

HTML

```
<body>  
<div>  
  <img>  
  <a> Click! </a>  
</div>  
</body>
```

OUTPUT

body



# Web Lab

CS50 for Lawyers

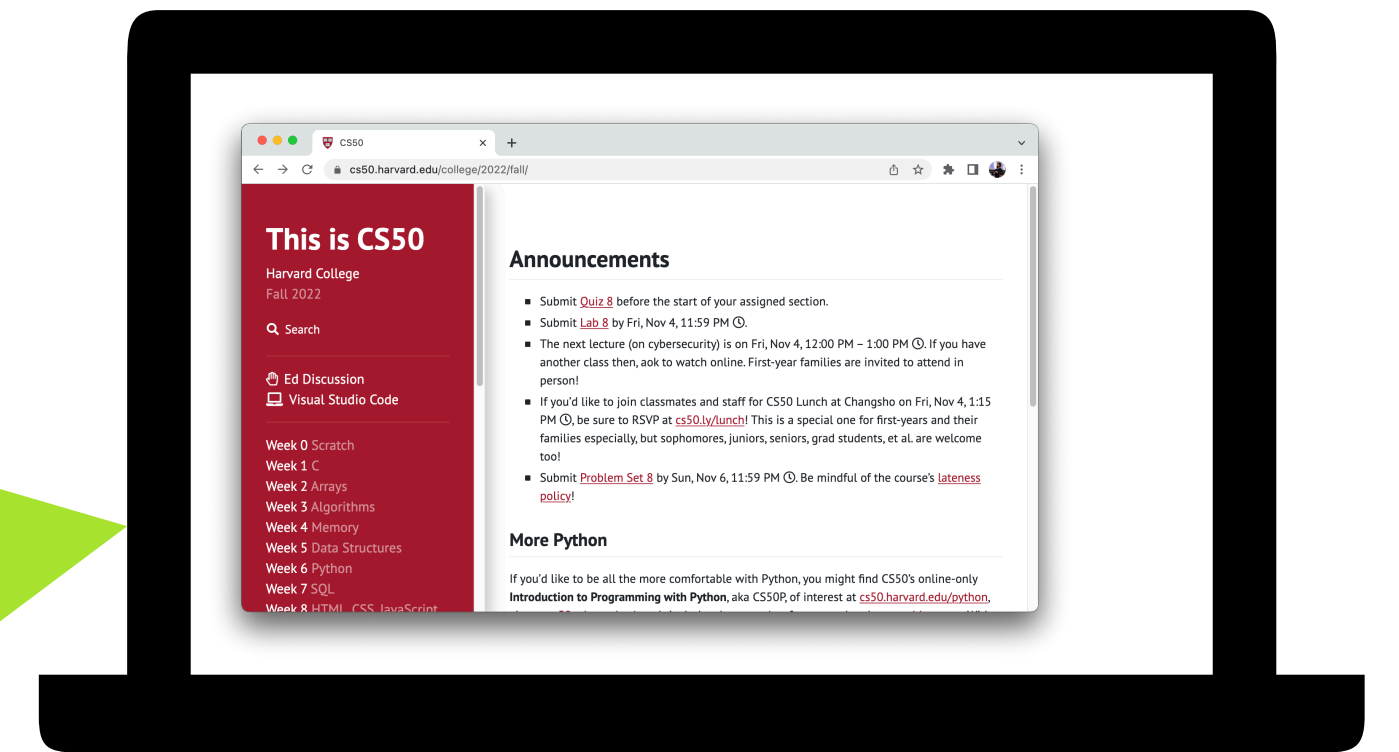
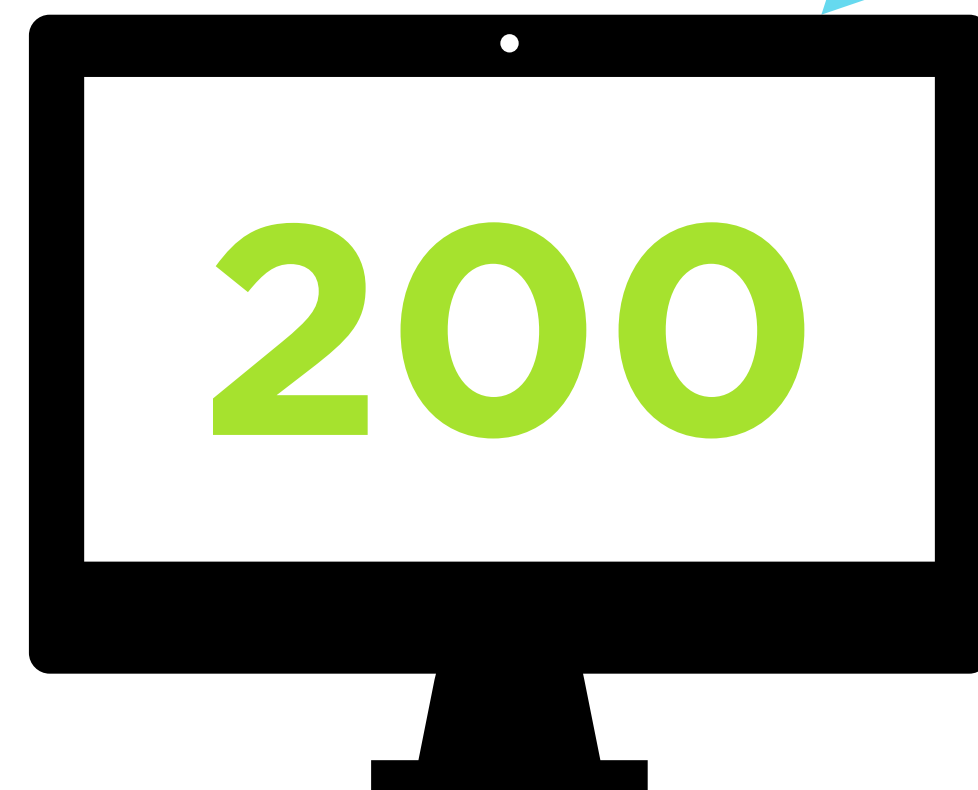
# HTTP

The protocol of the web

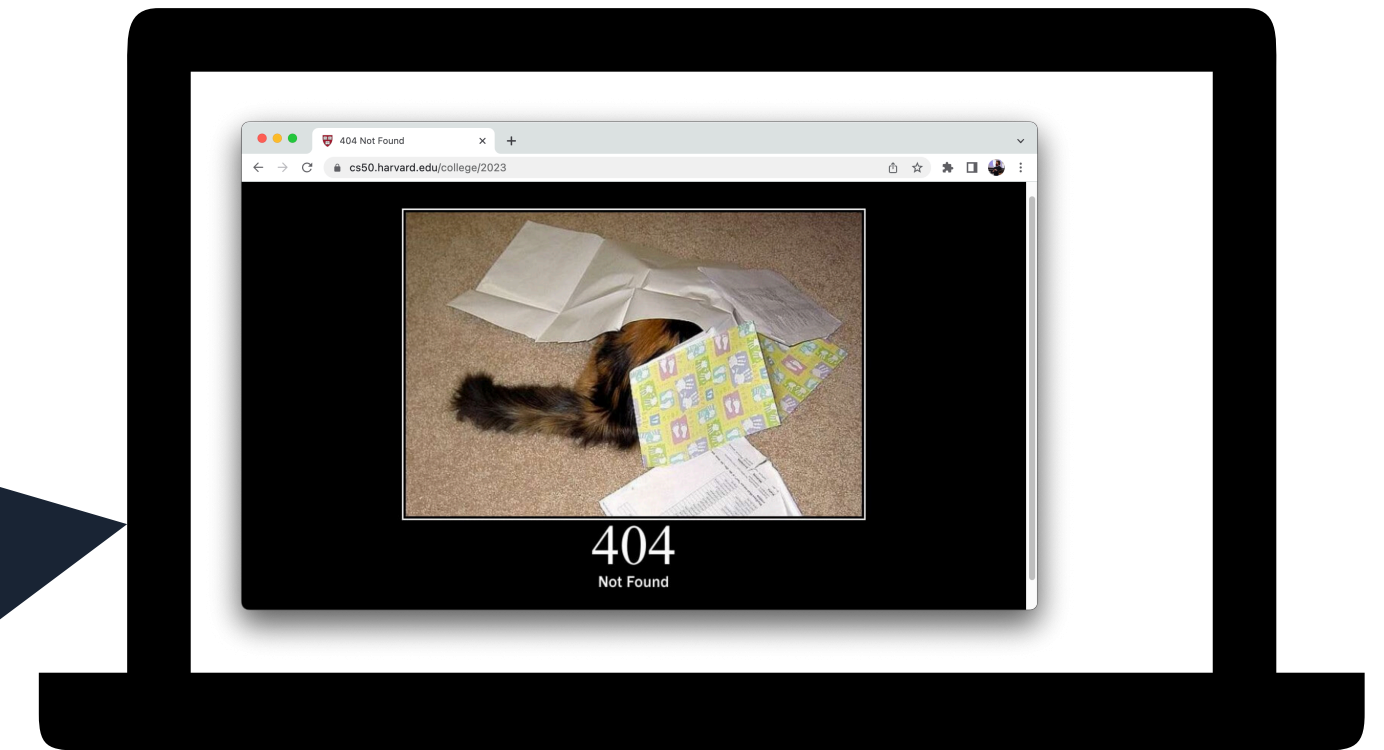
**<http://cs50.harvard.edu/...>**

<http://cs50.harvard.edu/...>

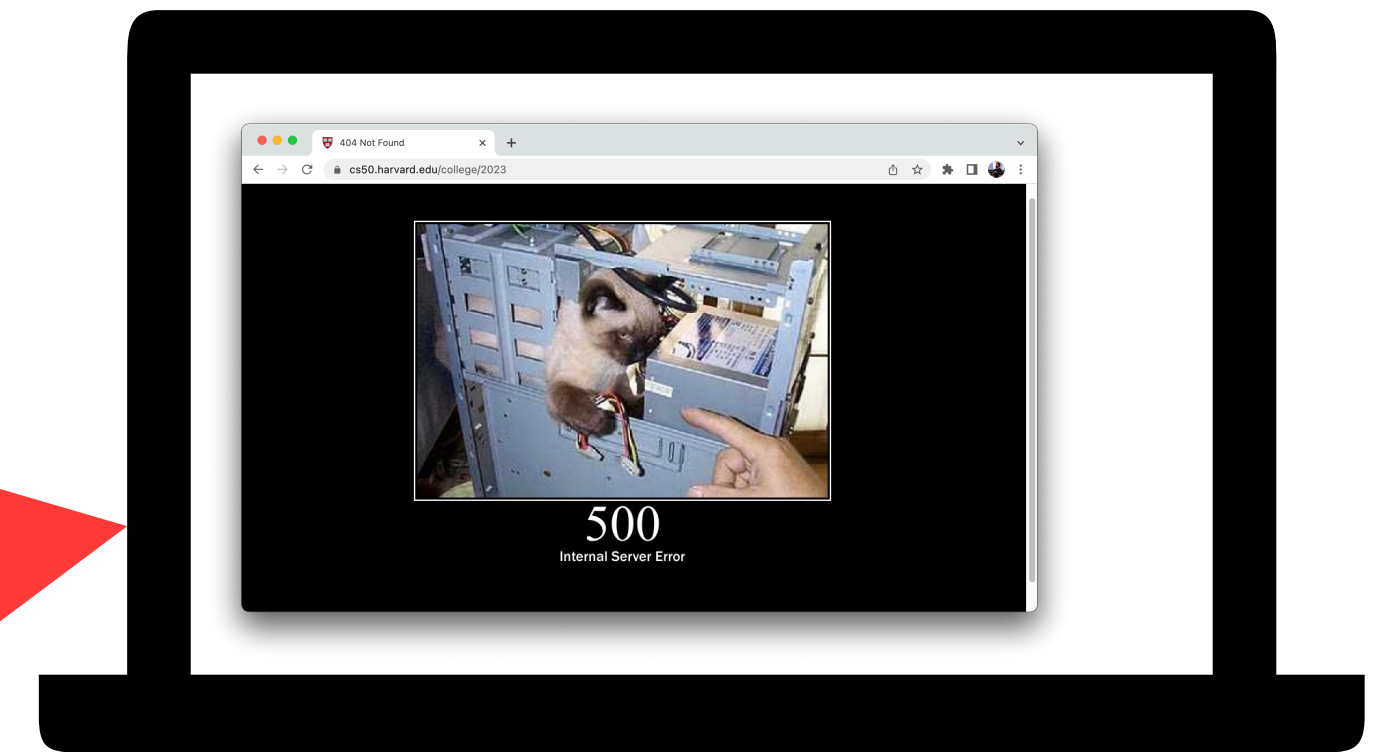
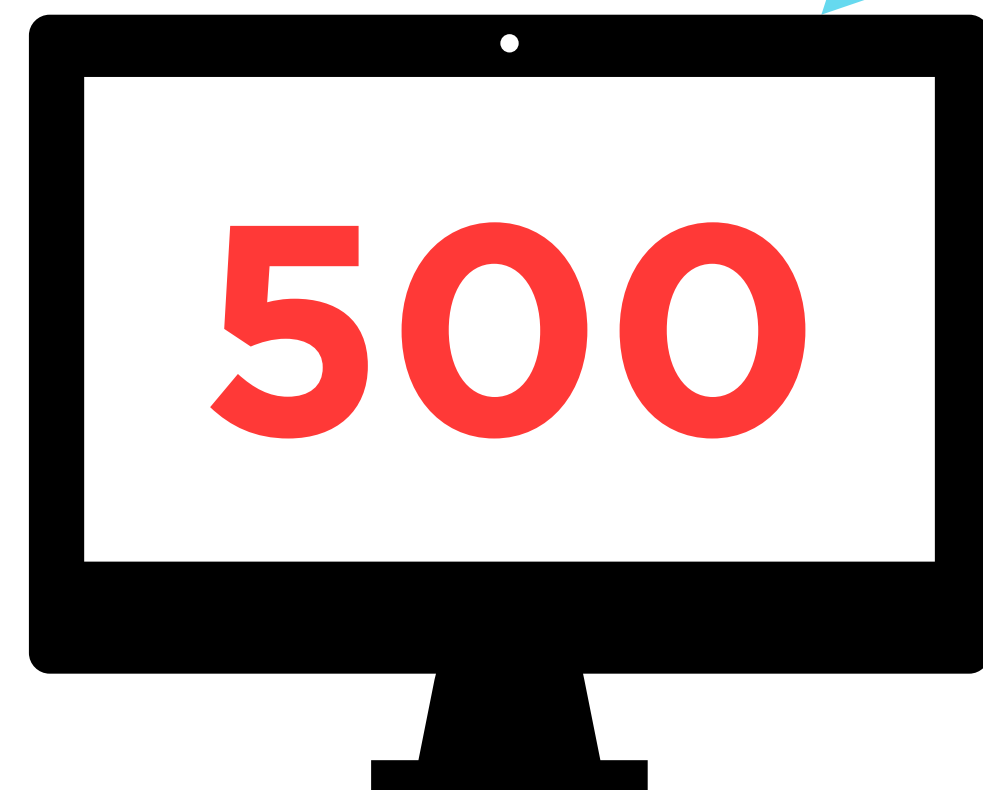
# HTTP



