green?
- Three approaches

# CSS: Classes

- Classes: named styles

- HTML

```
<li class="foodLi"> Chocolate </li>
```

```
<li class="foodLi"> Cherries </li>
```

```
<li class="foodLi"> Ice Cream </li>
```

- CSS:

```
.foodLi {  
    color : green;  
}
```

# CSS: Classes

- Classes: named styles

- HTML

```
<li class="foodLi"> Chocolate </li>
```

```
<li class="foodLi"> Cherries </li>
```

```
<li class="foodLi"> Ice Cream </li>
```

- CSS:

 .foodLi { **Dot: Making a class**

```
    color : green;  
}
```

# CSS: Classes

- Classes: named styles

- HTML

```
<li class="foodLi"> Chocolate </li>
```

```
<li class="foodLi"> Cherries </li>
```

```
<li class="foodLi"> Ice Cream </li>
```

- CSS:

```
.foodLi { Class name: anything you want  
    color : green;  
}
```



# CSS: Classes

- Classes: named styles

Specify class in HTML

- HTML

```
<li class="foodLi"> Chocolate </li>
```

```
<li class="foodLi"> Cherries </li>
```

```
<li class="foodLi"> Ice Cream </li>
```

- CSS:

```
.foodLi {  
    color : green;  
}
```

# CSS: IDs

- IDs: Name **one** element

- HTML:

```

```

- CSS:

```
#cakeImg {  
    float:right;  
}
```

# CSS: Combinators

- Combinators: Select by relationship
  - Style <li> inside of <ul>:
    - `ul li { ... }`
  - More advanced relationships exist

# Classes and IDs: Name and Reuse

- Class and ID: name style
  - Re-use as needed
  - Many elements in one page
  - Across pages
  - Name + use: recurring concept in CS



# CSS Basics

- Writing CSS in CodePen
- CSS Syntax
  - `selector { property : value ; }`
- Classes and IDs
  - Name and re-use

# Styling a Web Page with CSS

Colors and Names in CSS

# CSS Colors

- Last time: Basics of CSS
- This time: Colors
  - Names and numbers

# CSS Syntax

```
h1 {  
    text-align: center;  
    color: blue;  
}
```

- Last time:
  - `<h1>` blue and centered
  - Other colors?



# 140 Color Names

BlueViolet



Gold



LightSteelBlue



FireBrick



- 140 Standard color names
  - Remember them all?
  - <http://www.w3.org/TR/2003/CR-css3-color-20030514/#svg-color>
  - What about other colors?

# Millions of Colors?



Duke Blue

- Want more than 140 colors
- Average Human: Millions of Colors
- Example: 12 shades of blue

# Millions of Colors

- How to handle millions of colors?
  - Names?
  - Limit selection?
  - Numbers?
    - Everything is a number
    - Red + Green + Blue, 0–255
    - Can specify 16 million colors

# Color Numbers

BlueViolet

`rgb(138, 43, 226)`

Gold

`rgb(255, 215, 0)`

LightSteelBlue

`rgb(179, 196, 222)`

FireBrick

`rgb(178, 34, 34)`

- Can specify with `rgb` (red, green, blue)
  - Makes numerical value from R, G, & B



