

Intro to Web Development

Chandra Pratama
Freelance

About me!

1. Indraprasta PGRI University
2. IPB University

About me!



1. Indraprasta PGRI University Class of 2016
2. IPB University Class of 2010

About me!



IPB University
— Bogor Indonesia —

1. Indraprasta PGRI University class of 2016
2. IPB University Class of 2010

About me!

1. Indraprasta PGRI University class of 2016
2. IPB University Class of 2010



IPB University
— Bogor Indonesia —

Where I've worked

Where I've worked

1. Freelance*
2. Luki Silk Nuetrustie*
3. Panca Duta Prakarsa*

* co-op

Where I've worked



1. Freelance*
2. Luki Silk Nuetrustie - Office Administrator*
3. Panca Duta Prakarsa*

* co-op

Where I've worked



1. Freelance*
2. Luki Silk Nuetrustie*
3. Panca Duta Prakarsa- Project Admin*

* co-op

Where I work now

Freelance



Day to day
Freelance

Learning code
Writing code
Reviewing code
Pair programming



Languages & Frameworks E

HTML

CSS

JavaScript

GraphQL

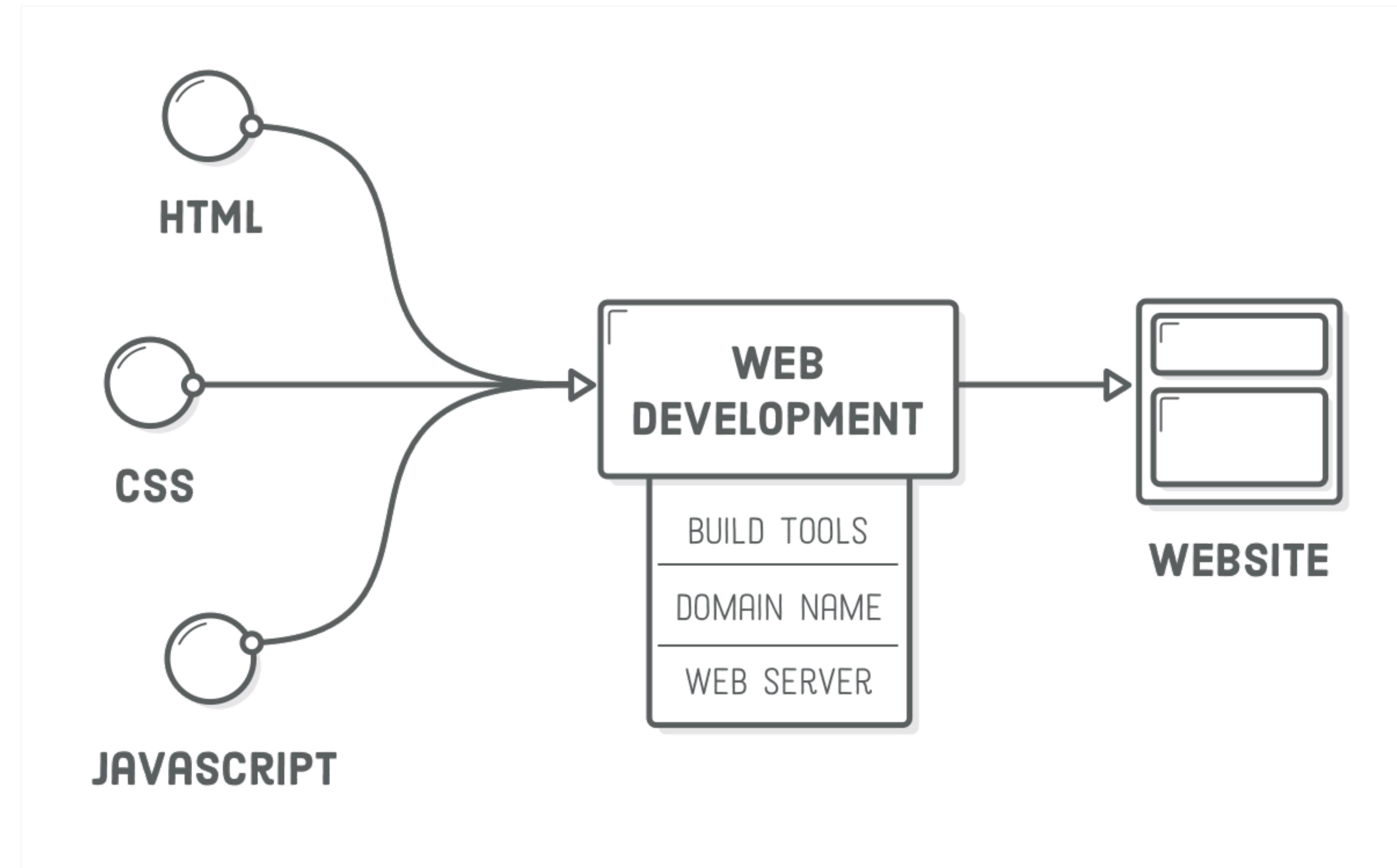
React

Gatsby

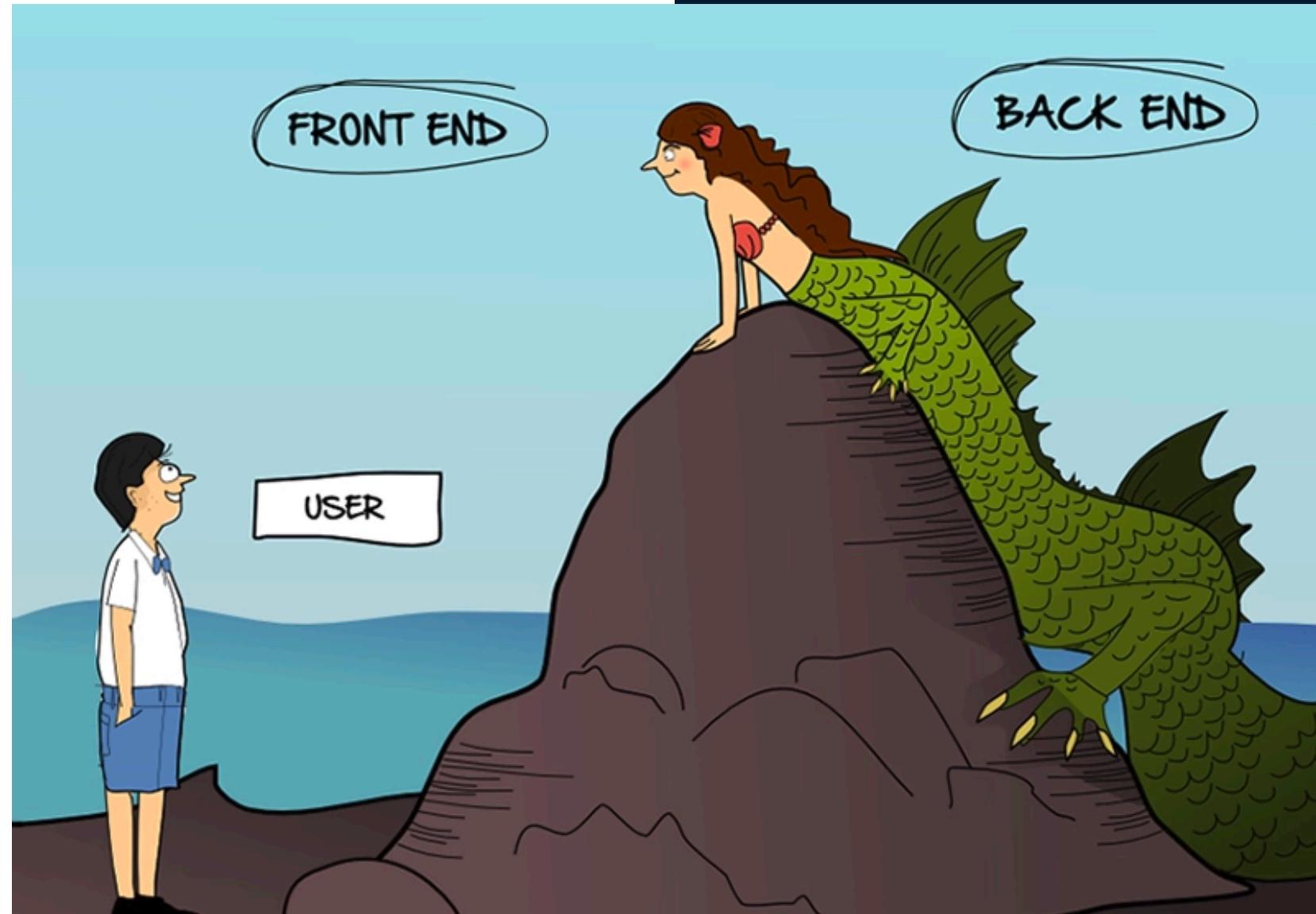
Ant

Foundations of Web Development

HTML + CSS + JavaScript



Front End

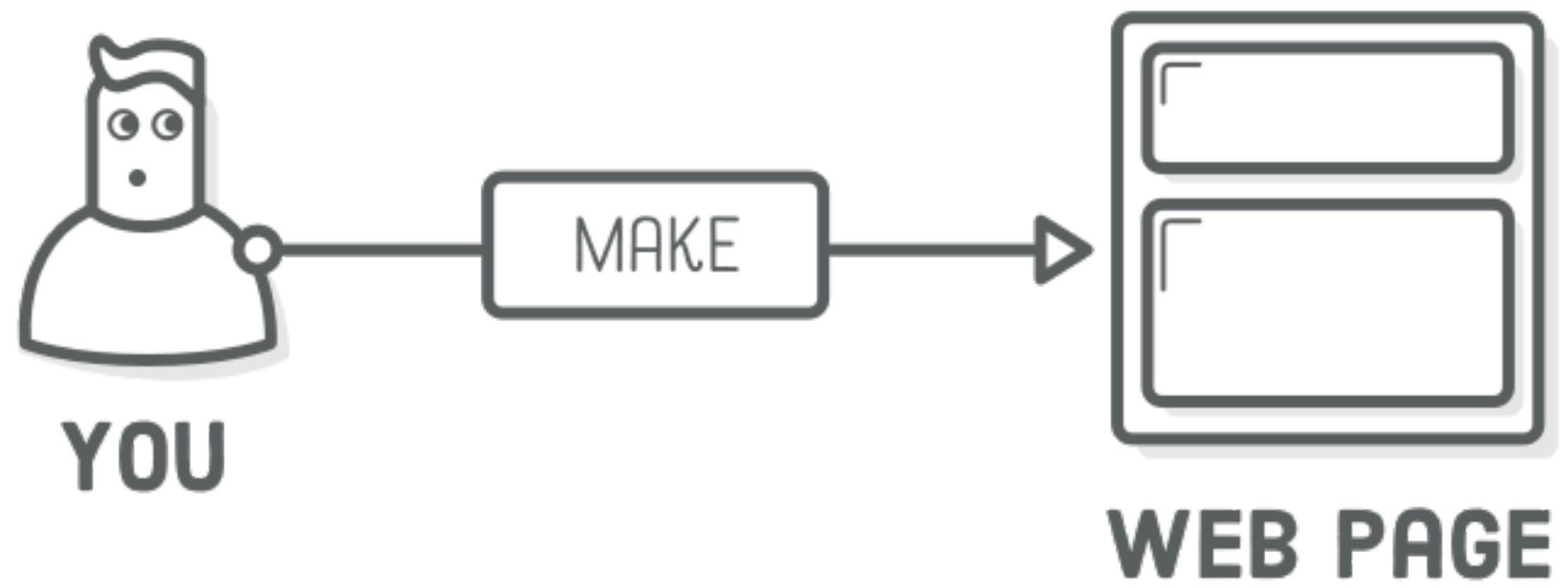


Back End

- Refers to what is loaded by the user's browser
- Deals with the visuals — making it look good and easy to use
- Deals with user interface (UI) and user experience (UX)
- Great for those interested in design

- Deals with behind the scenes functionality of the site