# HTTP



# HTTP

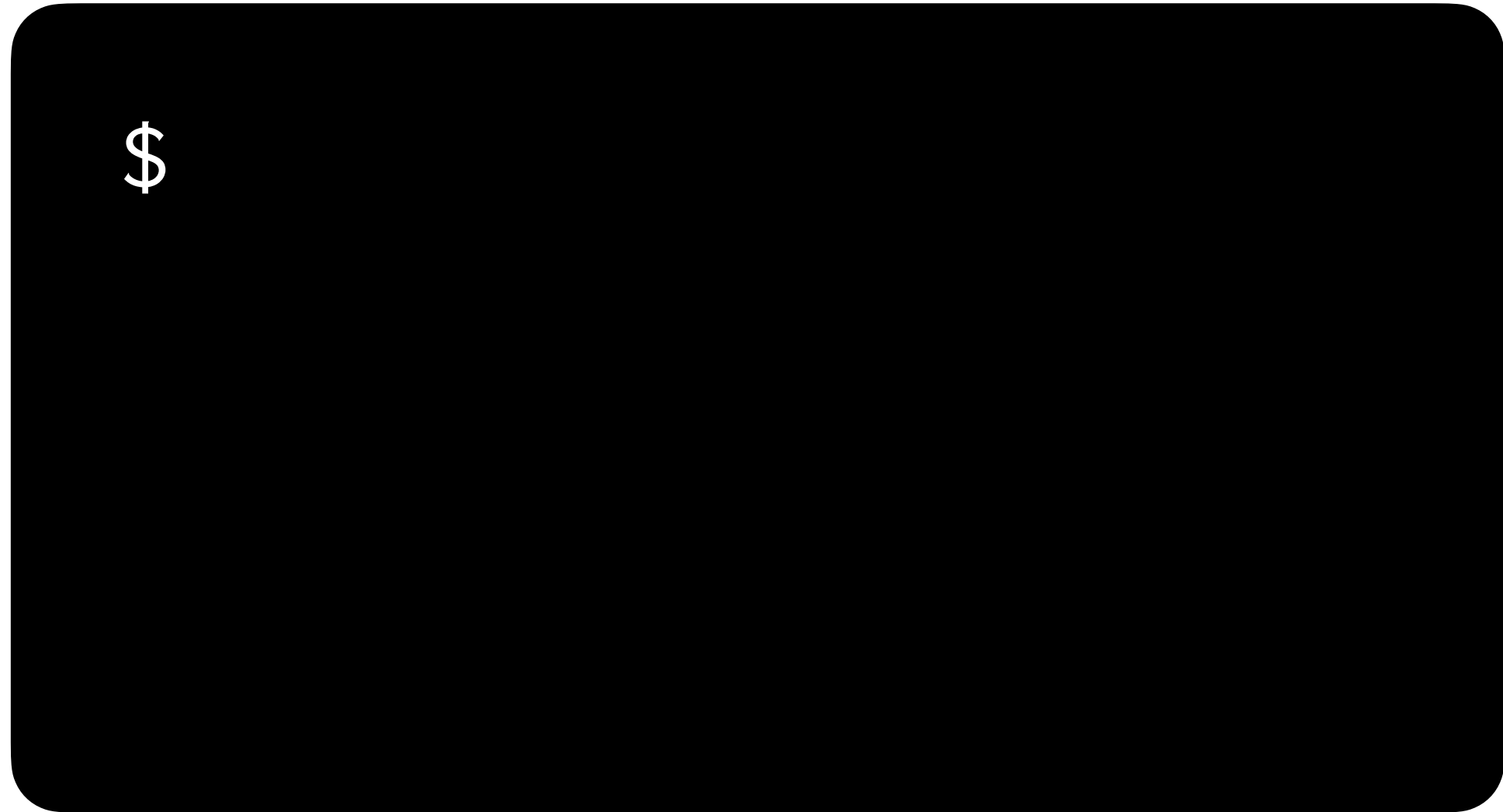


# Scoreboard

Creating a page with a dynamic scoreboard



<https://code.cs50.io/>



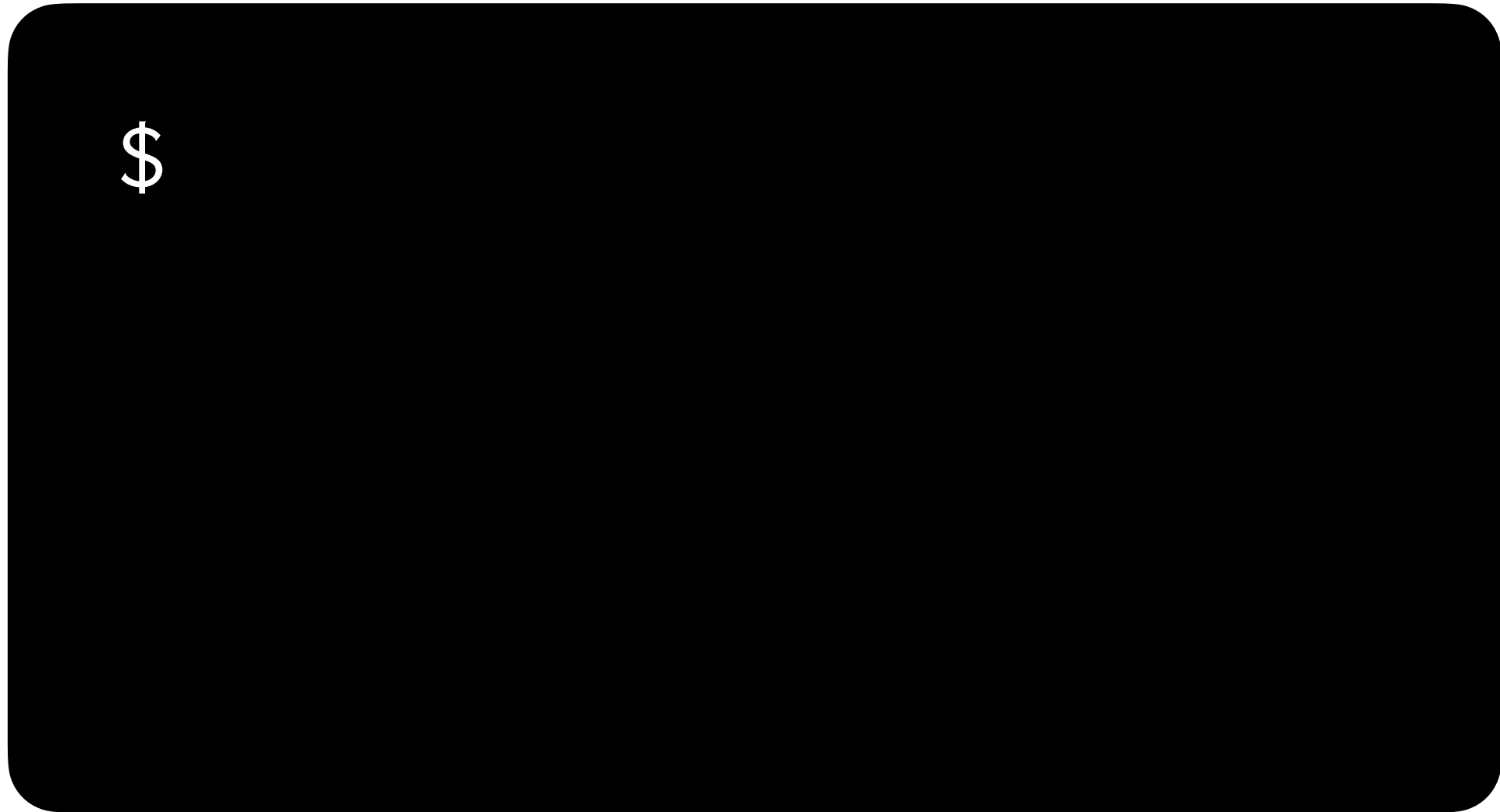
Home Directory



foo.py



bar.py



Home Directory



foo.py



bar.py

```
$ ls
```



Home Directory



foo.py



bar.py

```
$ ls
foo.py bar.py
```



Home Directory



foo.py



bar.py

```
$ mkdir lab2
```



Home Directory

