

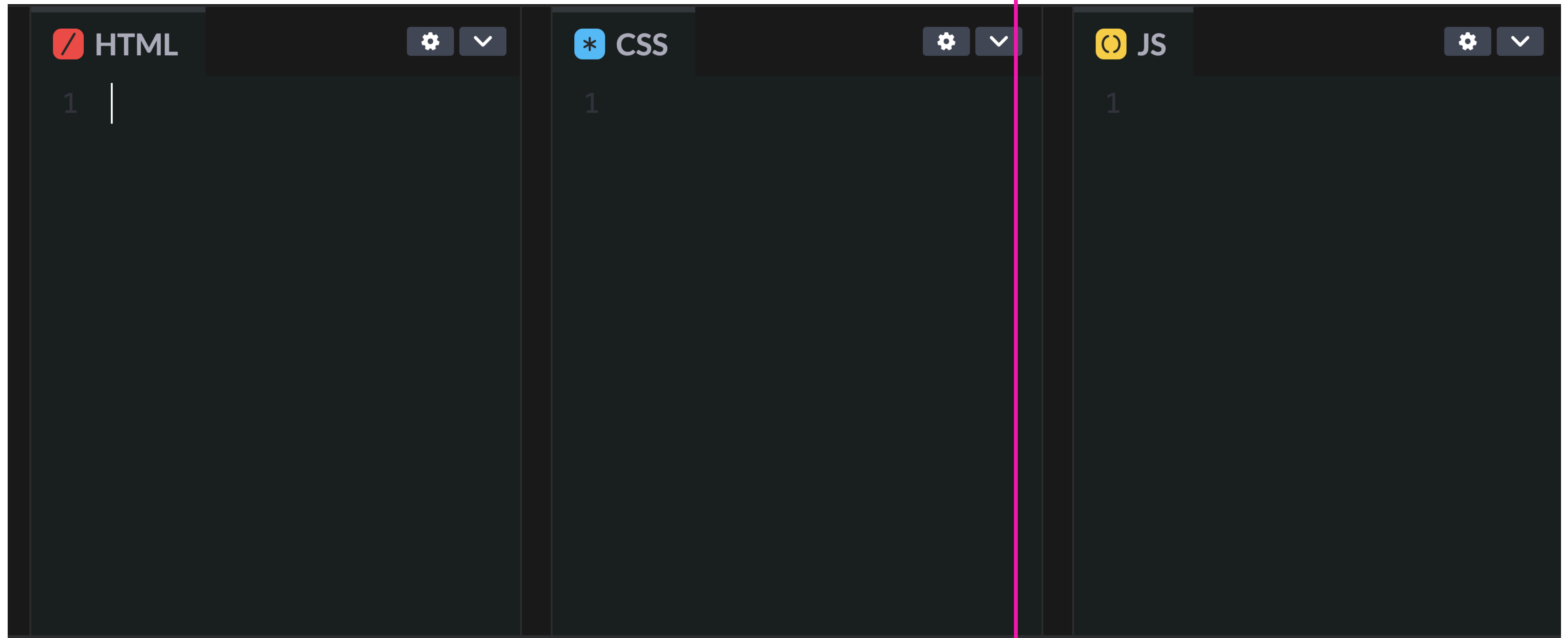
JavaScript

HTML content (bones)

CSS style (skin & clothes)

JS behavior (brain & muscles)