- Configuring servers and databases to power site
- More involved programming logic & data handling
- Great for those more interested in the logical side of programming or working with data

Today's goal!



Let's get started!

What happens when a webpage loads?

1. Website makes a request to a server
2. Your browser downloads code from the server
3. Your browser builds the page
4. You see the page!



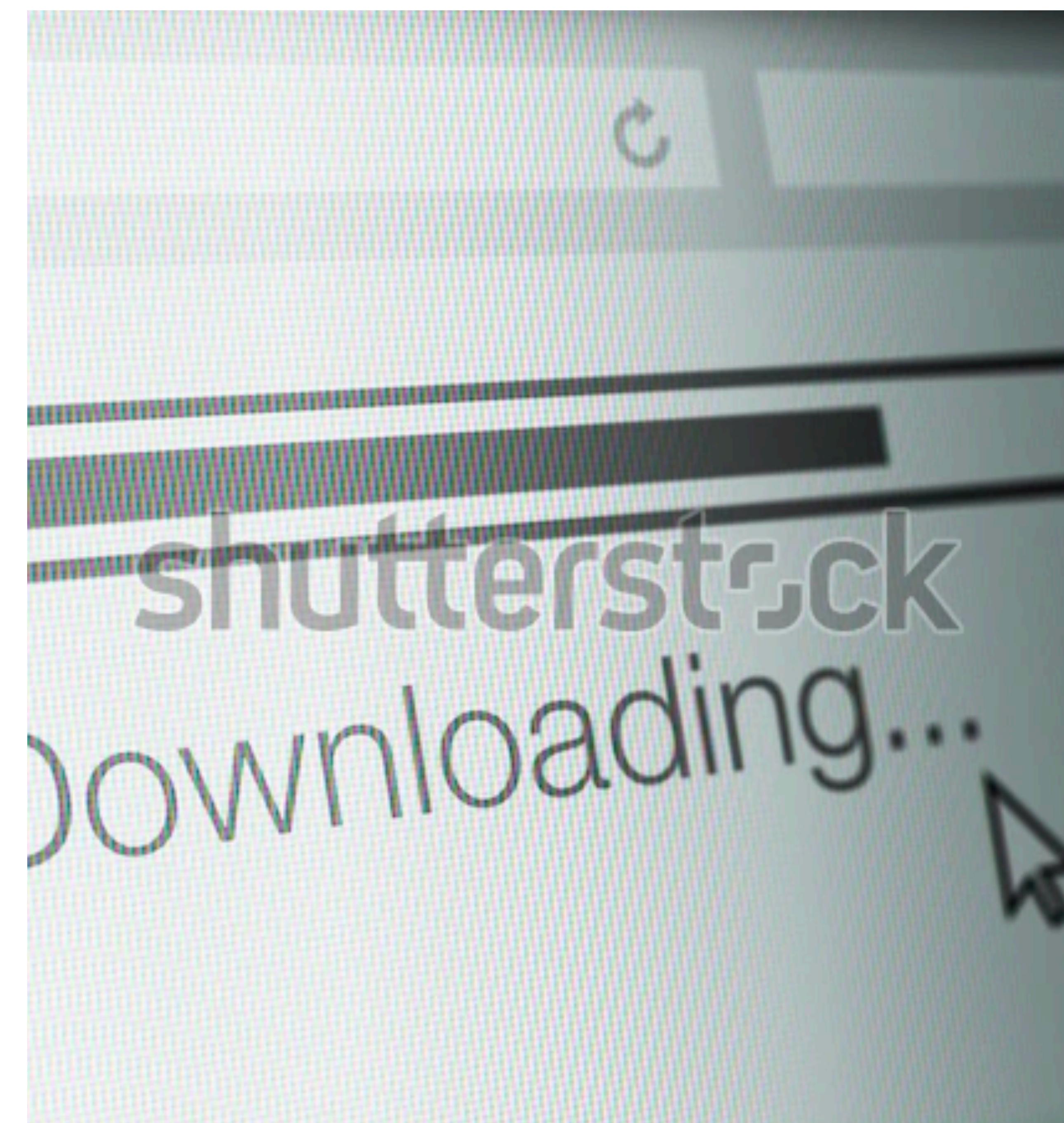
What happens when a webpage loads?

1. Website makes a request to a server
2. Your browser downloads code from the server
3. Your browser builds the page
4. You see the page!



What happens when a webpage loads?

1. Website makes a request to a server
2. Your browser downloads code from the server
3. Your browser builds the page
4. You see the page!



What happens when a webpage loads?

1. Website makes a request to a server
2. Your browser downloads code from the server
3. Your browser builds the page
4. You see the page!



What happens when a webpage loads?

1. Website makes a request to a server
2. Your browser downloads code from the server
3. Your browser builds the page
4. You see the page!





www.shutterstock.com · 603263660

HTML

Foundation

CSS

Aesthetics

JavaScript

Bells & Whistles

HTML



Emma Wedekind 🎀
@EmmaWedekind



HTML: Why does my cat only have 3 legs?

CSS: Why is my cat purple?

JS: Why is my cat walking sideways?