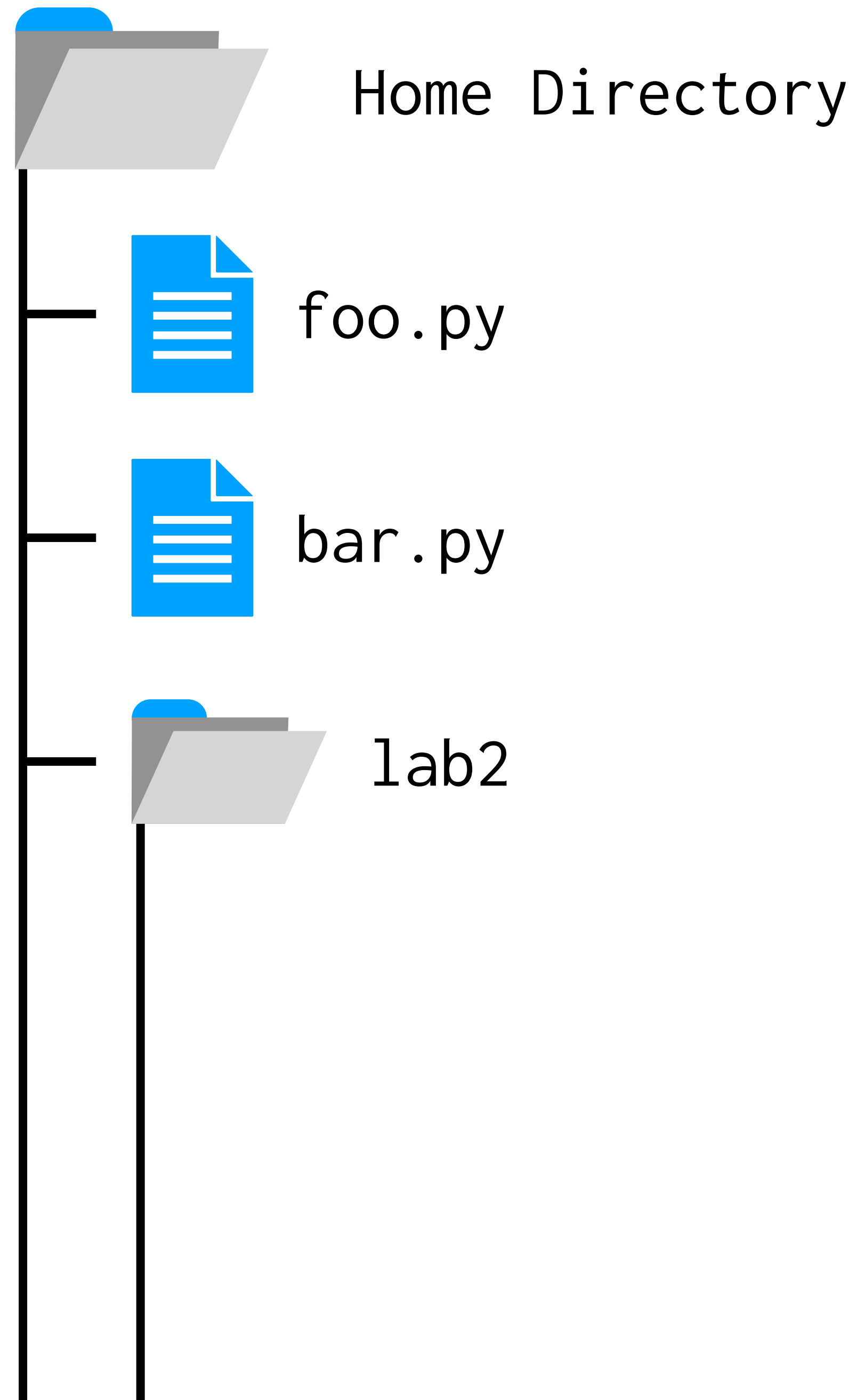


foo.py

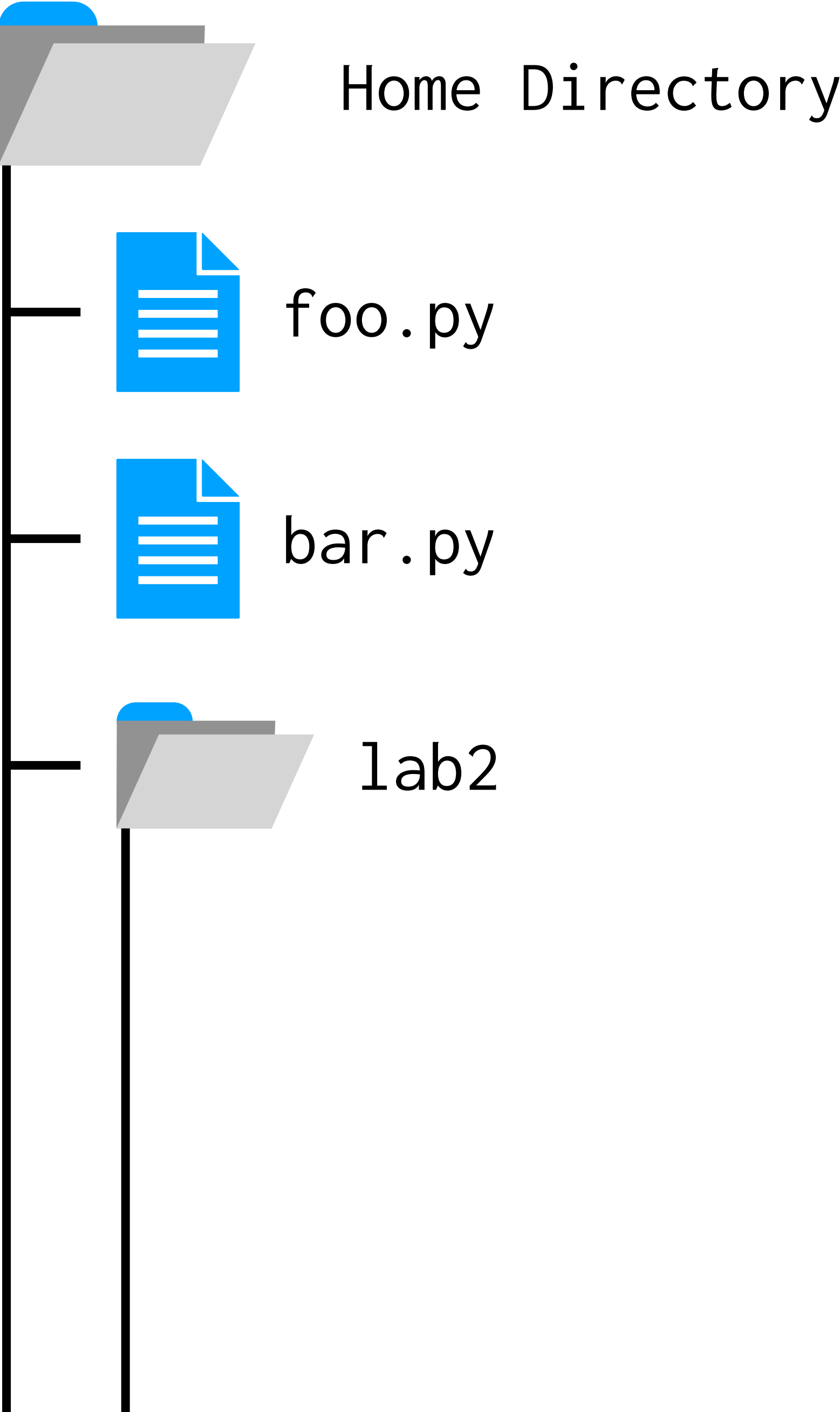


bar.py

```
$ mkdir lab2
```

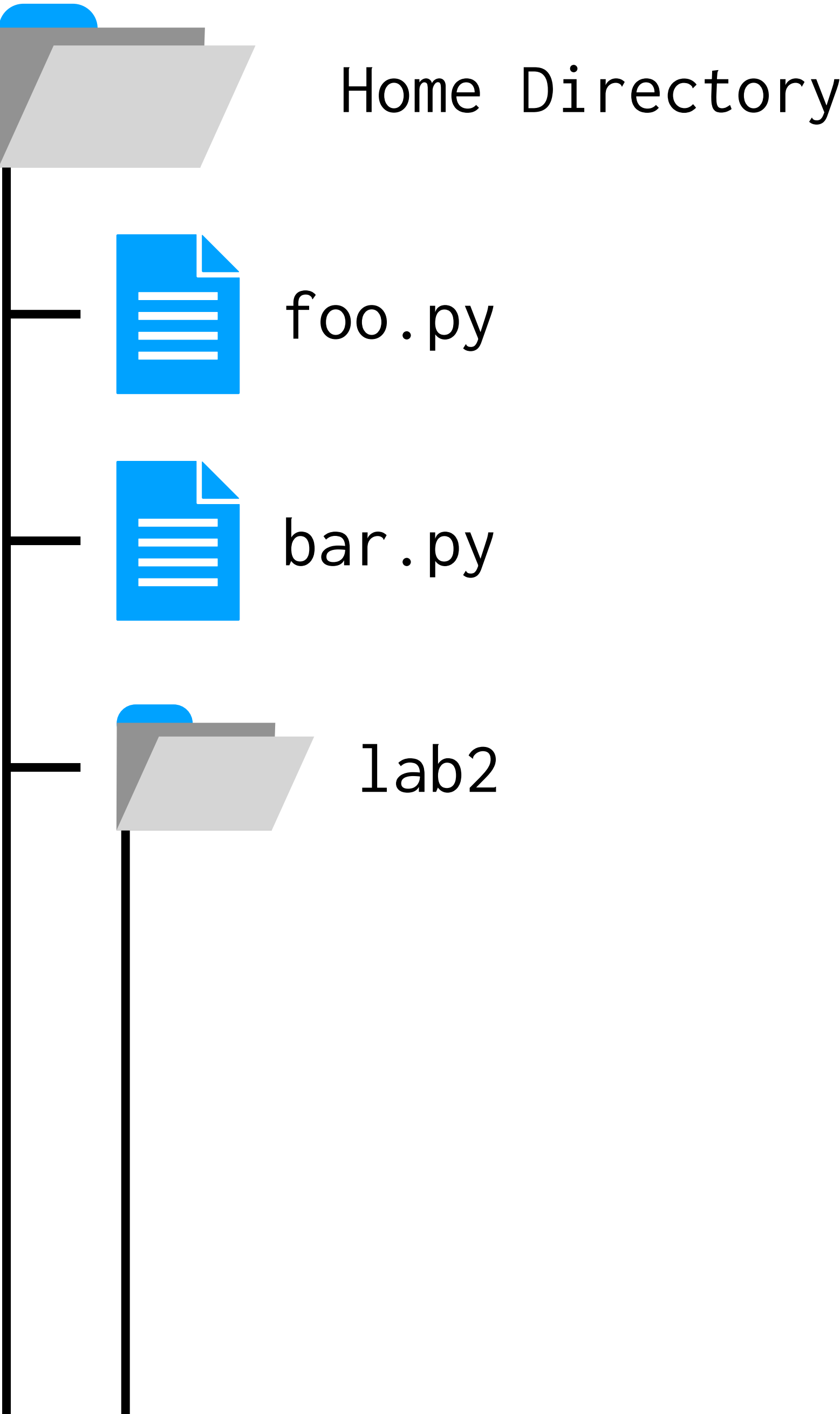


```
$ cd lab2
```

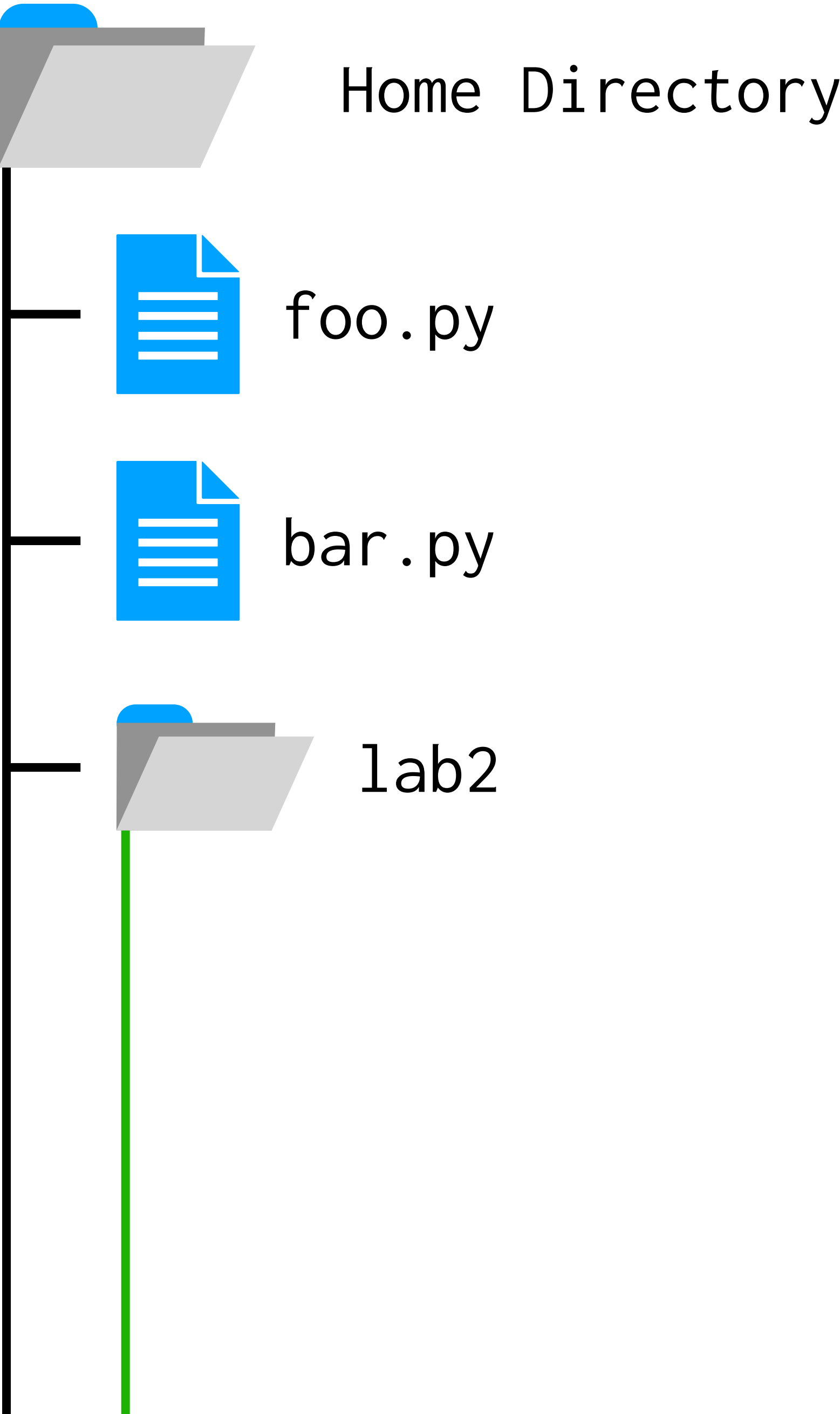




```
lab2/ $
```

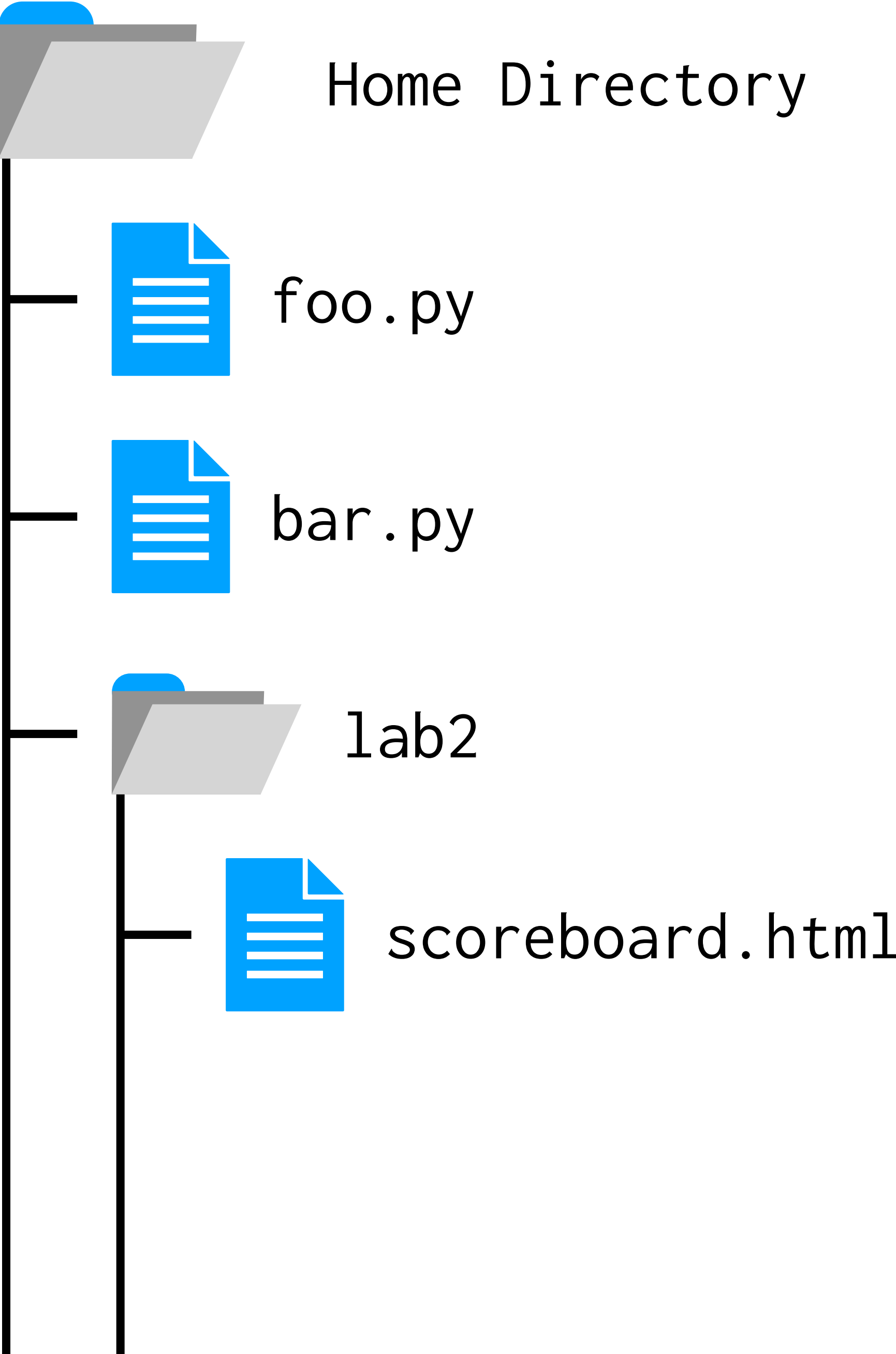


```
lab2/ $
```

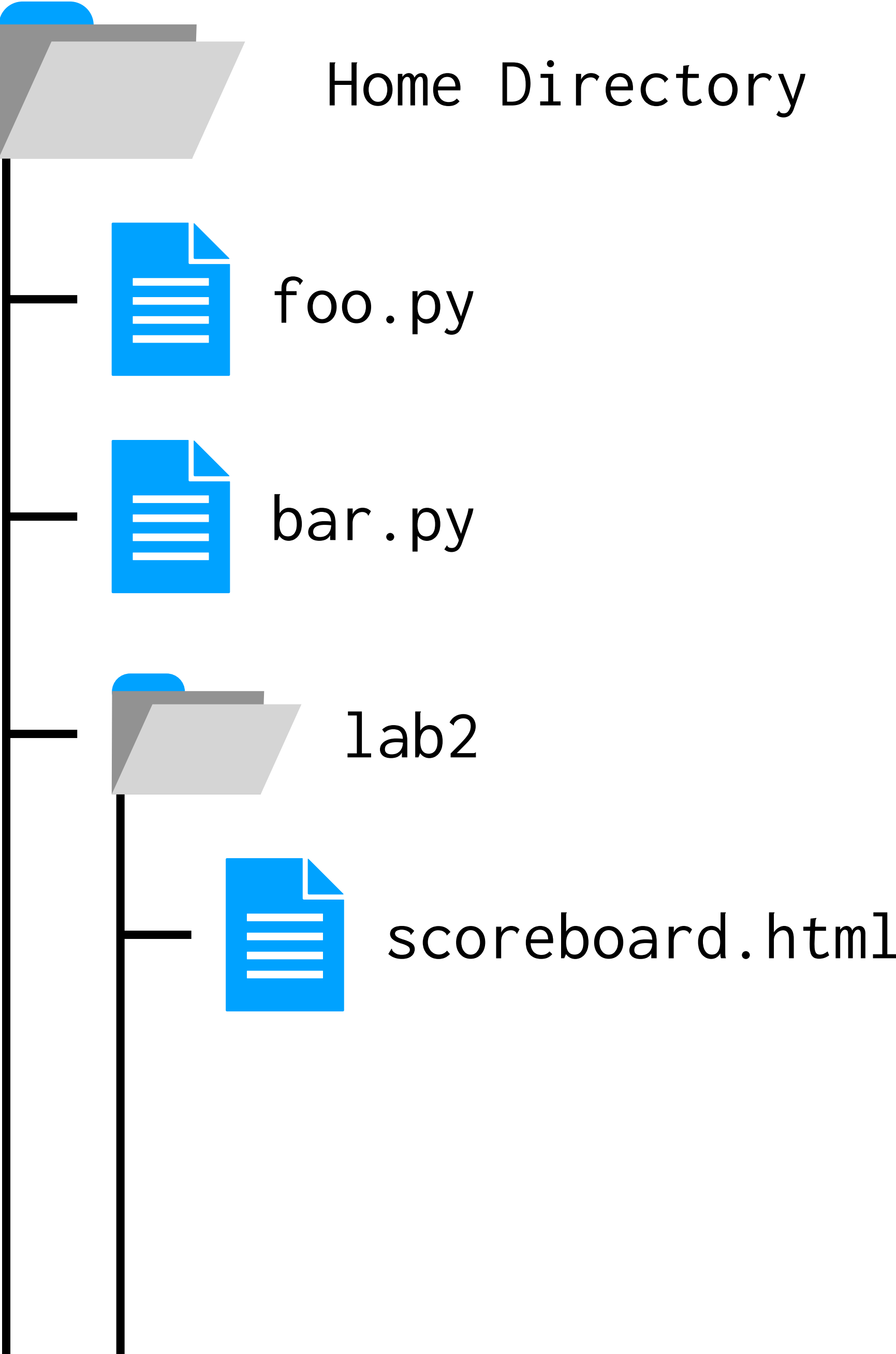


```
wget https://cdn.cs50.net/hls/2023/winter/labs/2/scoreboard.html
```