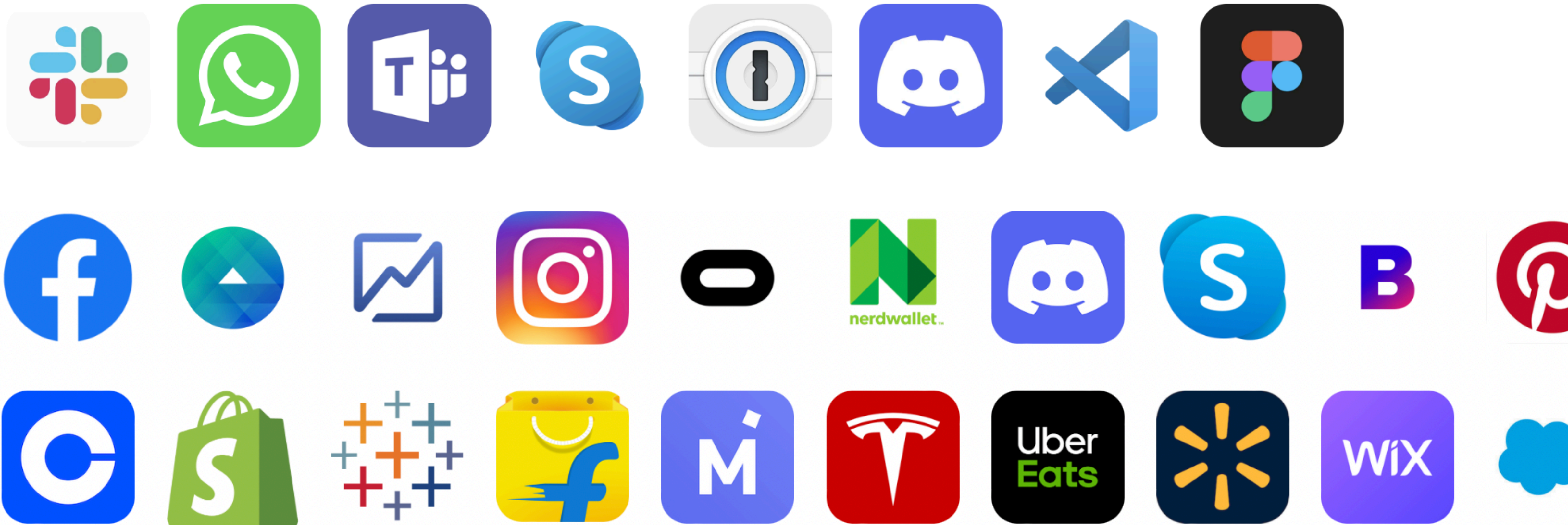
The third language!



JavaScript is the language for adding interactivity to a web page, based on **user actions or events** from the user (or other conditions, like a given amount of time since load) .

It is the only language that can change
webpage content without requiring a
refresh.

And it is everywhere



There are many flavors

Data Types & the Console

Here are some primitive data types.