4:05 AM · Jul 24, 2019 · Twitter for iPhone

Foundation

CSS

Aesthetics

JavaScript

Bells & Whistles



HTML

Framing & Construction

CSS

Painting & Decoration

JavaScript

Electricity & Plumbing



HTML

Framing & Construction

CSS

Painting & Decoration

JavaScript

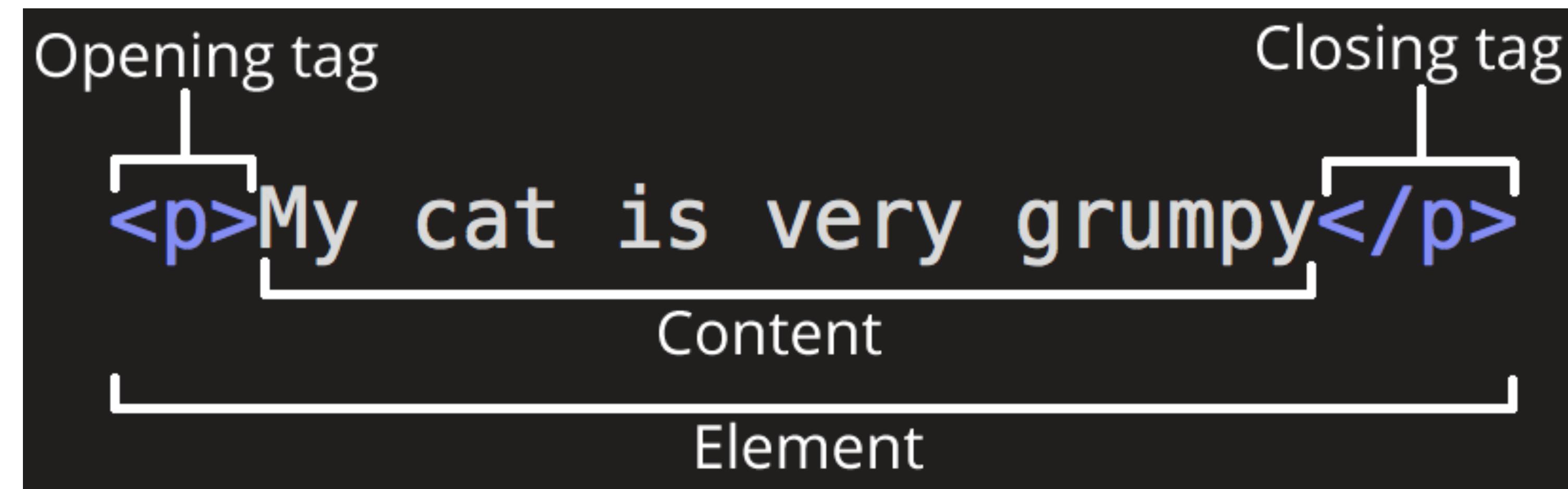
Electricity & Plumbing

HTML

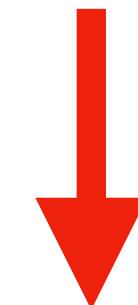
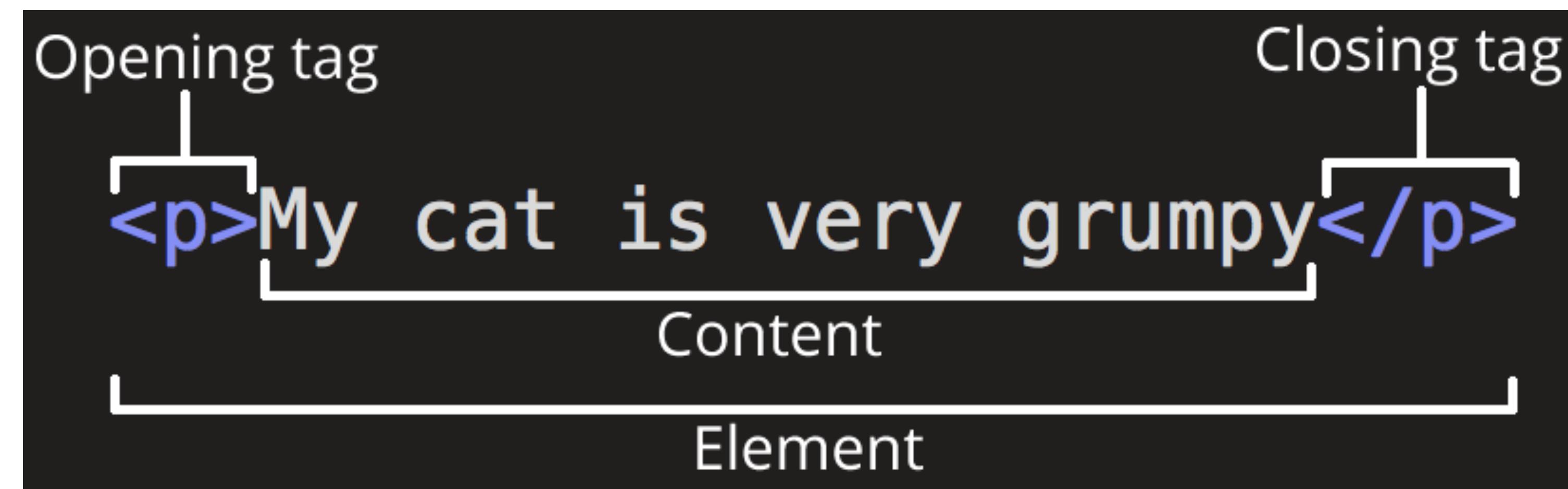
Hypertext Markup Language



HTML Elements are just opening and closing tags



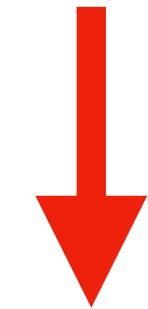
HTML Elements are just opening and closing tags



My cat is very grumpy

Attributes let us group HTML elements

```
Attribute  
<p class="editor-note">My cat is very grumpy</p>
```



My cat is very grumpy

Heading Tags (h1, h2, h3, h4, h5, h6)



```
<h1>My main title</h1>  
  
<h2>My top level heading</h2>  
  
<h3>My subheading</h3>  
  
<h4>My sub-subheading</h4>
```

My main title
My top level heading
My subheading
My sub-subheading

Paragraphs (<p>), Sections (<section>), Spans ()



```
<section>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit</p>

  <span>Spans</span>
  <span>display</span>
  <span>inline</span>
  <span>by</span>
  <span>default</span>

  <p>
    <span>I'm a span inside a paragraph!</span>
  <p>
</section>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit
 Spans display inline by default
 I'm a span inside a paragraph!

Lists (,) & List Items ()



```
<h2>Ordered List</h2>
<ol>
  <li>thing 1</li>
  <li>thing 2</li>
  <li>thing 3</li>
</ol>