# Color Numbers

BlueViolet

#8A2BE2

Gold

#FFD700

LightSteelBlue

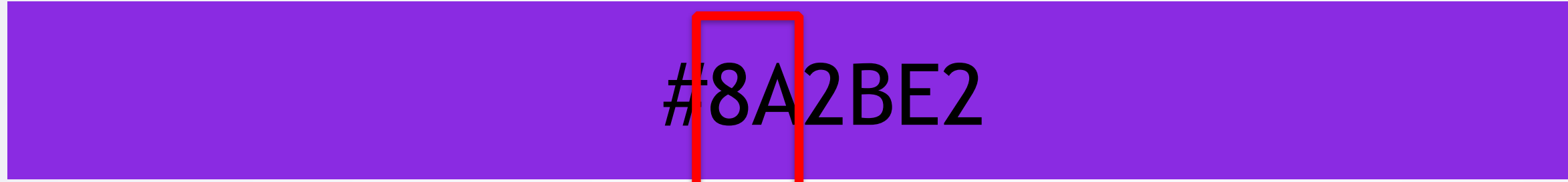

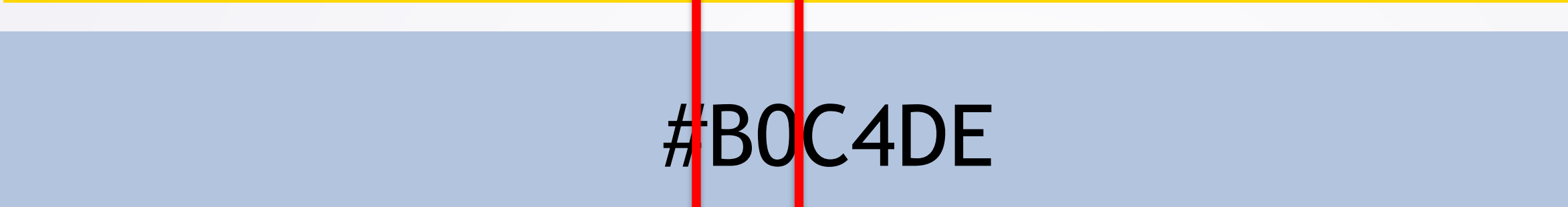

#B0C4DE

FireBrick

#B22222

- Or entire number: # + 6 **hex** digits
  - Hexadecimal: base 16
  - Digits: 0–9, A (10)—F(15)

# Color Numbers

BlueViolet	 #8A2BE2
Gold	 #FFD700
LightSteelBlue	 #B0C4DE
FireBrick	 #B22222

- Or entire number: # + 6 **hex** digits
  - Hexadecimal: base 16
  - Digits: 0–9, A (10)—F(15)

# Color Numbers

BlueViolet	#8A2BE2
Gold	#FFD700
LightSteelBlue	#B0C4DE
FireBrick	#B22222

- Or entire number: # + 6 **hex** digits
  - Hexadecimal: base 16
  - Digits: 0–9, A (10)—F(15)

# Color Numbers

BlueViolet	#8A2BE2
Gold	#FFD700
LightSteelBlue	#B0C4DE
FireBrick	#B22222

- Or entire number: # + 6 **hex** digits
  - Hexadecimal: base 16
  - Digits: 0–9, A (10)—F(15)



# Color Numbers

BlueViolet

#8A2BE2

- Hexadecimal values:
  - Each place is 16x previous
  - 1's, 16's, 256's, ... places

# Color Numbers

BlueViolet

#8A2BE2

$$8 * 16 + 10 * 1 = 138$$

- Hexadecimal values:
  - Each place is 16x previous
  - 1's, 16's, 256's, ... places

# Color Numbers

BlueViolet

#8A2BE2

$$8 * 16 + 10 * 1 = 138$$

$$2 * 16 + 11 * 1 = 43$$

- Hexadecimal values:
  - Each place is 16x previous
  - 1's, 16's, 256's, ... places

# Color Numbers

BlueViolet

#8A2BE2

$$8 * 16 + 10 * 1 = 138$$

$$14 * 16 + 2 * 1 = 226$$

$$2 * 16 + 11 * 1 = 43$$

- Hexadecimal values:
  - Each place is 16x previous
  - 1's, 16's, 256's, ... places

# Color Picker Tools

- May use color picker tool
  - Let us see an example



# CSS: Colors

- 140 Standard Names
- Specify by number
  - `rgb(red, green, blue)`
  - #hex value
  - Color picker:
    - [https://developer.mozilla.org/en-US/docs/Web/CSS/Tools/ColorPicker\\_Tool](https://developer.mozilla.org/en-US/docs/Web/CSS/Tools/ColorPicker_Tool)