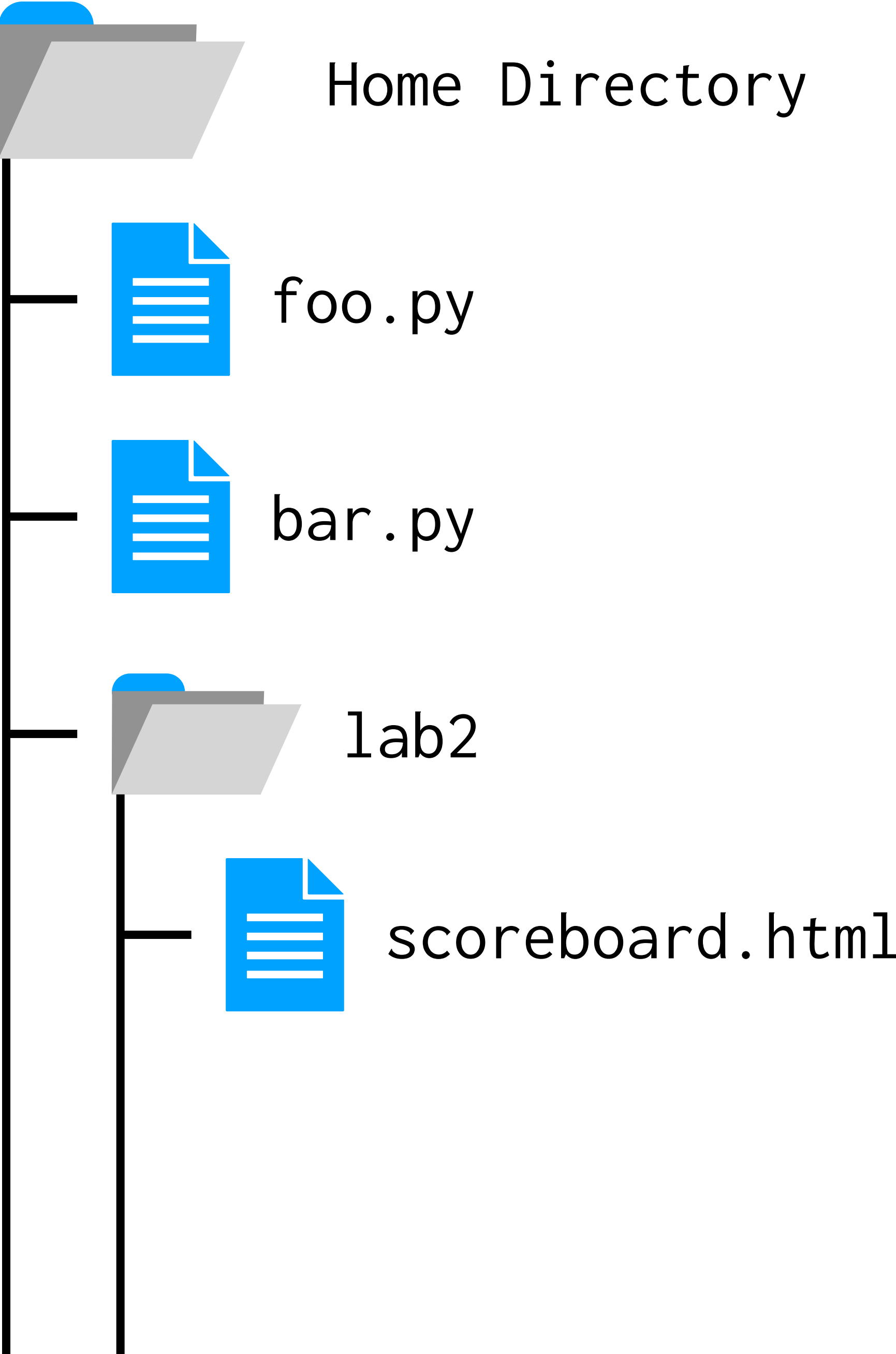
```
lab2/ $
```



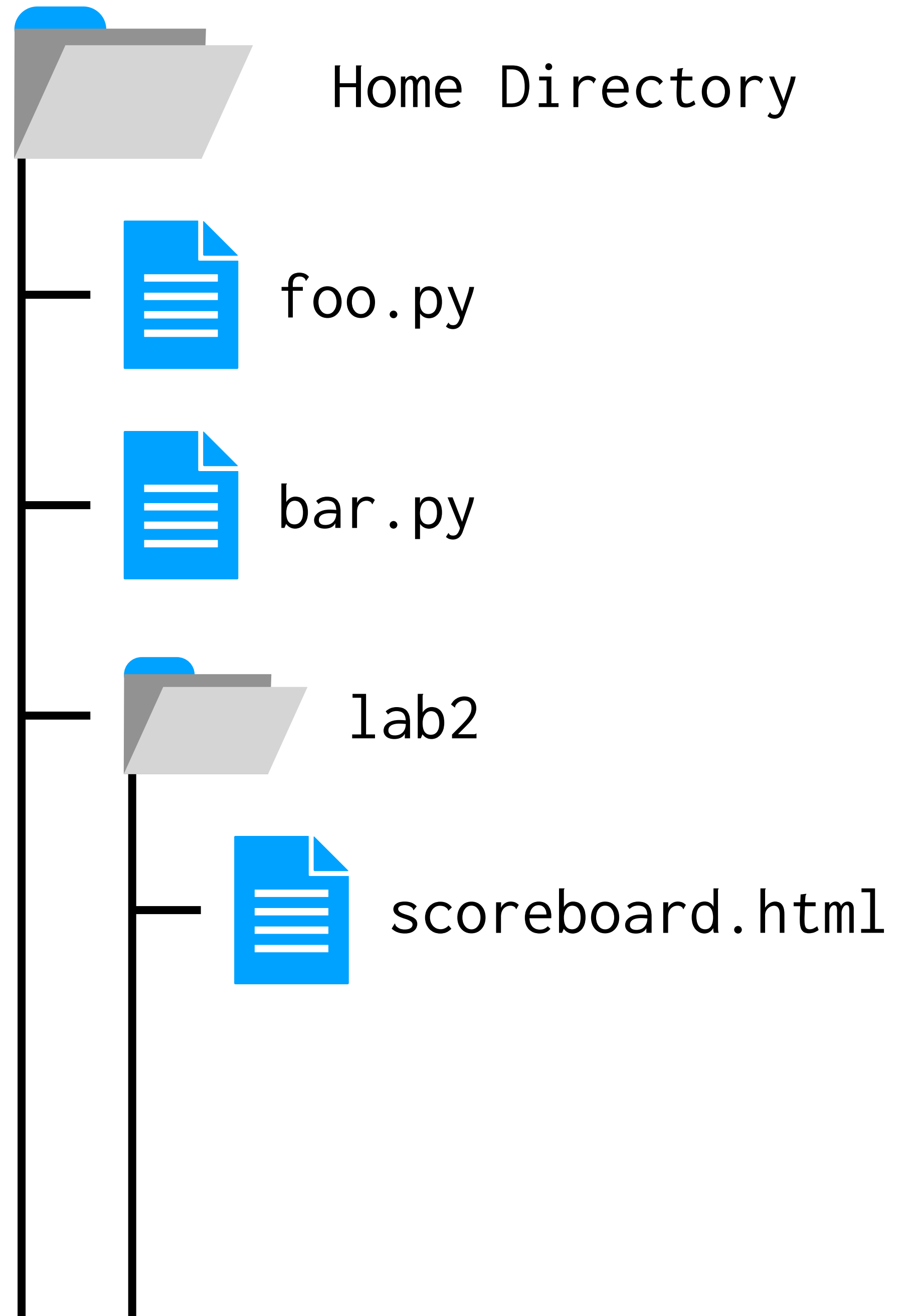
```
lab2/ $ ls
```



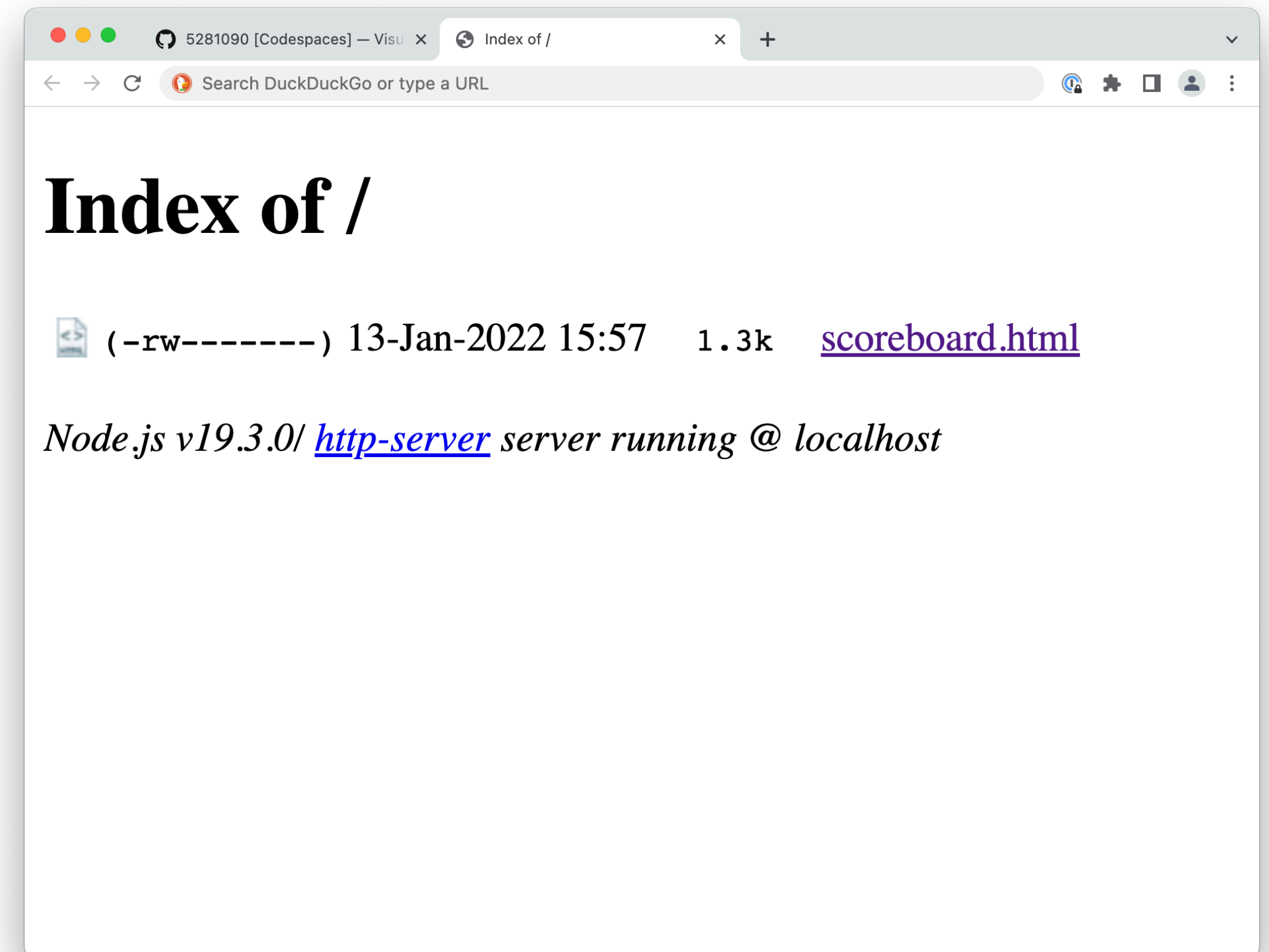
```
lab2/ $ ls
scoreboard.html
```



```
lab2/ $ http-server
```



```
lab2/ $ http-server
```