<h2>Unordered List</h2>
<ul>
  <li>technologists</li>
  <li>thinkers</li>
  <li>builders</li>
</ul>
```

Ordered List

- 1. thing 1
- 2. thing 2
- 3. thing 3

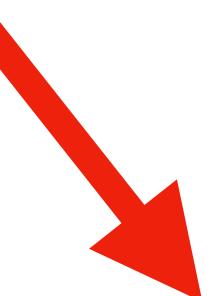
Unordered List

- technologists
- thinkers
- builders

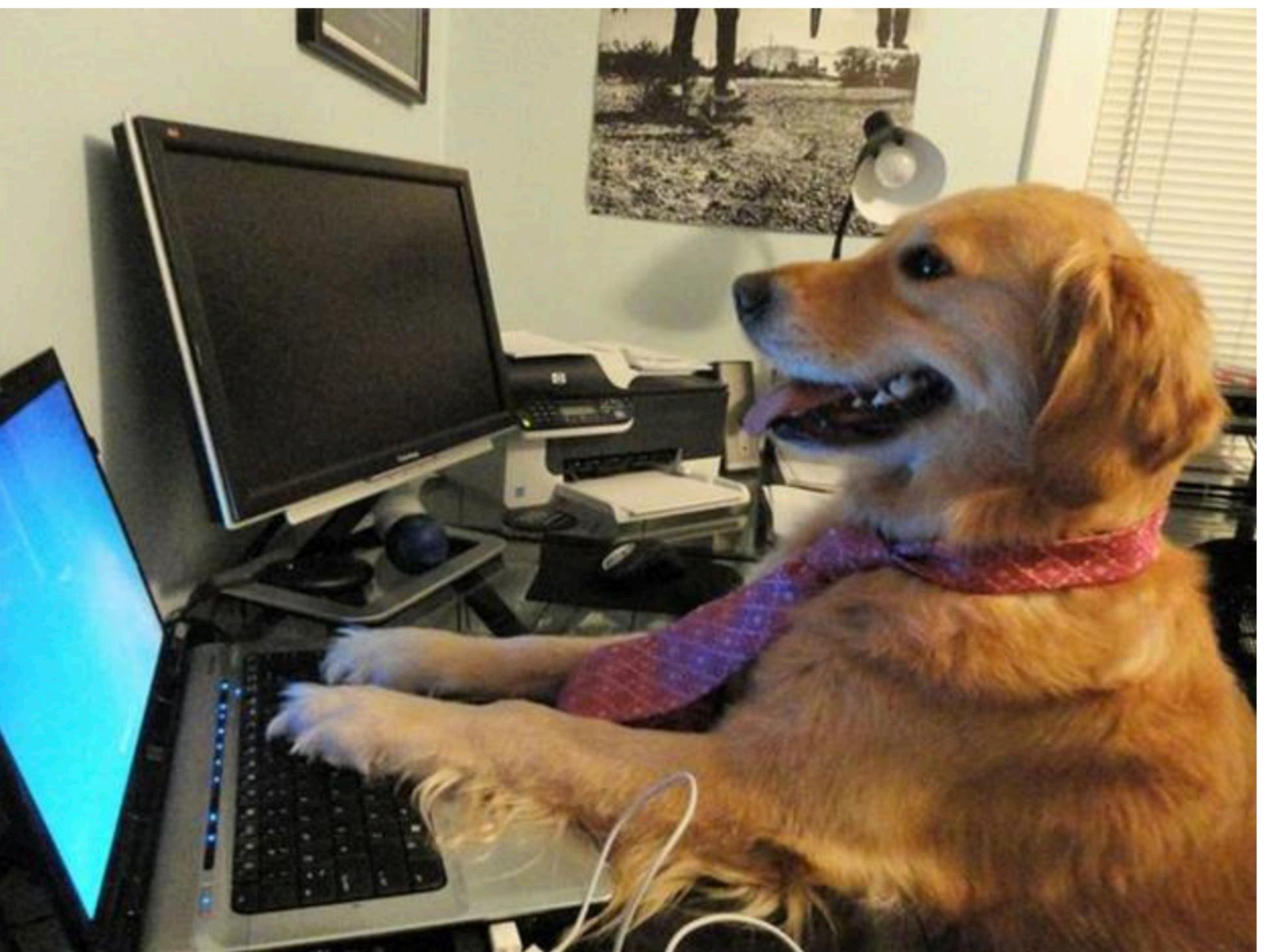
Links () & Images ()

```
● ● ●  
Google Something  

```



[Google Something](#)

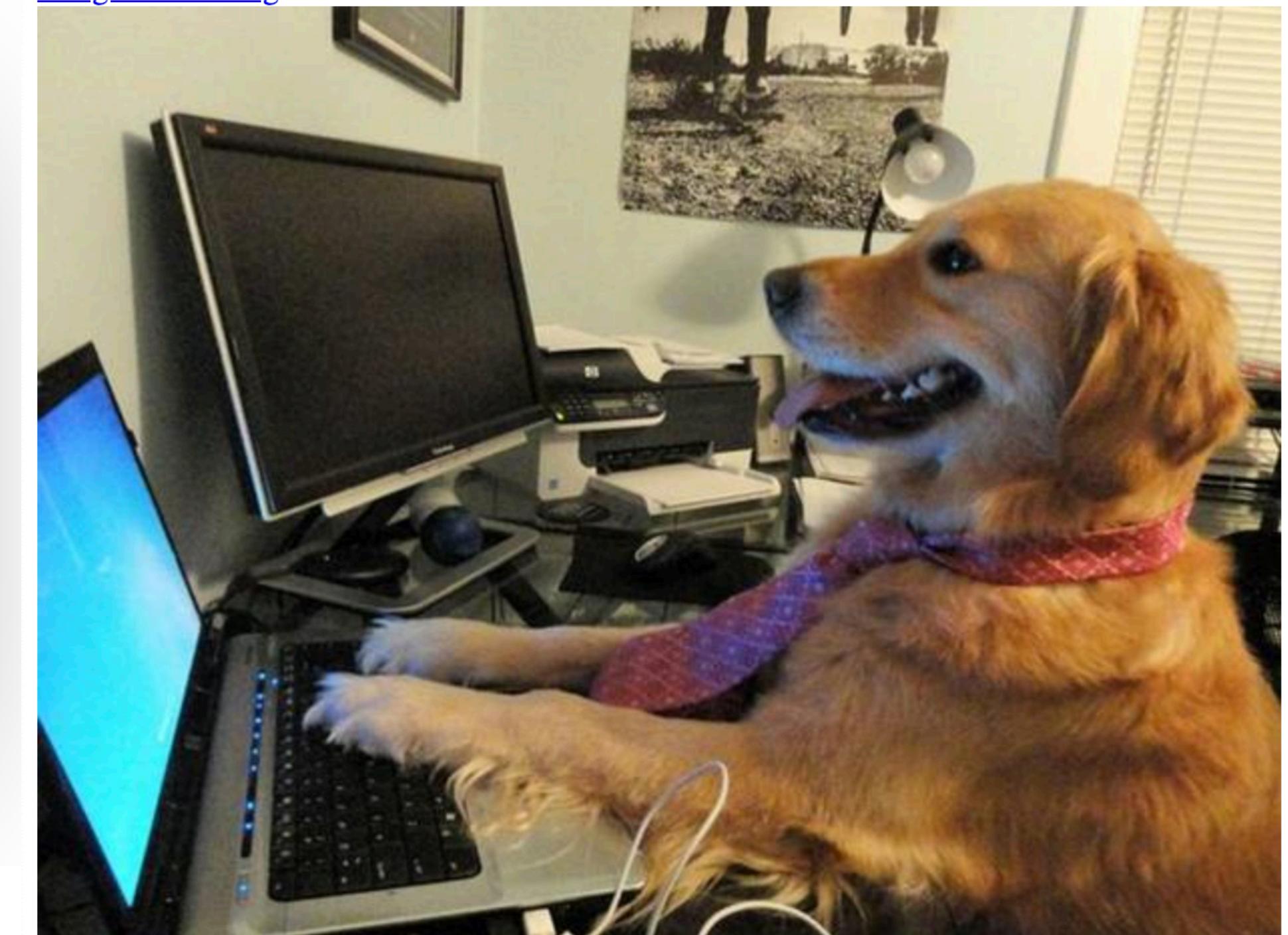


Divs

The `<div>` tag defines a division or a section in an HTML document

```
● ● ●  
<div>  
  <a href="https://google.com">Google Something</a>  
</div>  
  
<div>  
    
</div>
```

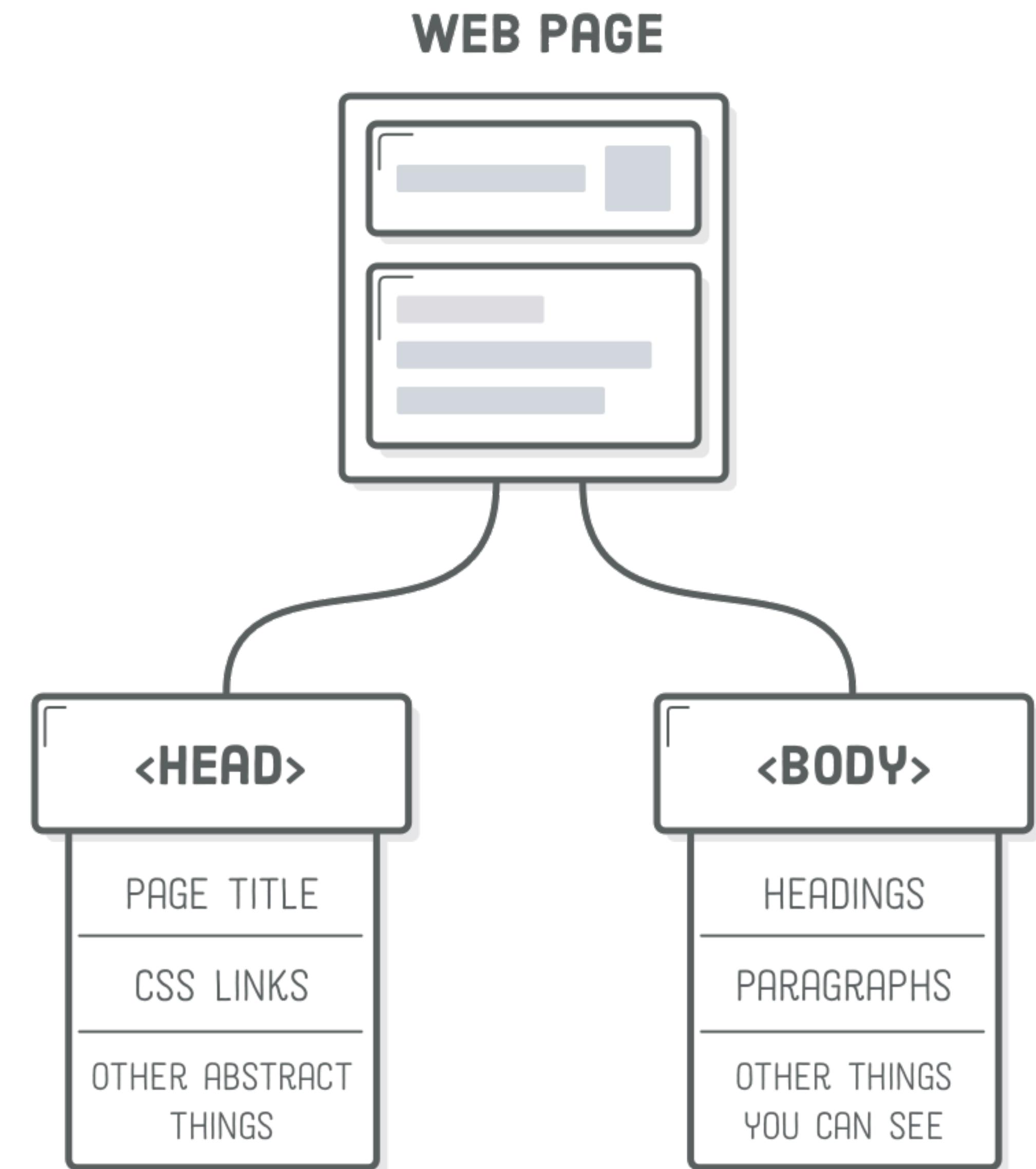
[Google Something](#)



Structure of a web page

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Metadata goes here -->
  </head>
  <body>
    <!-- Content goes here -->
  </body>
</html>
```