1. number
2. string
3. boolean
4. null

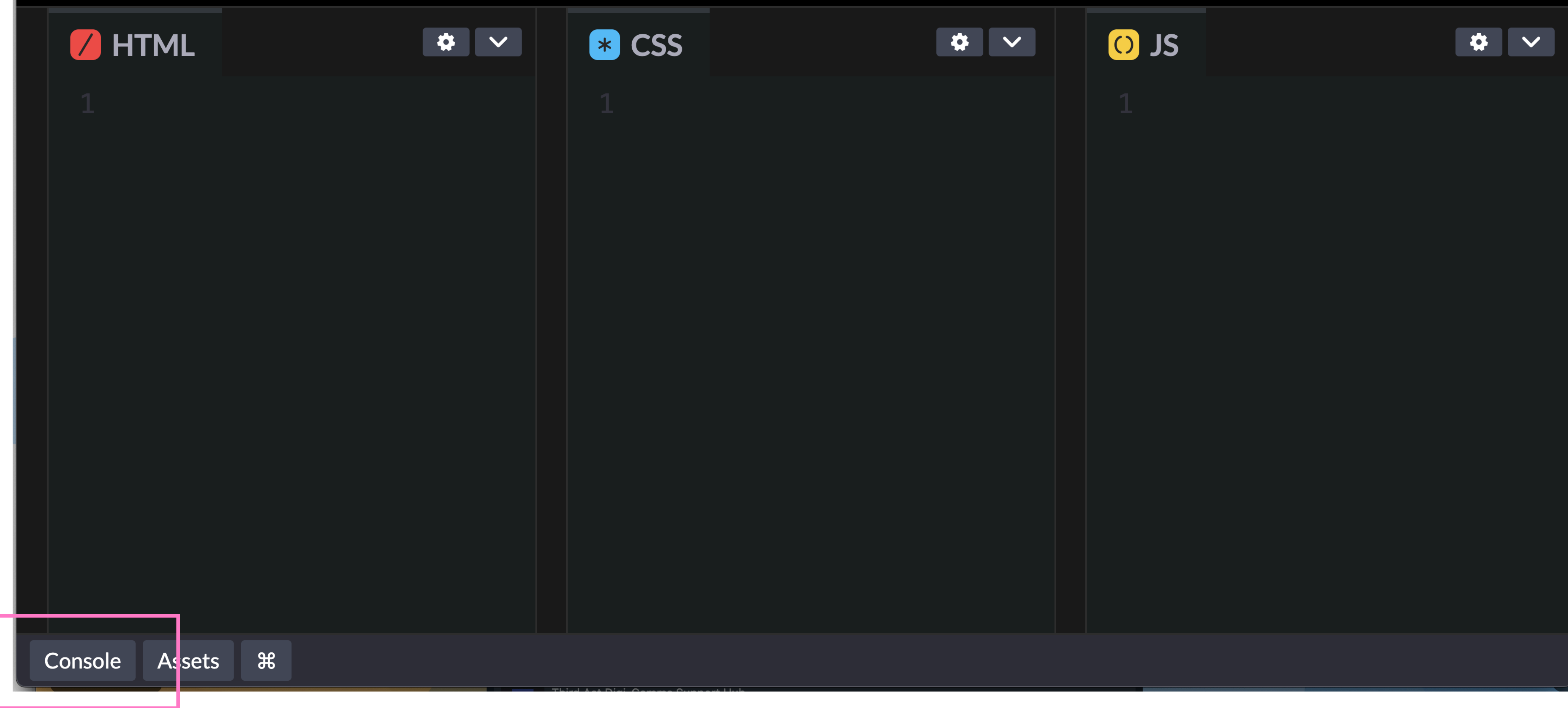
4, 5.666, -8

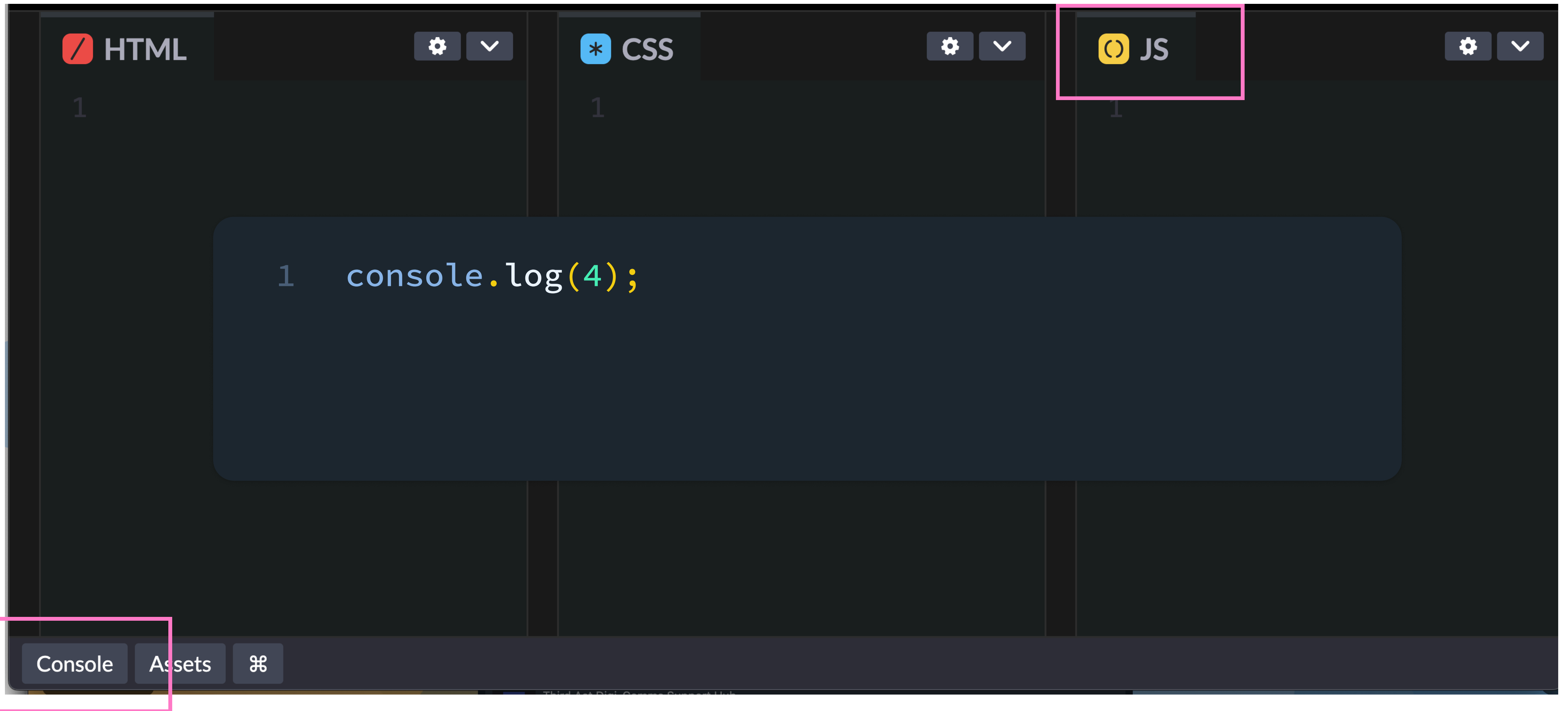
“Interaction”, “hello”

true, false

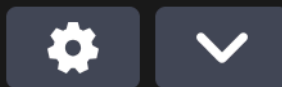
null (empty)

...and more complex ones we'll get to





 HTML



 CSS



 JS



1

1

1

```
1 console.log(4+4);
```

Console

Assets



 HTML

1

 CSS

1

 JS

1

```
1 console.log(4+4 === 10);  
2 // three equal signs is strict equality
```