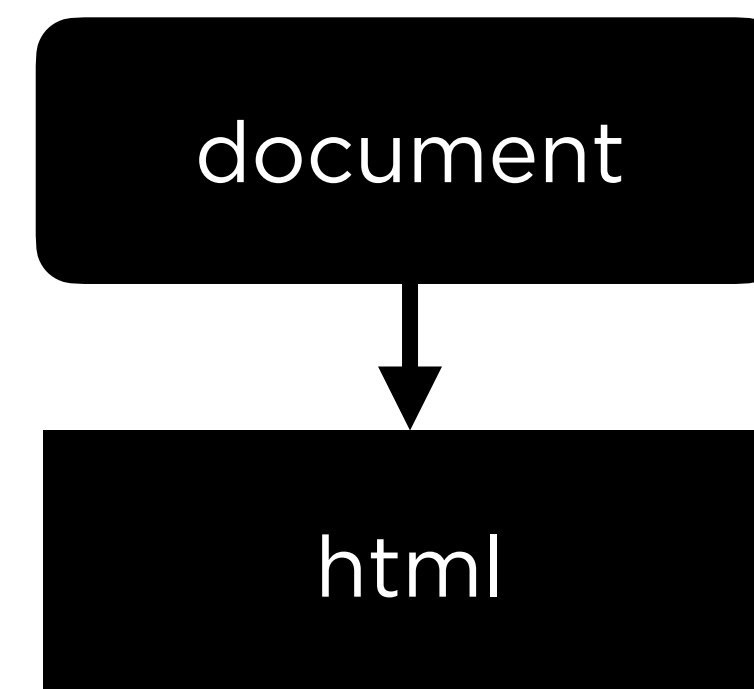


# HTML

The structure of web pages

```
<!DOCTYPE html>  
<html>
```

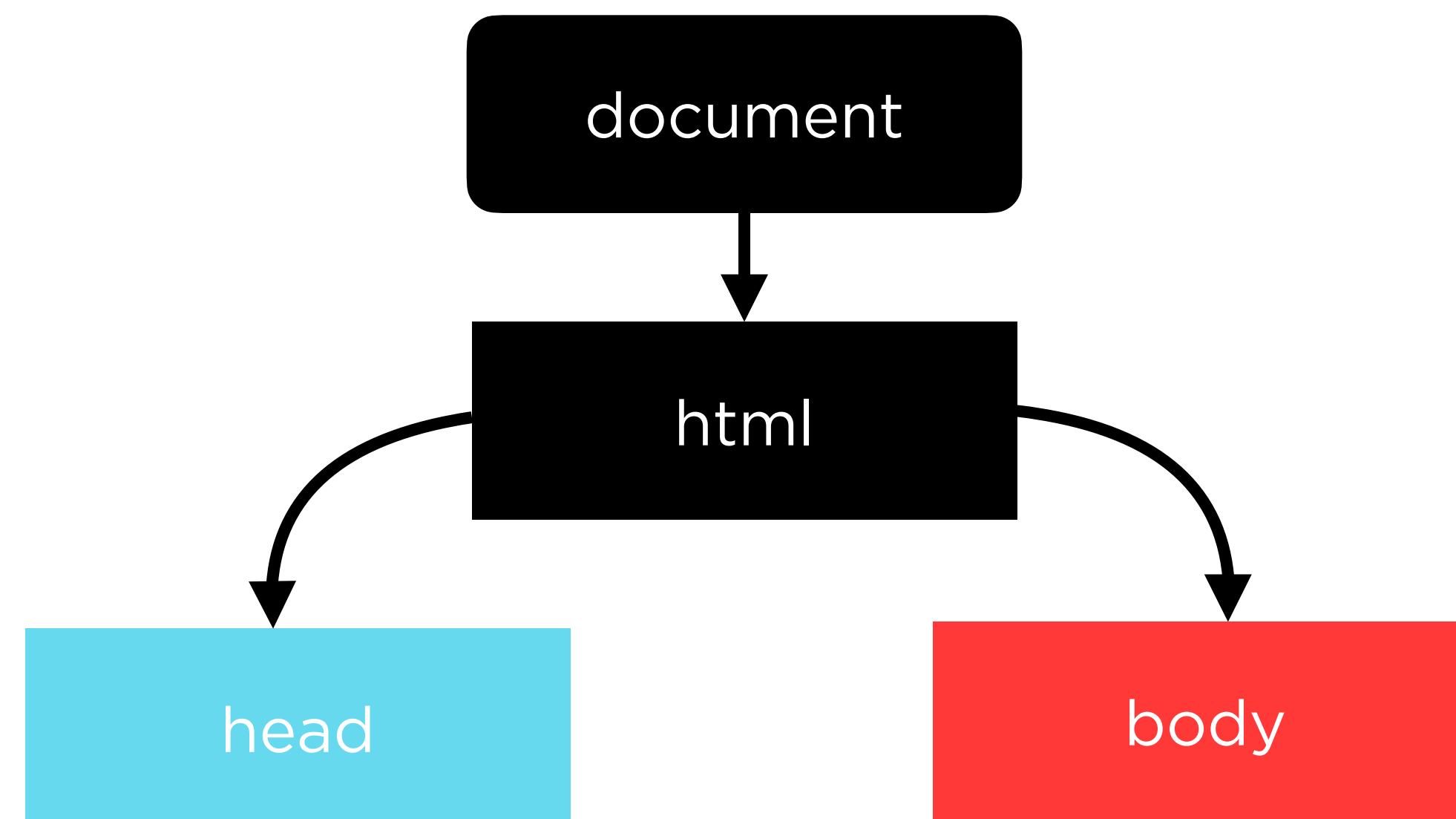
```
</html>
```



```
<!DOCTYPE html>
<html>
  <head>

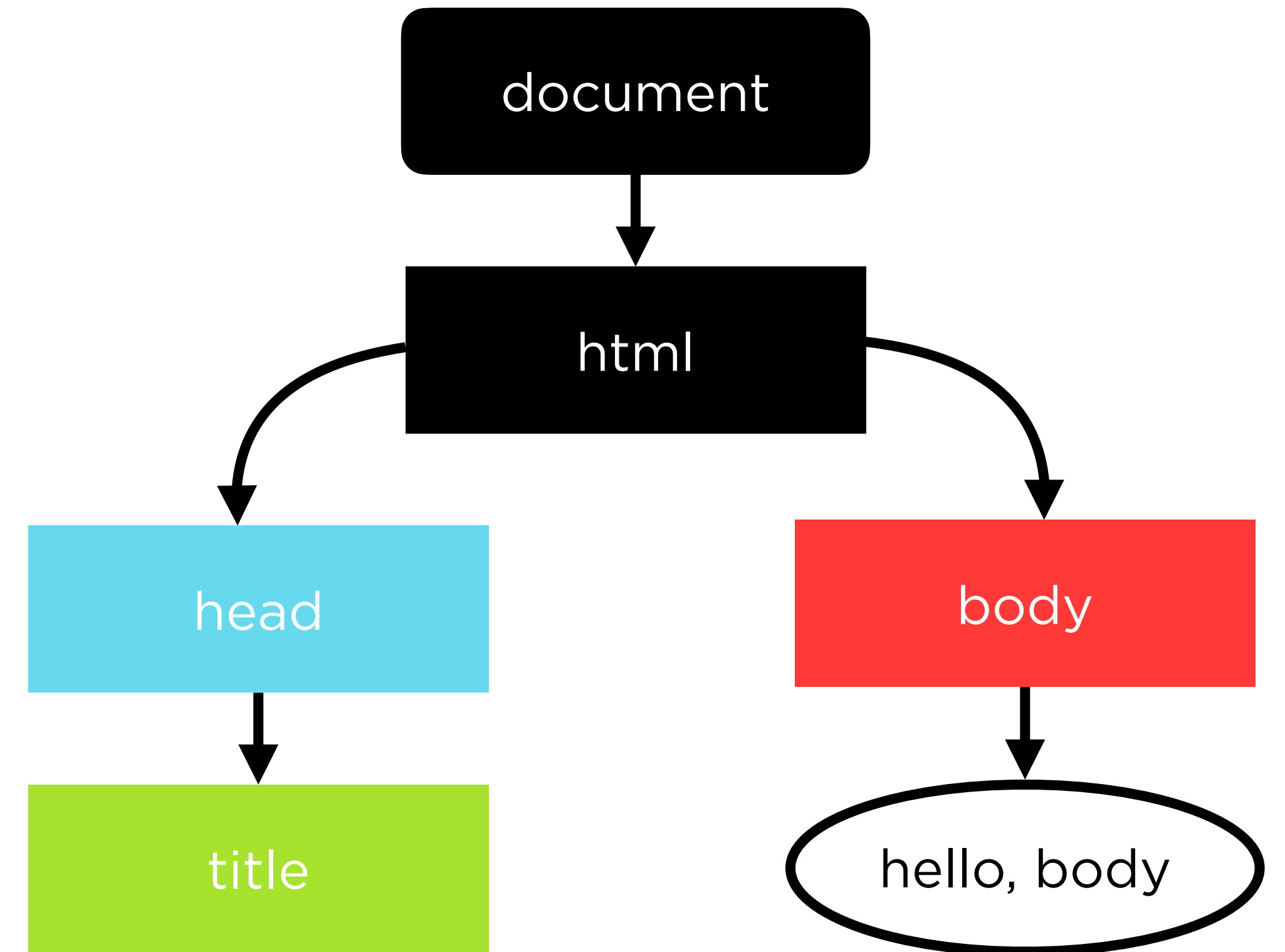
  </head>
  <body>

  </body>
</html>
```

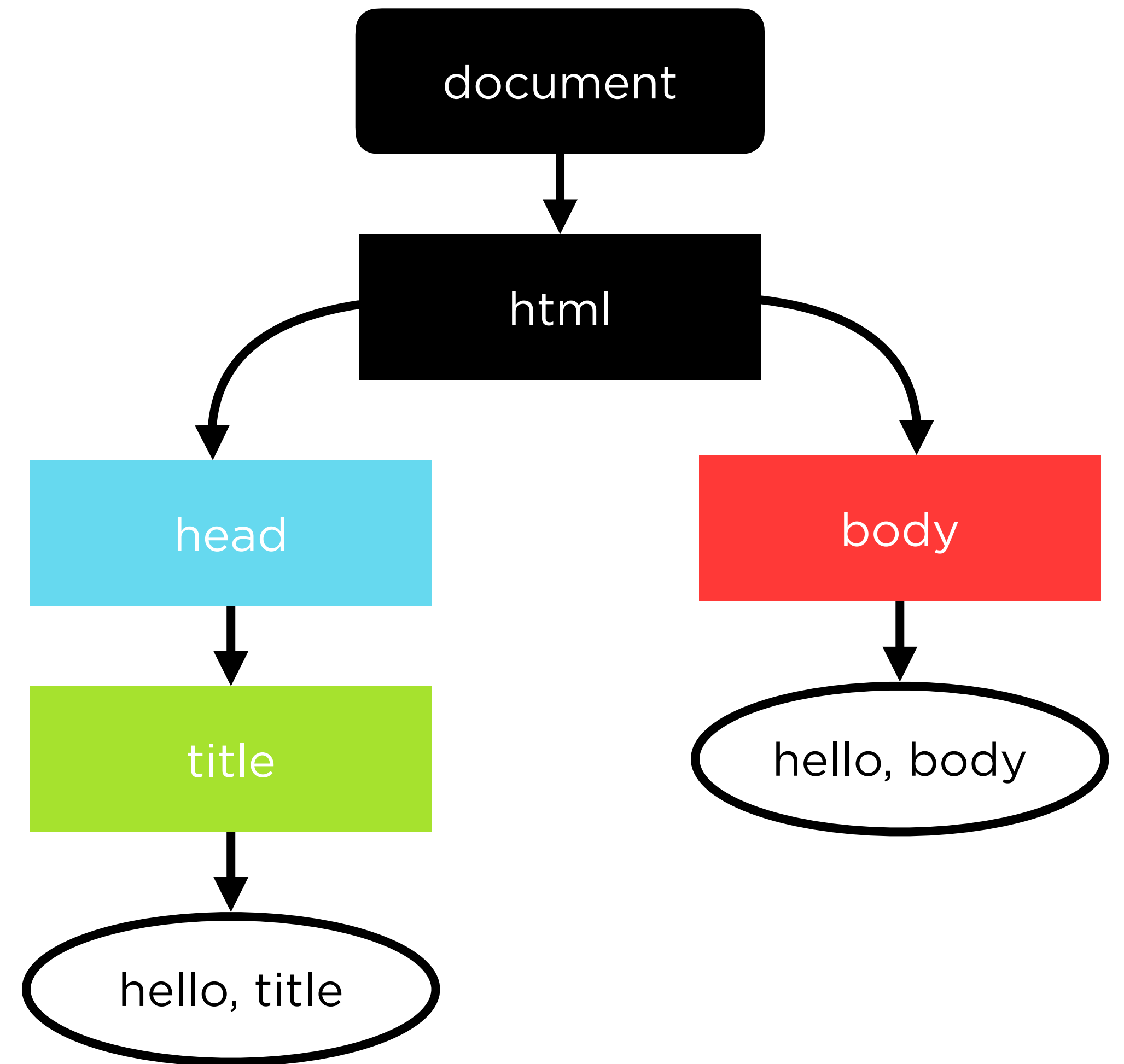


```
<!DOCTYPE html>
<html>
  <head>
    <title>

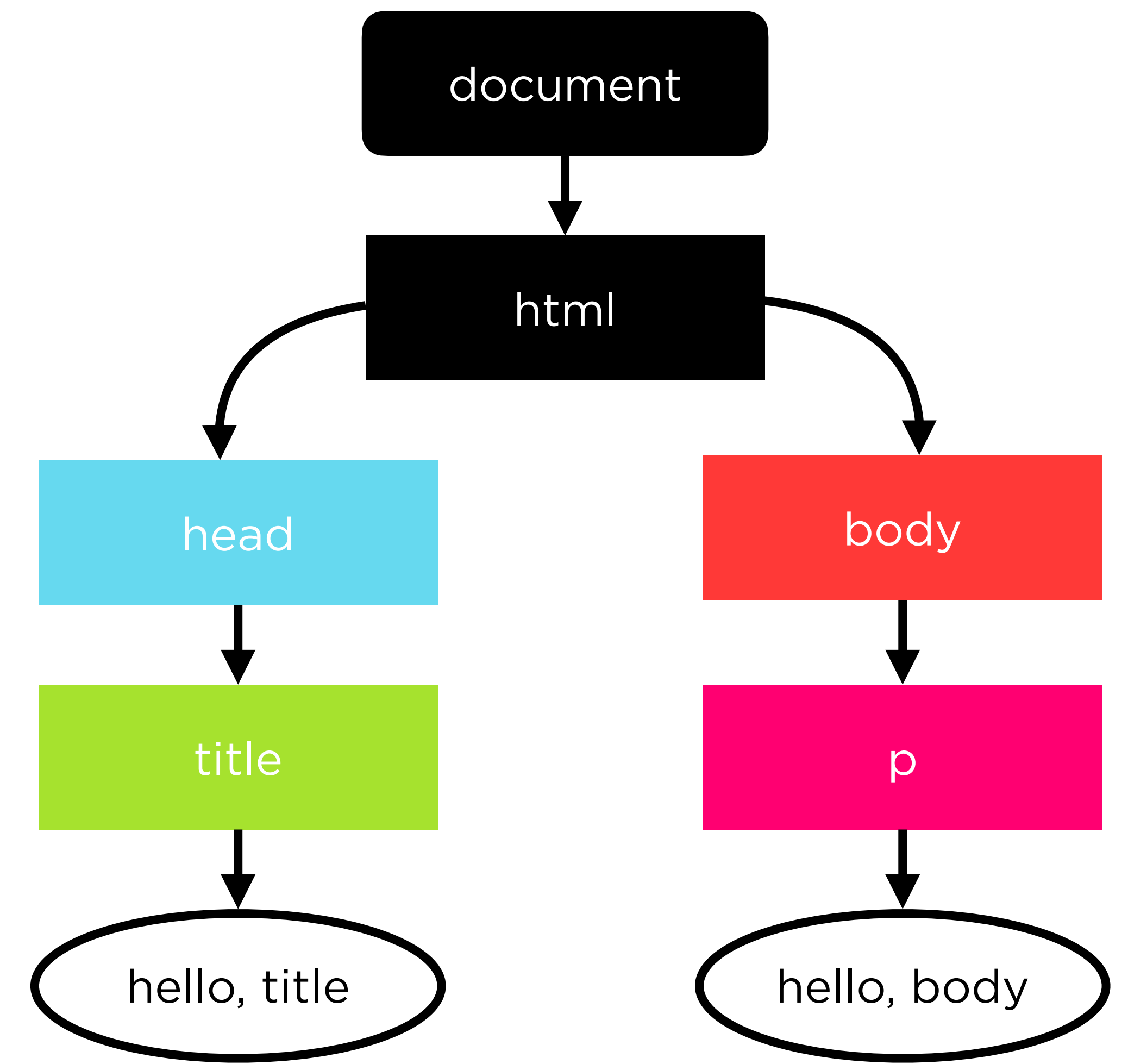
    </title>
  </head>
  <body>
    hello, body
  </body>
</html>
```



```
<!DOCTYPE html>
<html>
  <head>
    <title>
      hello, title
    </title>
  </head>
  <body>
    hello, body
  </body>
</html>
```