A simple `index.html` file

Header section

The main content of the site!

Footer things

```
● ● ●

<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>

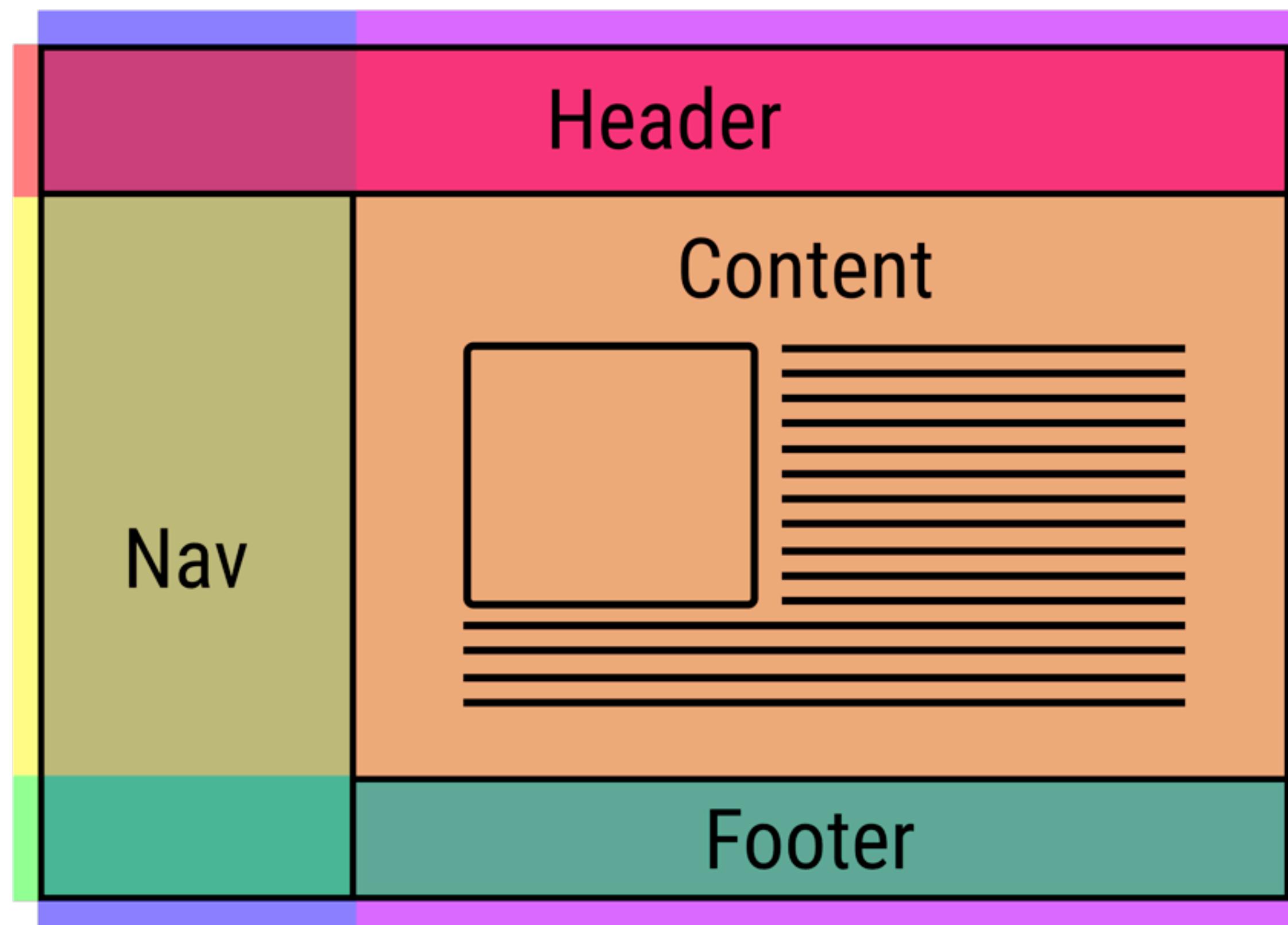
  <body>
    <!-- Put your HTML elements in here -->

    <header>
      Header section
    </header>

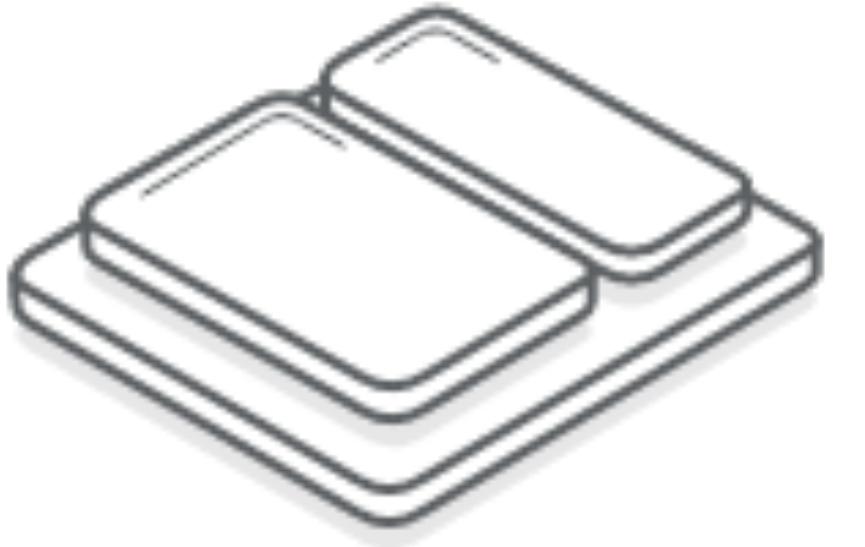
    <main>
      <p>The main content of your site!</p>
    </main>

    <footer>
      Footer things
    </footer>
  </body>
</html>
```