Console

Assets



 HTML

1

 CSS

1

 JS

1

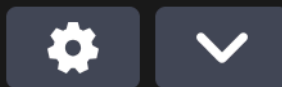
```
1 console.log(4 === 4); // true
2 console.log("4" === 4); // false
3 console.log("4" == 4); // true
4 // three equal signs is strict equality
```

Console

Assets



 HTML



 CSS



 JS



1

1

1

```
1 console.log(typeof(4)); // number  
2
```

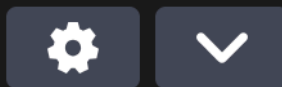
Console

Assets



Defining Variables

 HTML



 CSS



 JS



1

1

1

```
1  const name = "Divya";  
2  let name = "Divya";  
3  var name = "Divya"; // deprecated  
4  
5
```

Console

Assets



 HTML



 CSS



 JS



1

1

1

```
1  const name = "Divya";  
2  console.log(name === "Divya");  
3  
4
```

Console

Assets



 HTML

1

 CSS

1

 JS

1

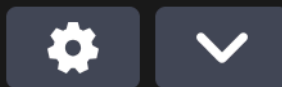
```
1  const name = "Divya";  
2  name = "Div"; // const is binding so this will not  
   work  
3  
4
```

Console

Assets



 HTML



 CSS



 JS



1

1

1

```
1  let name = "Divya";  
2  name = "Div"; // works  
3  
4
```

Console

Assets



A Preview

What does it look like?

```
1  const flavor = "vanilla";
2
3  const showFlavor = (fl) => console.log(fl);
4
5  div.addEventListener("click", showFlavor);
6
7  showFlavor(flavor);
8
9  // vanilla
10
11
```

What does it look like?

```
1  const flavor = "vanilla";  
2  
3  const showFlavor = (fl) => console.log(fl);  
4  
5  div.addEventListener("click", showFlavor);  
6  
7  showFlavor(flavor);  
8  
9  // vanilla  
10  
11
```

defining a variable

What does it look like?

```
1  const flavor = "vanilla";  
2  
3  const showFlavor = (fl) => console.log(fl);  
4  
5  div.addEventListener("click", showFlavor);  
6  
7  showFlavor(flavor);  
8  
9  // vanilla  
10  
11
```

defining a variable

What does it look like?

```
1  const flavor = "vanilla";
2
3  const showFlavor = (fl) => console.log(fl);
4
5  div.addEventListener("click", showFlavor);
6
7  showFlavor(flavor);
8
9  // vanilla
```

defining a variable

defining a function
logging to the console

What does it look like?

```
1  const flavor = "vanilla";
2
3  const showFlavor = (fl) => console.log(fl);
4
5  div.addEventListener("click", showFlavor);
6
7  showFlavor(flavor);
8
9  // vanilla
```

1

0

1

1

defining a variable

defining a function
logging to the console

listen for an event

What does it look like?

```
1  const flavor = "vanilla";
2
3  const showFlavor = (fl) => console.log(fl);
4
5  div.addEventListener("click", showFlavor);
6
7  showFlavor(flavor);
8
9  // vanilla
```

1

0

1

1

The diagram consists of a dark blue code editor on the left and a white background on the right. Four horizontal pink lines connect specific parts of the code to their descriptions:

- A line from the `const flavor = "vanilla";` line to the text "defining a variable".
- A line from the `=>` arrow in the `const showFlavor = (fl) => console.log(fl);` line to the text "defining a function".
- A line from the `showFlavor` argument in the `div.addEventListener("click", showFlavor);` line to the text "logging to the console".
- A line from the `showFlavor(flavor);` line to the text "listen for an event".
- A line from the `showFlavor(flavor);` line to the text "call a function".

What does it look like?

```
1  const flavor = "vanilla";
2
3  const showFlavor = (fl) => console.log(fl);
4
5  div.addEventListener("click", showFlavor);
6
7  showFlavor(flavor);
8
9  // vanilla
```

defining a variable

defining a function
logging to the console

listen for an event

call a function

comment

1
0
1
1

What does it look like?

```
1  const flavor = "vanilla";
2
3  const showFlavor = (fl) => console.log(fl);
4
5  div.addEventListener("click", showFlavor);
6
7  showFlavor(flavor);
8
9  // vanilla
10
11
11
```


What does it look like?

```
1  const flavor = "vanilla";
2
3  const showFlavor = (fl) => console.log(fl);
4
5  div.addEventListener("click", showFlavor);
6
7  showFlavor(flavor);
8
9  // vanilla
```

1
0
1
1