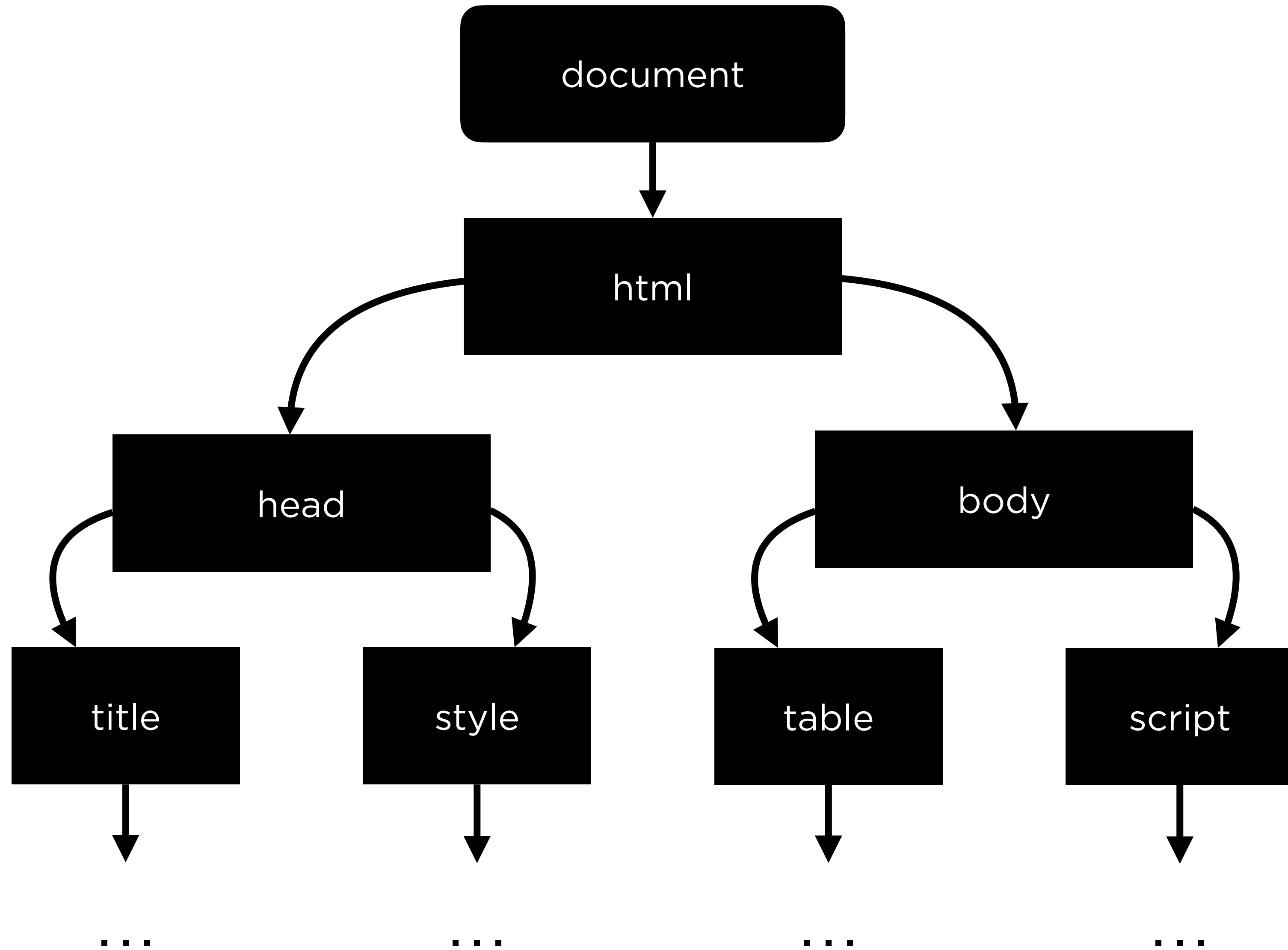
```
<!DOCTYPE html>
<html>
  <head>
    <title>
      hello, title
    </title>
  </head>
  <body>
    <p>hello, body</p>
  </body>
</html>
```



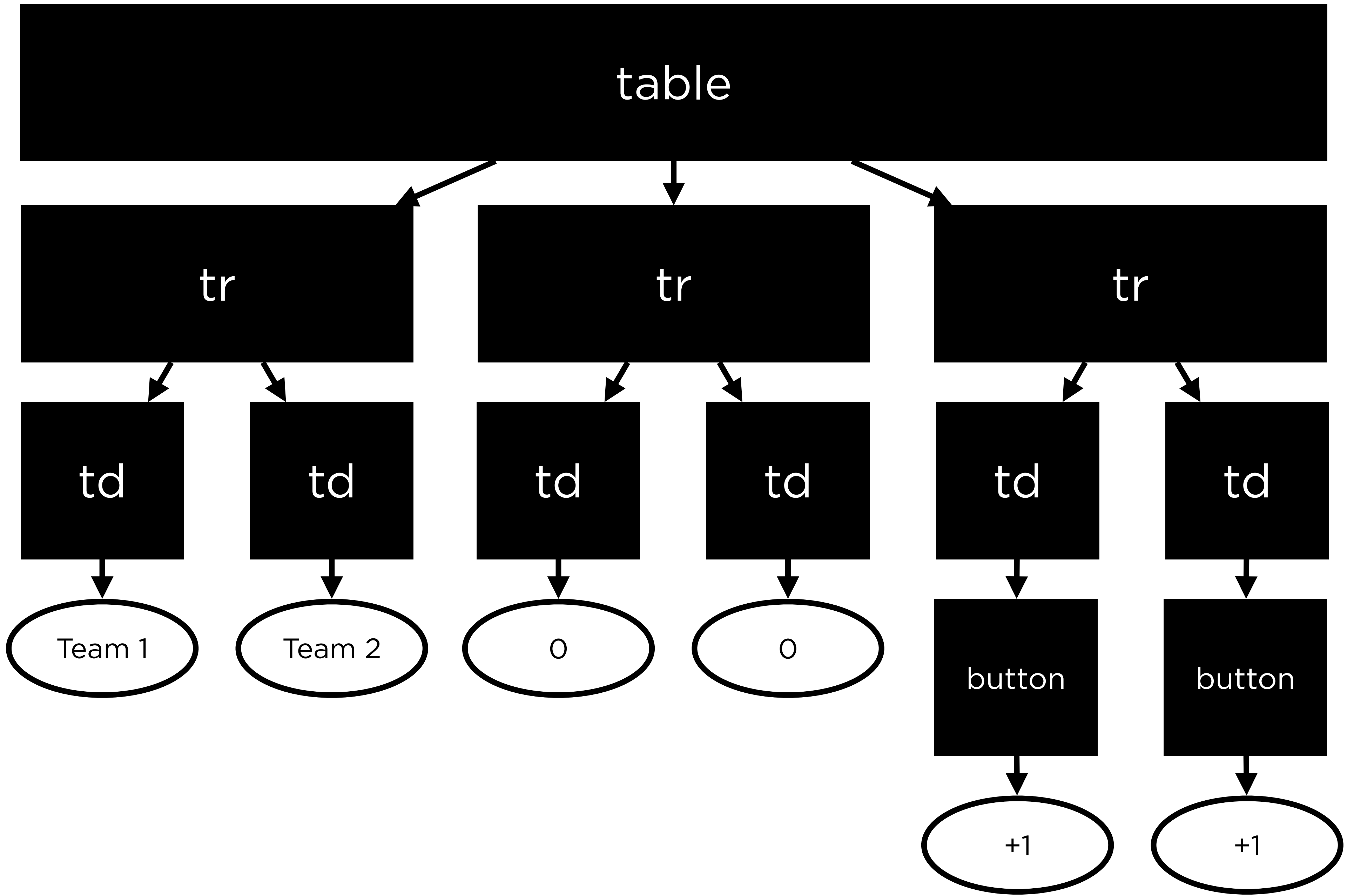
scoreboard.html — 5281090 [C x] Scoreboard x +

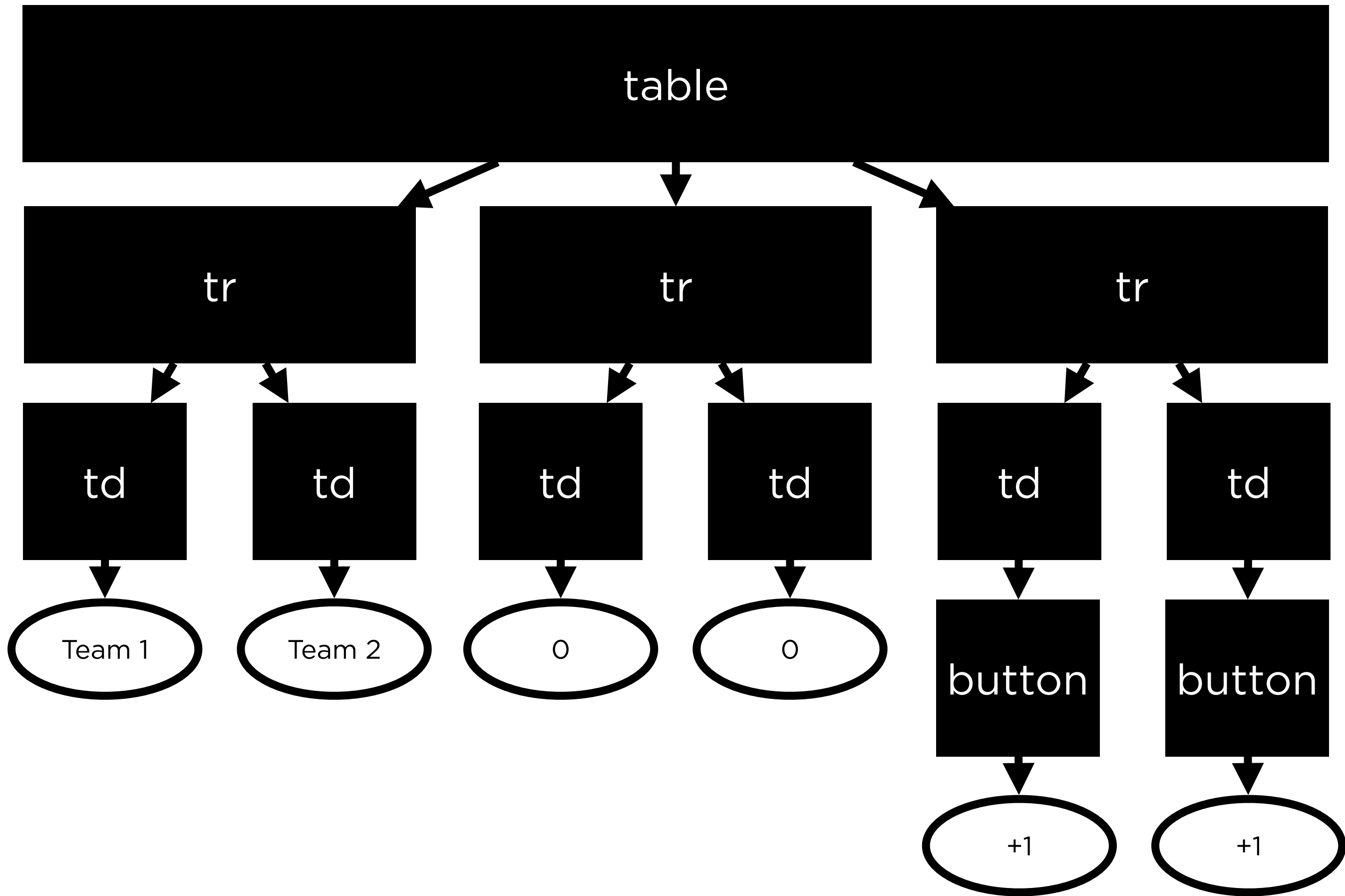
← → ↻ Search DuckDuckGo or type a URL 🔍 ⚙️ 🏠 👤 ⋮