Laying out a webpage

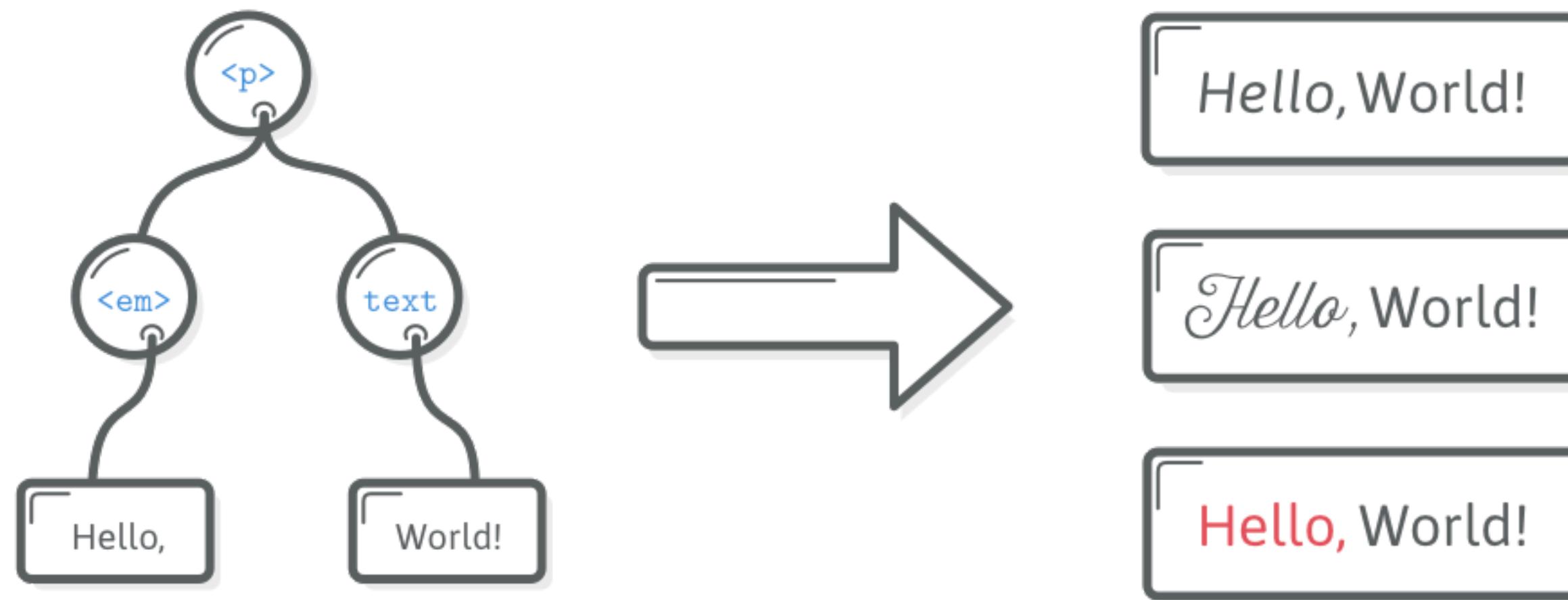


Every element in HTML is a rectangular box



Most of the HTML elements on your page can be thought of as boxes sitting on top of each other. You can use CSS to set their size, color, position, etc.





HTML
(STRUCTURE)

CSS
(PRESENTATION)

CSS

Cascading Style Sheets



Stylesheet = File containing CSS

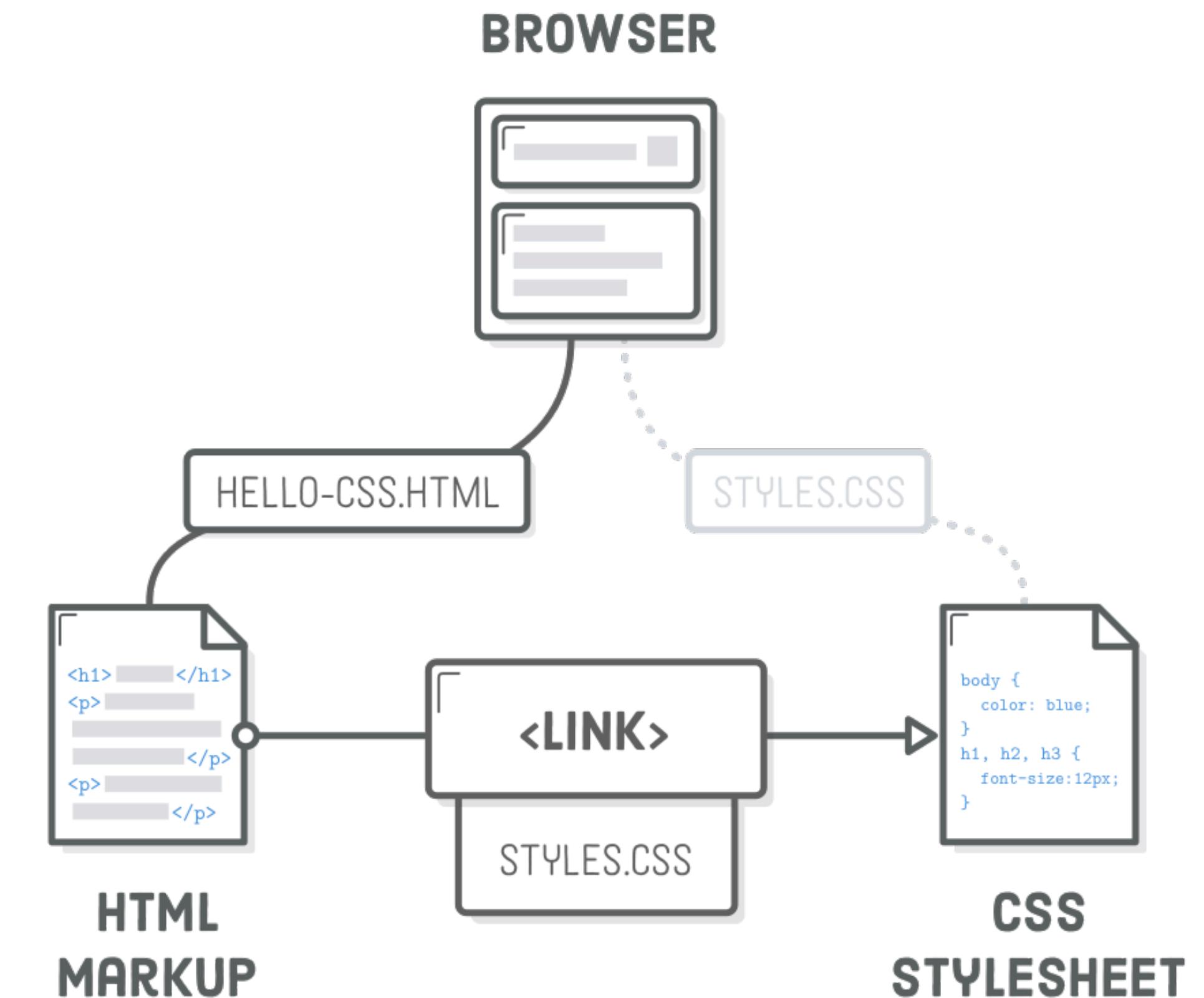
Stylesheet = File containing CSS

style.css

Linking a CSS Stylesheet to your HTML document



```
<head>
  <meta charset='UTF-8' />
  <title>Hello, CSS</title>
  <link rel='stylesheet' href='styles.css' />
</head>
```



HTML (index.html)



```
<p>I'm a paragraph</p>
```

CSS (style.css)

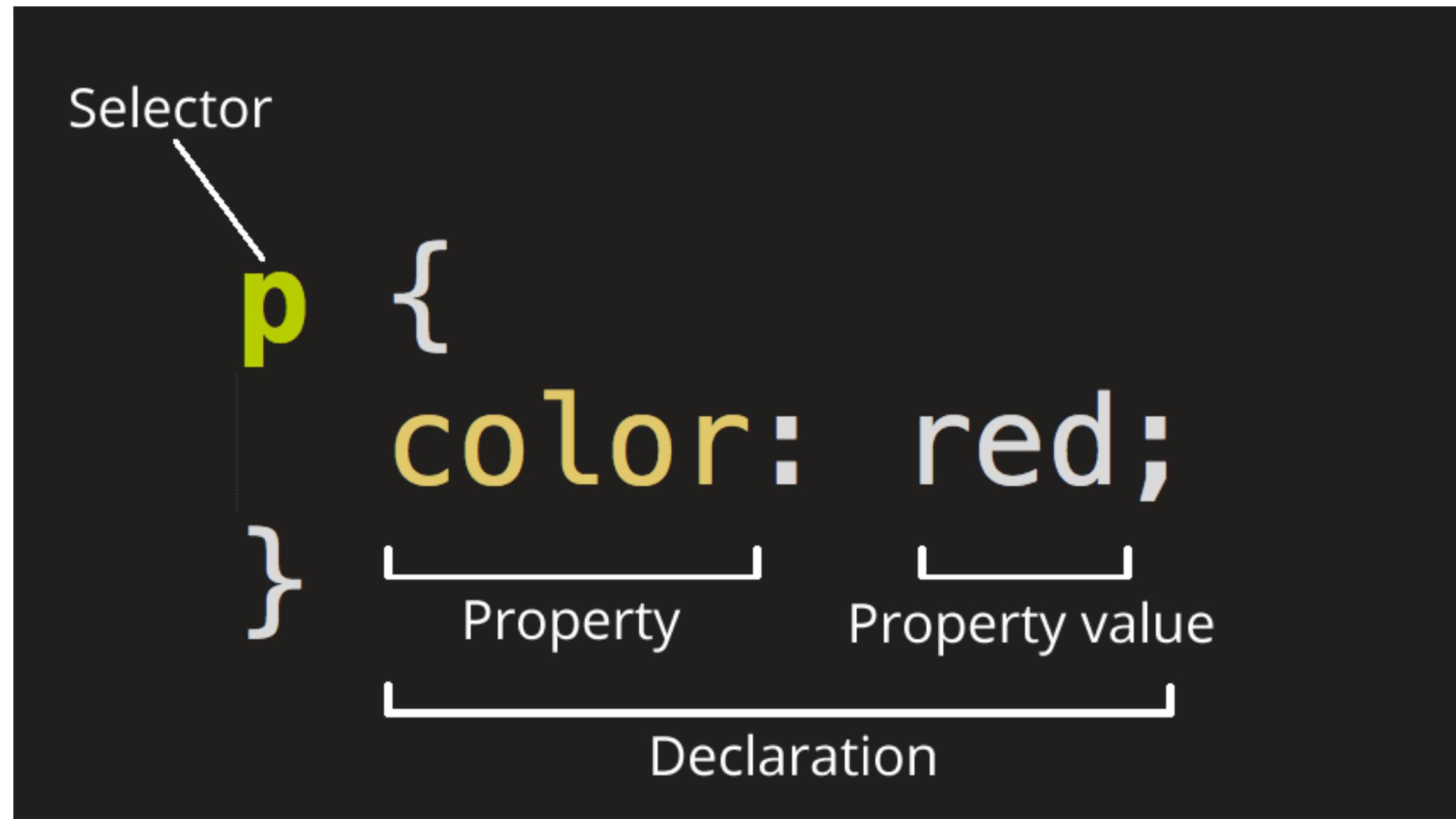


```
p {  
  color: red;  
  font-size: 50px;  
  font-family: 'Roboto', sans-serif;  
}
```