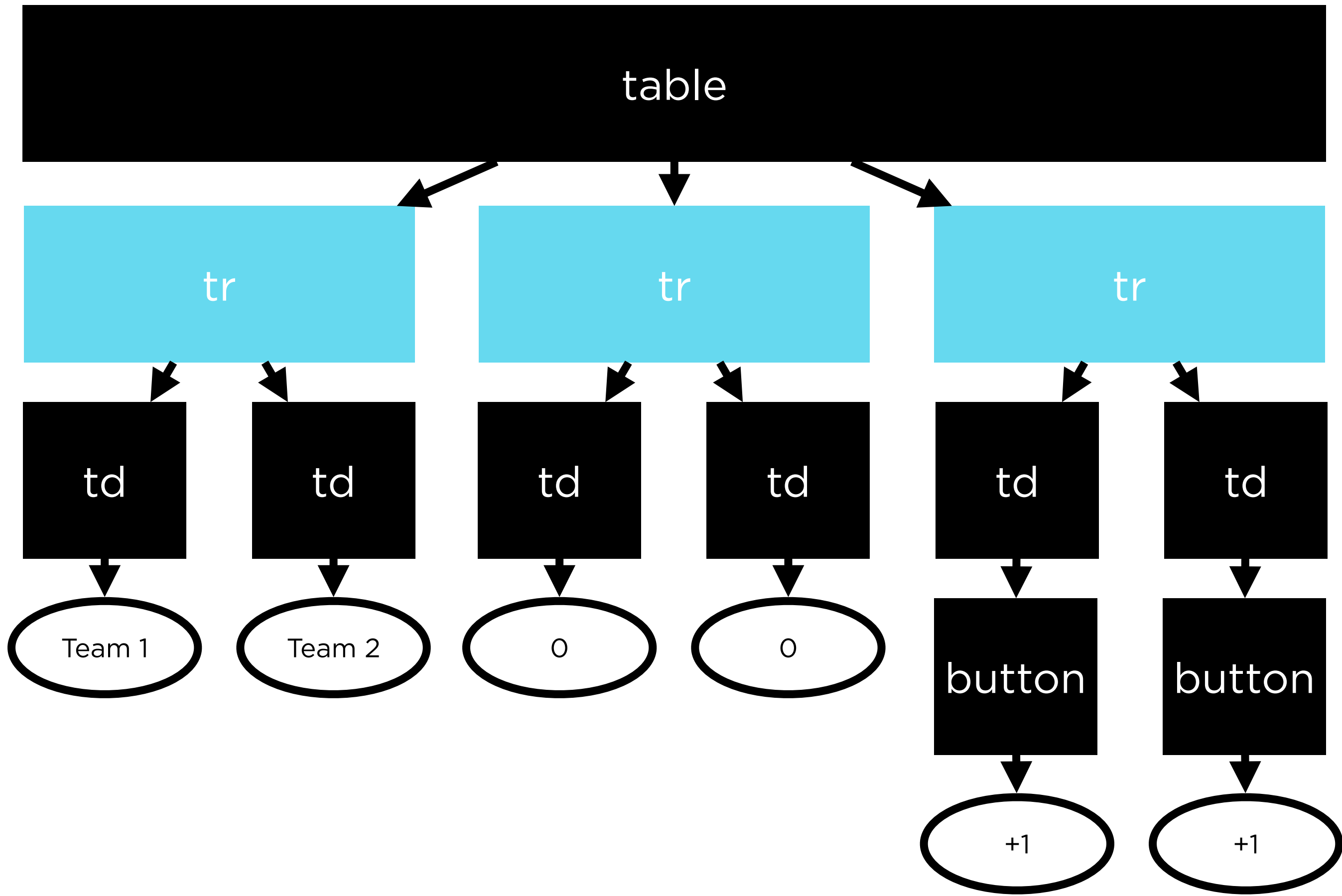
Team 1	Team 2
0	0
+1	+1

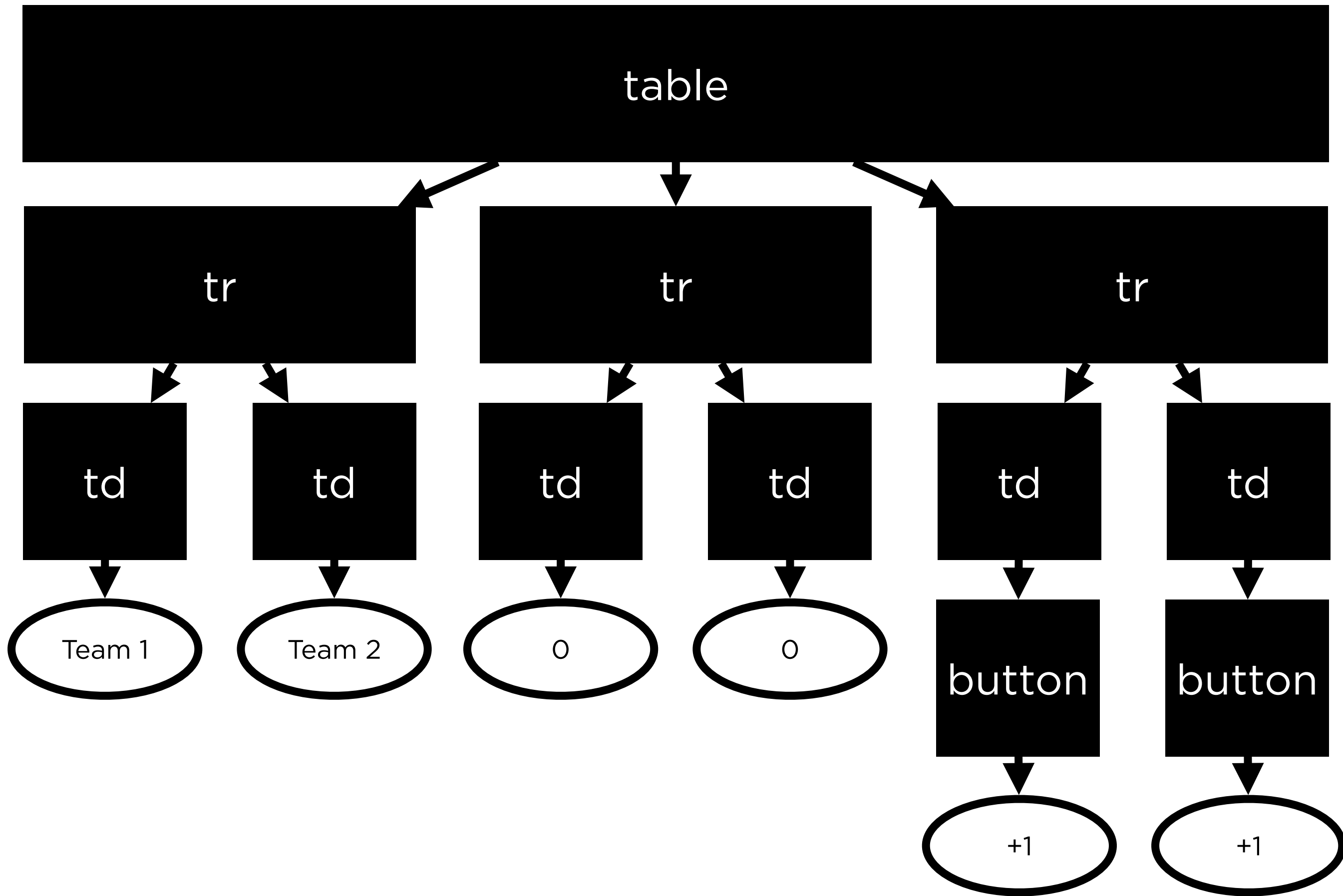


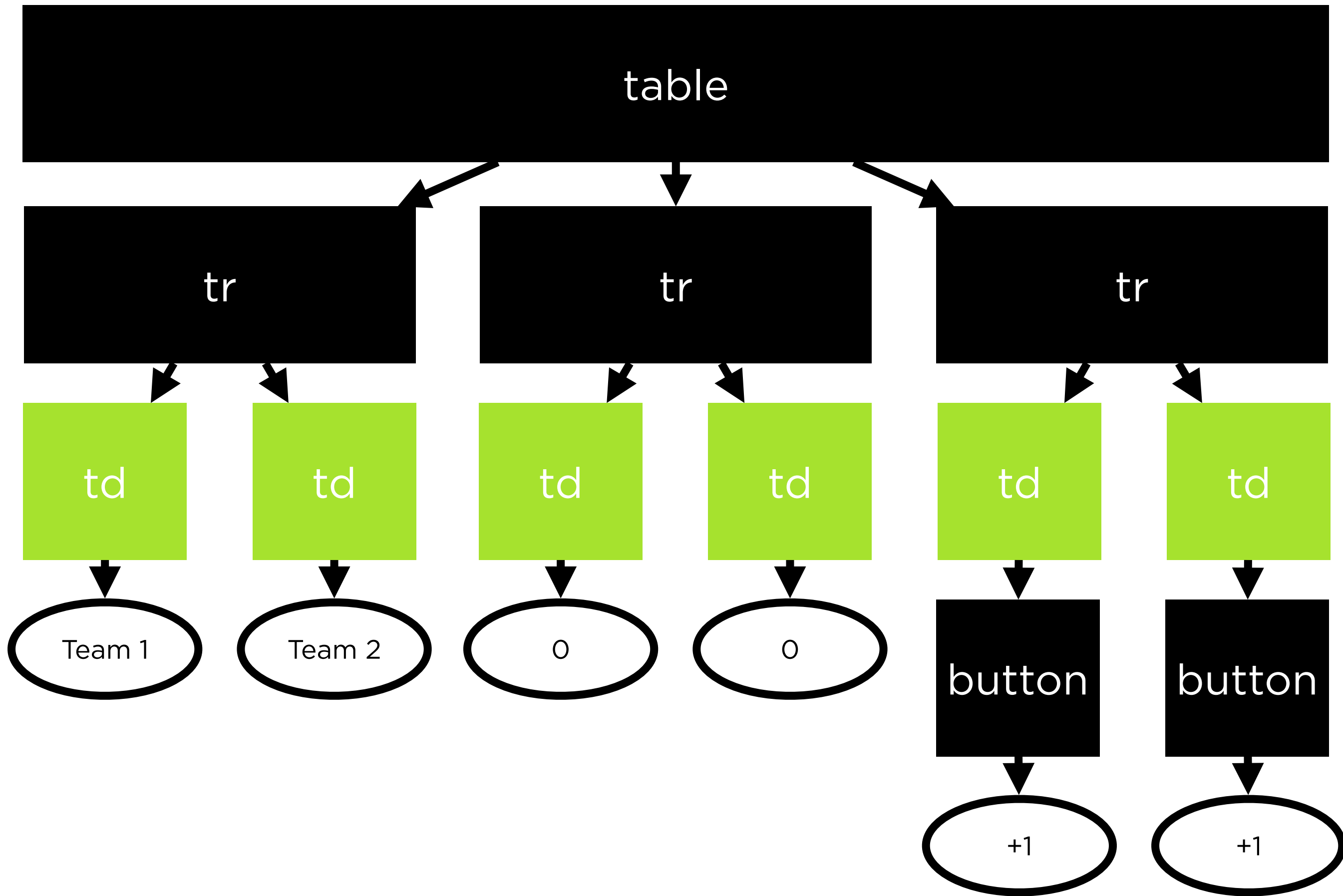




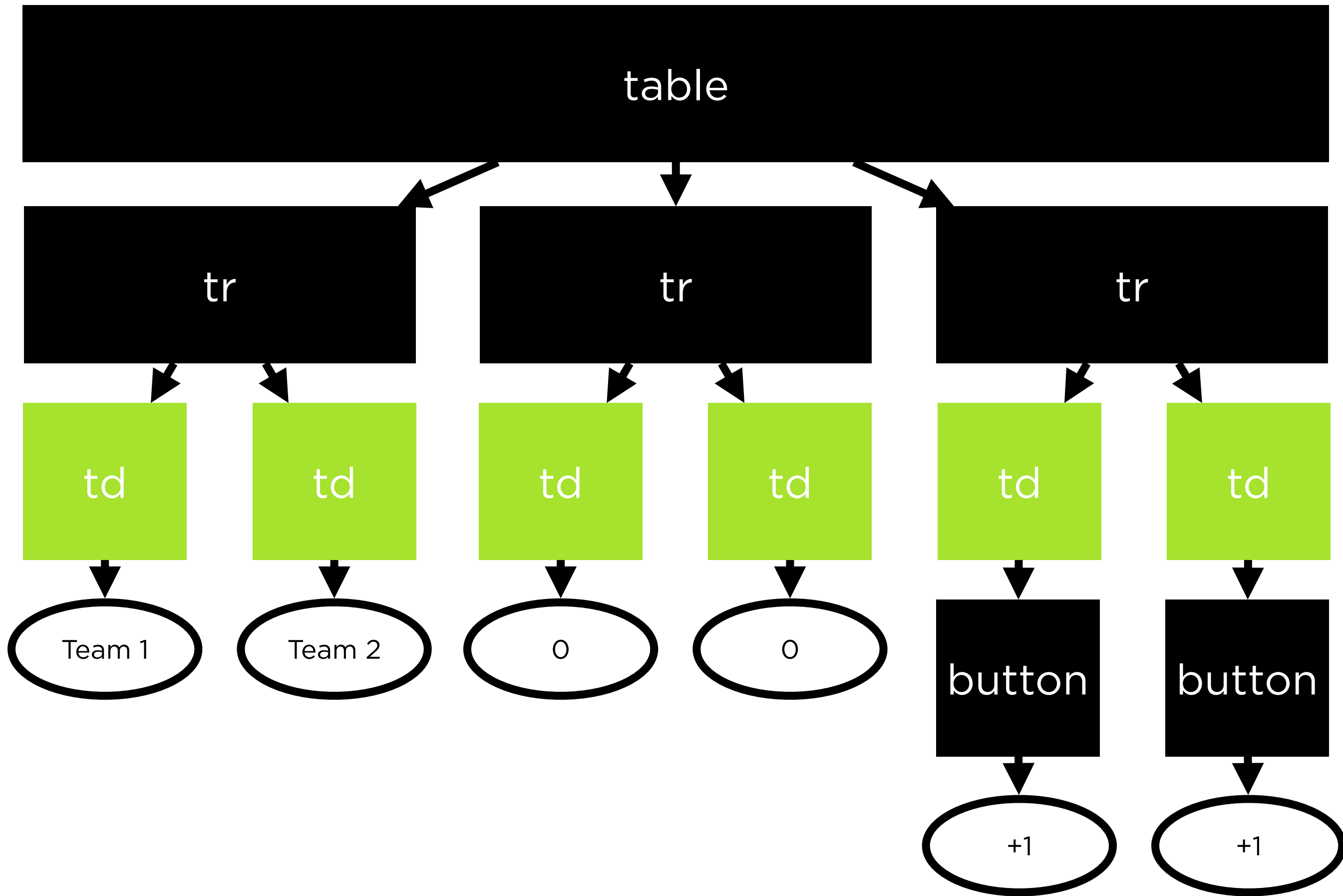


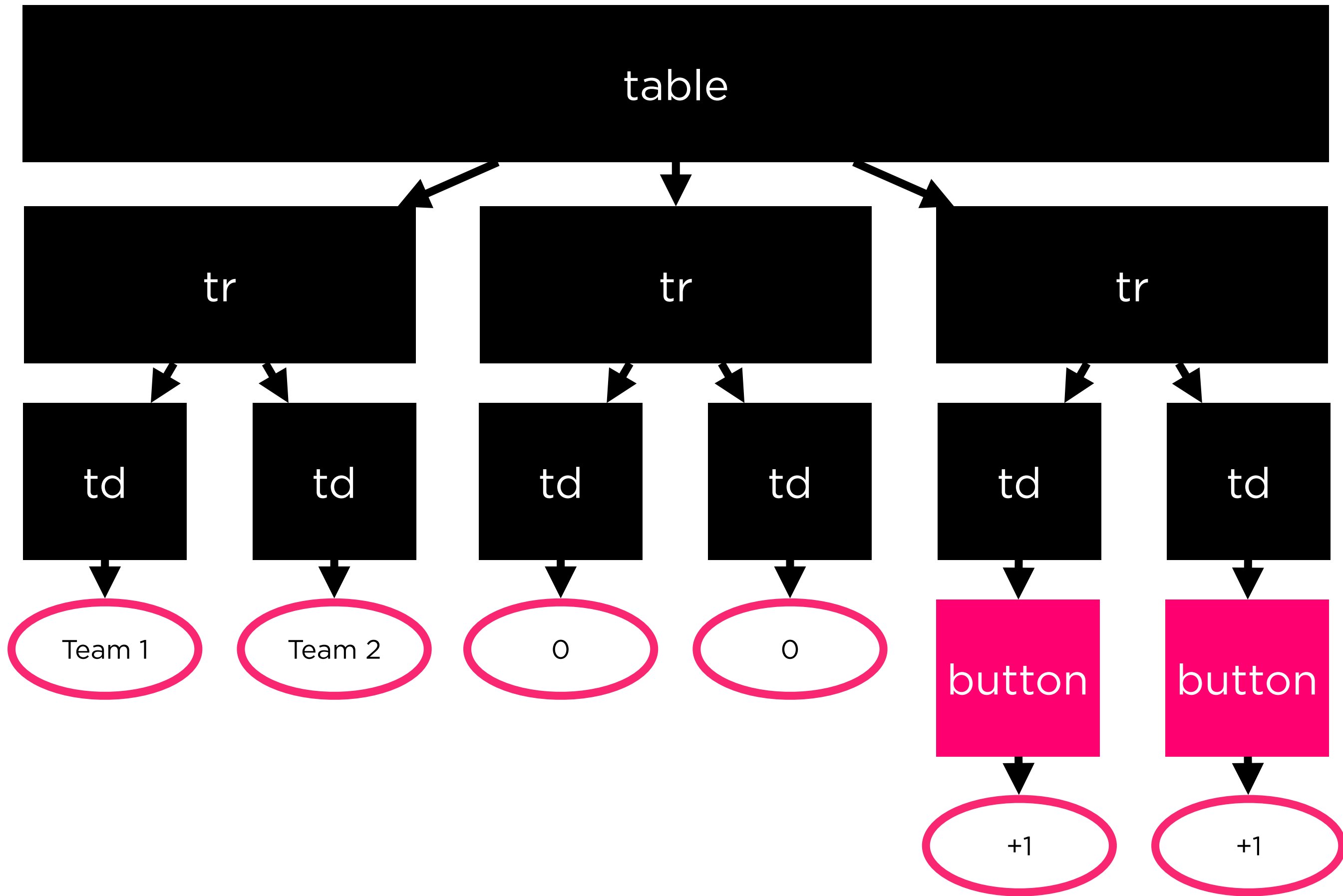


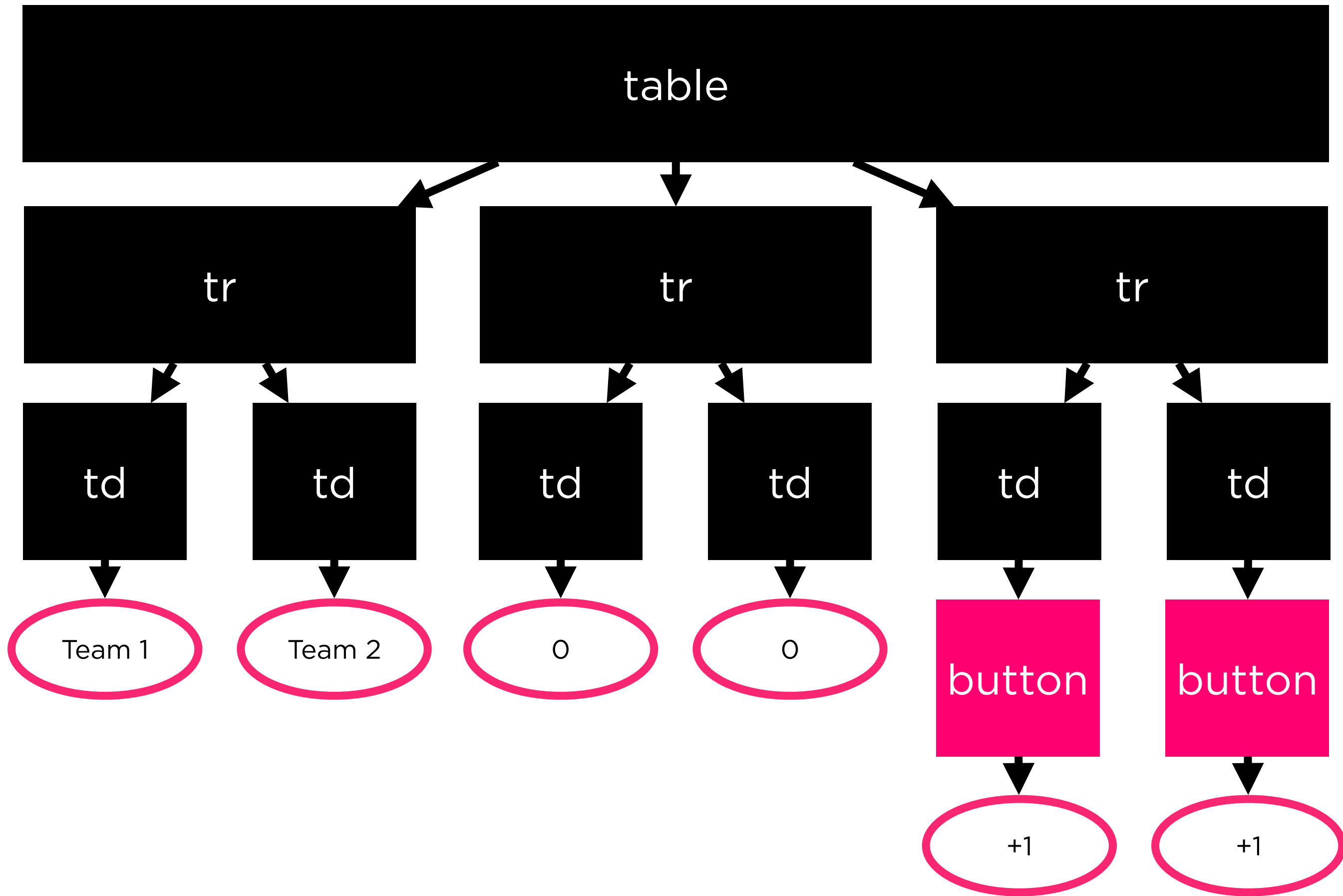



A table with three empty rows, represented by a black outline with three horizontal sections.



Team 1	Team 2
0	0
<input type="button" value="+1"/>	<input type="button" value="+1"/>



# Headings

Titling a page

**Heading 1**

**Heading 2**

**Heading 3**

**Heading 4**

**Heading 5**

**Heading 6**

<h1></h1>

<h2></h2>

<h3></h3>

<h4></h4>

<h5></h5>

<h6></h6>

# Paragraphs

Adding regular text

This is paragraph text. It's regular text that's part of my webpage. Each new paragraph tag will create a new paragraph.

Like this!

<p>This is paragraph text. It's regular text that's part of my webpage. Each new paragraph tag will create a new paragraph.</p>

<p>Like this!</p>

# Images

Adding photos to your page

```
<img></img>
```



```
</img>
```



```
</img>
```



# Your turn!

In **scoreboard.html**, experiment by adding your choices of...

- A title
- Additional table rows
- Instructions for the scoreboard
- Images
- Something else!

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>


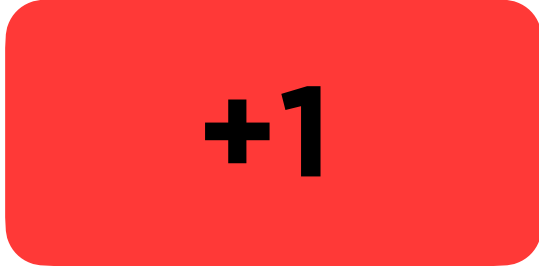
# CSS

Styling web pages

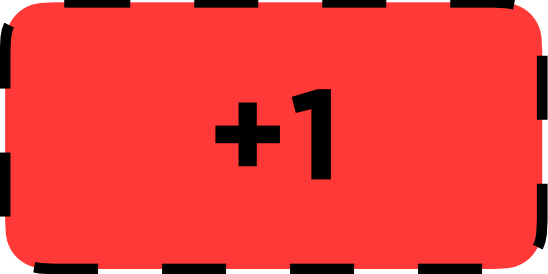
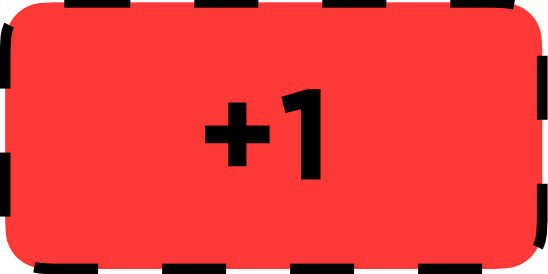
```
selector {  
  property: value;  
}
```

<b>Team 1</b>	<b>Team 2</b>
<b>0</b>	<b>0</b>
<b>+1</b>	<b>+1</b>

```
button {  
  background-color: red;  
}
```

Team 1	Team 2
0	0
	

```
button {  
  background-color: red;  
  border: 4pt dashed;  
}
```

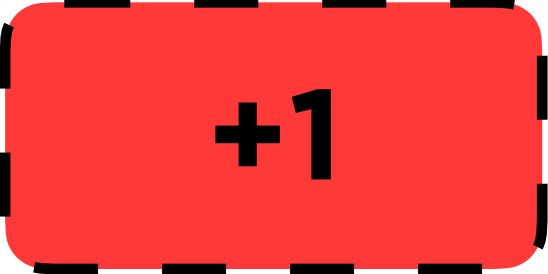
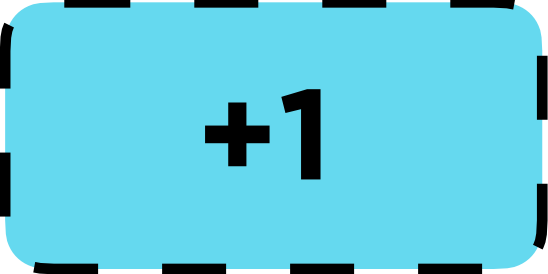
<b>Team 1</b>	<b>Team 2</b>
0	0
	



# **ID Selectors**

Selecting single elements

```
#team1-button {  
  background-color: red;  
  border: 4pt dashed;  
}  
  
#team2-button  
{  
  background-color: blue;  
  border: 4pt dashed;  
}
```

Team 1	Team 2
0	0
	

```
#team1-button {  
  background-color: red;  
  border: 4pt dashed;  
}
```

```
#team2-button {  
  background-color: blue;  
  border: 4pt dashed;  
}
```

...

```
<button id="team1-button">  
  Team 1  
</button>
```

...

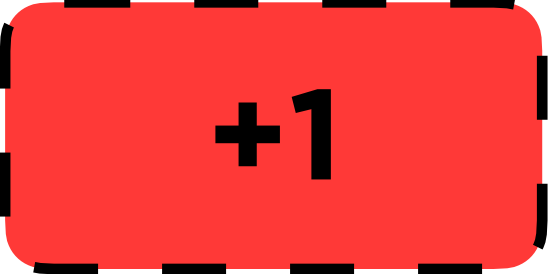
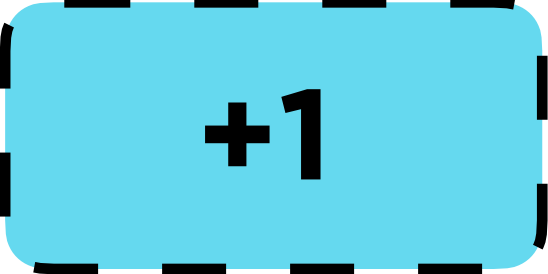
```
<button id="team2-button">  
  Team 2  
</button>
```

...

# Type Selectors

Selecting all elements of a certain type

```
td {  
  background-color: gray;  
}
```

Team 1	Team 2
0	0
	

```
td {  
  background-color: gray;  
}
```

<b>Team 1</b>	<b>Team 2</b>
<b>0</b>	<b>0</b>
<b>+1</b>	<b>+1</b>

# **Class Selectors**

Selecting groups of elements

```
.title {  
  background-color: gray;  
}
```

Team 1	Team 2
0	0
<b>+1</b>	<b>+1</b>



```
.title {  
  background-color: gray;  
}
```

...

```
<tr>  
  <td class="title">  
    Team 1  
  </td>  
  <td class="title">  
    Team 2  
  </td>  
</tr>
```

...

# Font Family

Changing fonts

```
p {  
  font-family: serif;  
}
```

In the great green room there was a  
telephone and a red balloon.

```
p {  
  font-family: sans-serif;  
}
```

In the great green room there was a telephone and a red balloon.

```
p {  
  font-family: monospace;  
}
```

In the great green room  
there was a telephone and  
a red balloon.

# Backgrounds

Changing colors and images

```
body {  
    background-color: seagreen;  
}
```

```
body {  
    background-color: #FFB6C1;  
}
```



```
body {  
    background-color: rgb(255,255,255);  
}
```

# Size

Resizing elements

```
img {  
  width: 50px;  
}
```



```
img {  
  width: 200px;  
}
```



```
img {  
  width: 600px;  
}
```



```
img {  
  width: 600px;  
  height: 300px;  
}
```



# Your turn!

In **scoreboard.html**, experiment by adding your choice of...

- An id, class, and type selector
- A resized image
- A background image or color
- Updated font size and family
- Something else!

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



# JavaScript

Making web pages dynamic

# querySelector

Selecting elements from HTML

```
document.querySelector('#add1');
```

Find element with id "add1"

```
document.querySelector('img');
```

Find first **img** element

# Event Listeners

Defining what should happen when the user interacts with our site

```
document.querySelector('button');
```

Find first **button** element

```
document.querySelector('button').addEventListener(...);
```

Add an event listener to the button element we selected

```
addEventListener(event, function)
```

When **event** happens, run **function**.



```
addEventListener('click', function)
```

```
addEventListener('mouseover', function)
```

```
addEventListener(event, changeColor)
```

```
addEventListener(event, updateScore)
```

```
function changeColor() {  
    document.querySelector('body').style.backgroundColor = 'blue';  
}  
  
document.querySelector('button').addEventListener('click', changeColor);
```

```
document.querySelector('button').addEventListener('click', function() {  
    document.querySelector('body').style.backgroundColor = 'blue';  
});
```

```
document.querySelector('button').addEventListener('click', function() {  
    document.querySelector('body').style.backgroundColor = 'blue';  
});
```

```
document.querySelector('button').onclick = function() {  
    document.querySelector('body').style.backgroundColor = 'blue';  
};
```



```
document.querySelector('button').onclick = function() {  
    document.querySelector('body').style.backgroundColor = 'blue';  
};
```

# Conditionals

Syntax for branching logic

```
if (x < y) {  
    alert("x is less than y");  
}
```

```
if (x < y) {  
    alert("x is less than y");  
} else {  
    alert("x is at least as big as y");  
}
```

# Your turn!

In **scoreboard.html**, experiment by implementing your choice of...

- An alert, via `alert("This is an alert!")`, when a team achieves some number of points
- A message, within the HTML, when a team achieves some number of points
- A button to reset the scores
- Something else of your choosing!

# Shopping

Creating a shopping list

# Your turn!

In **shopping.html**, experiment by implementing your choice of...

- A button to reset the shopping cart
- An error message when a user attempts to add an empty item
- Something else of your choosing!

# Bootstrap

Sprucing up your site



# Office Hours

# Git, GitHub, Software Licences, and the World of Open-Source Software

Today, 6:00 PM, WCC 1010