Result (in the browser)

I'm a paragraph

“Select all the paragraph elements and turn the text within them red”



Common CSS Properties

color

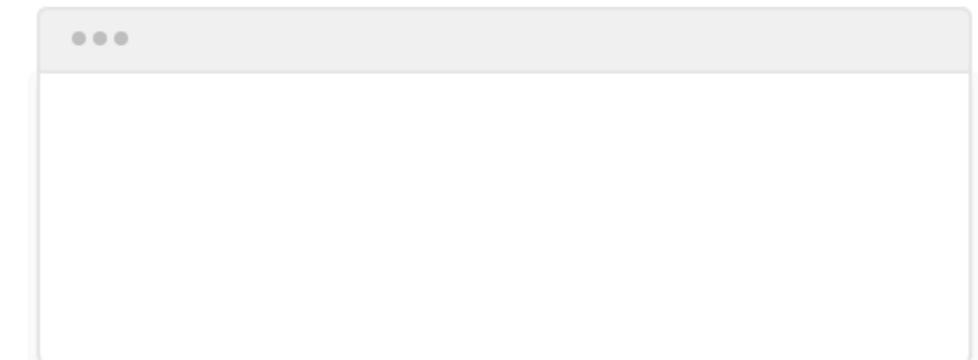
color

In collection: [Typography](#) [Permalink](#) [Share](#) [Can I use](#) [MDN](#)

Defines the color of the **text**.

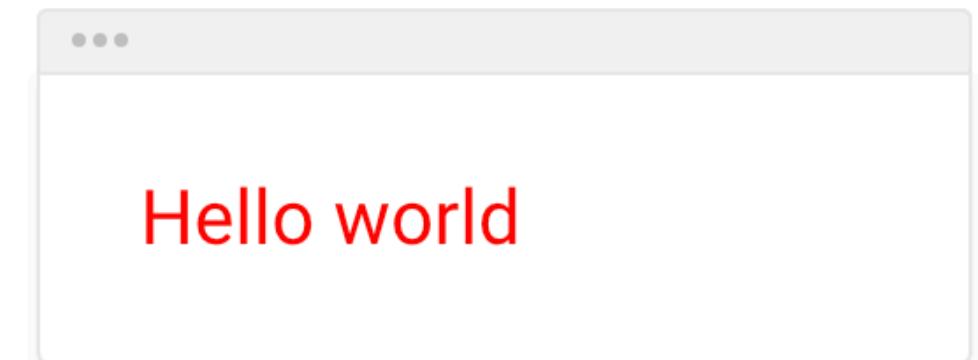
```
color: transparent;
```

Applies a **transparent** color to the text. The text will still take up the space it should.



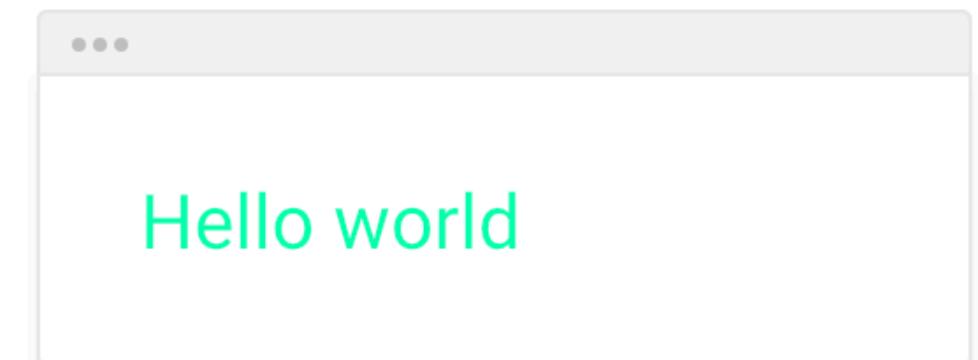
```
color: red;
```

You can use one of the [140+ color names](#).



```
color: #05ffb0;
```

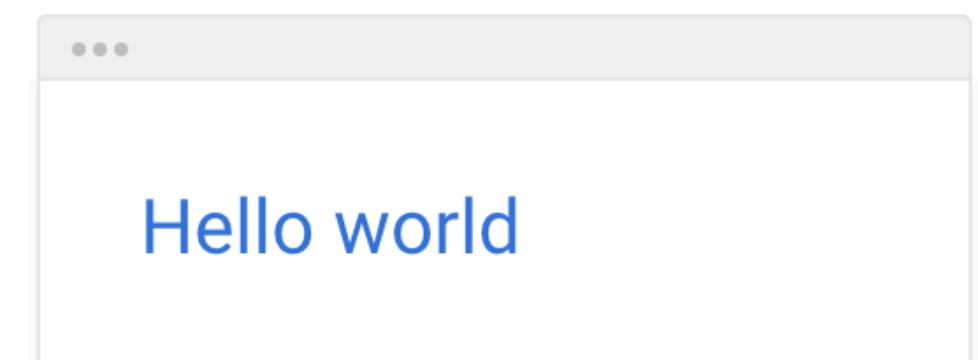
You can use **hexadecimal** color codes.



```
color: rgb(50, 115, 220);
```

You can use **rgb()** color codes:

- the first value is for `red`
- the second value is for `green`
- the third value is for `blue`



Each of them can have a value between **0** and **255**.

Common CSS Properties

background-color

background-color

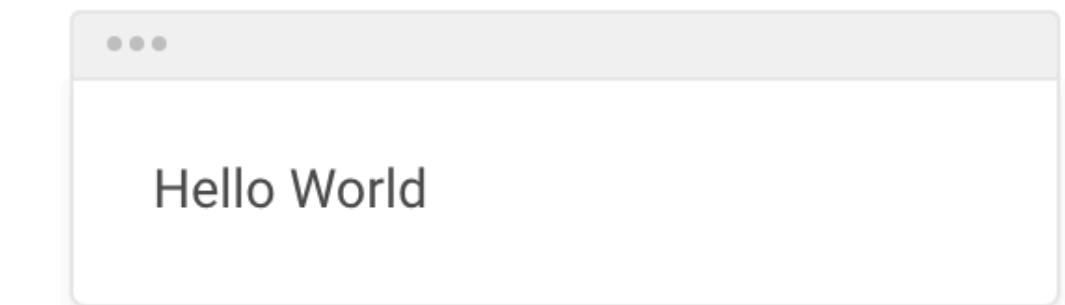
In collection: [Backgrounds](#) [Permalink](#) [Share](#) [MDN](#)

Defines the color of the element's background.

`background-color: transparent;`

default

By default, the background color is **transparent**, basically meaning that there is **no** background color.



`background-color: red;`

You can use one of the [140+ color names](#).



`background-color: #05ffb0;`

You can use **hexadecimal** color codes.

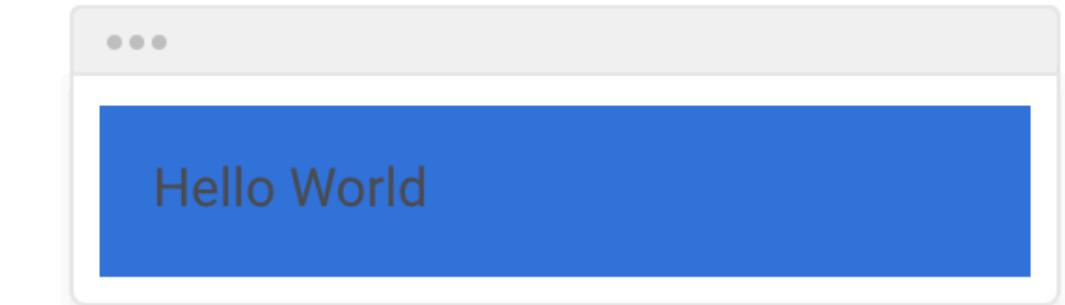


`background-color: rgb(50, 115, 220);`

You can use **rgb()** color codes:

- the first value is for `red`
- the second value is for `green`
- the third value is for `blue`

Each of them can have a value between **0** and **255**.



Common CSS Properties

font-size

font-size

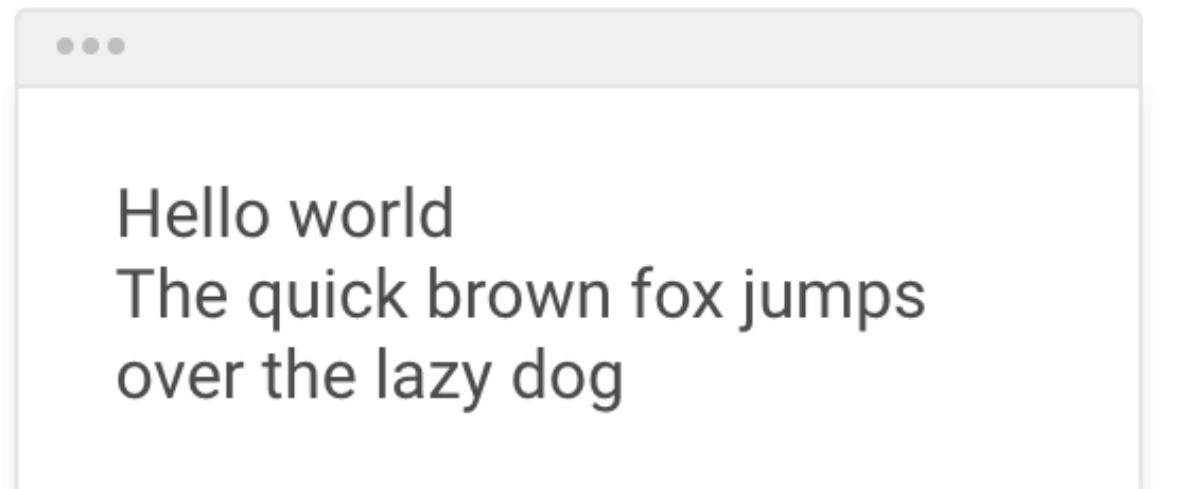
In collection: [Typography](#) [Permalink](#) [Share](#) [MDN](#)

Defines the size of the text.

`font-size: medium;`

`default`

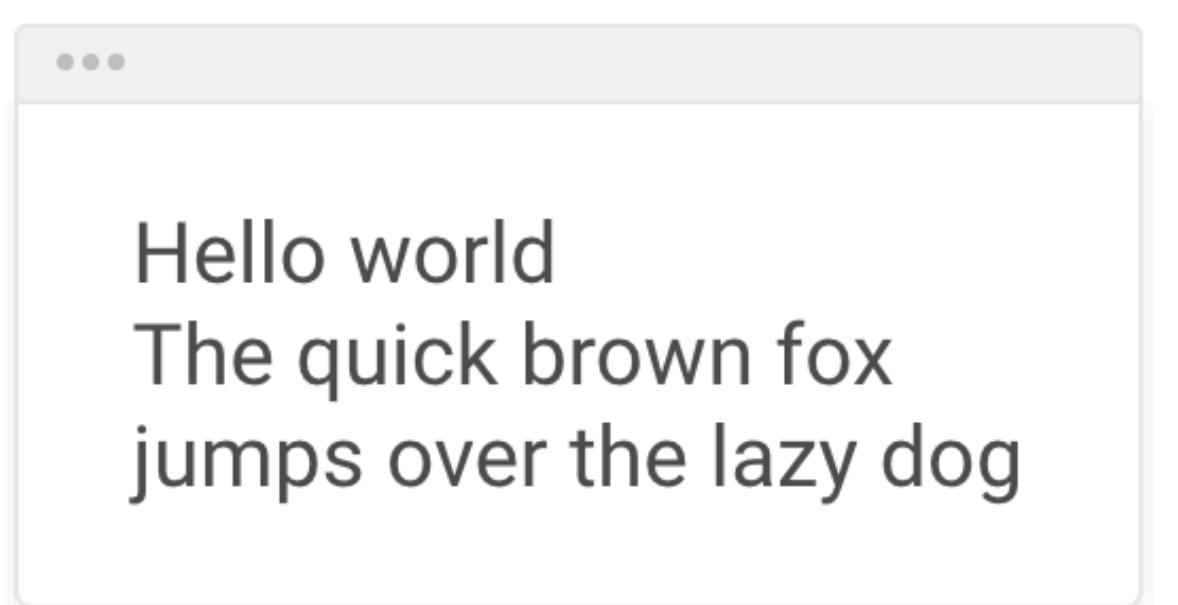
The text will use the browser's default **medium** size.



Hello world
The quick brown fox jumps
over the lazy dog

`font-size: 20px;`

You can use **pixel** values.



Hello world
The quick brown fox
jumps over the lazy dog

Common CSS Properties

border

border

[Permalink](#) [Share](#) [MDN](#)

Shorthand property for [border-width](#) [border-style](#) and [border-color](#).

```
border: 4px dotted red;
```

The **order** is important:

- width
- style
- color



```
border: 2px solid;
```

Only the *color* is **optional**. If you omit it, the color applied will be the color of the **text**.



Common CSS Properties

width

width

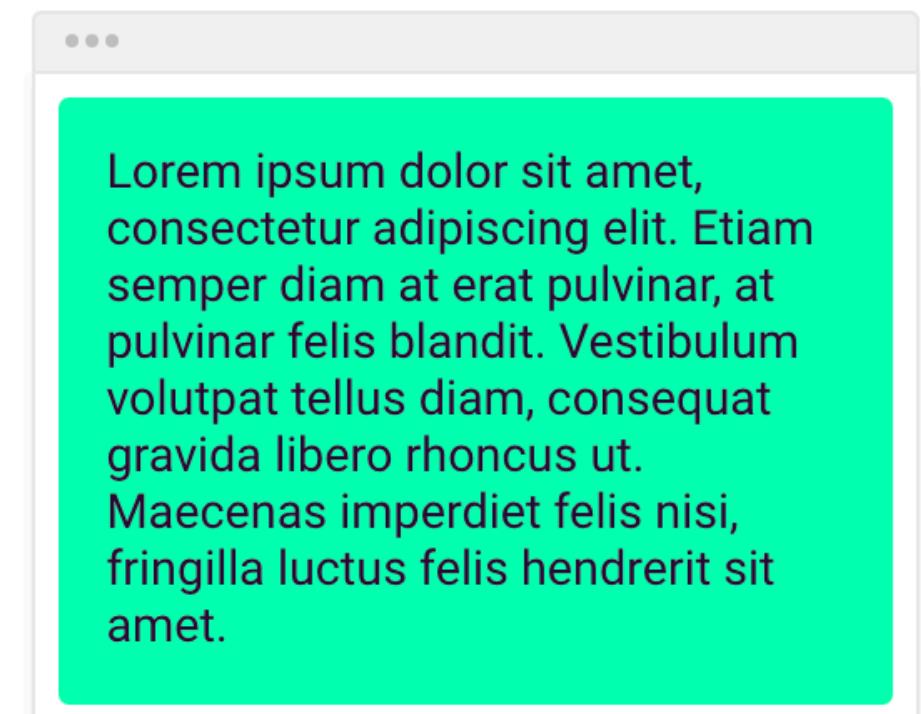
In collection: [Box-Model](#) [Permalink](#) [Share](#) [MDN](#)

Defines the width of the element.

width: auto;

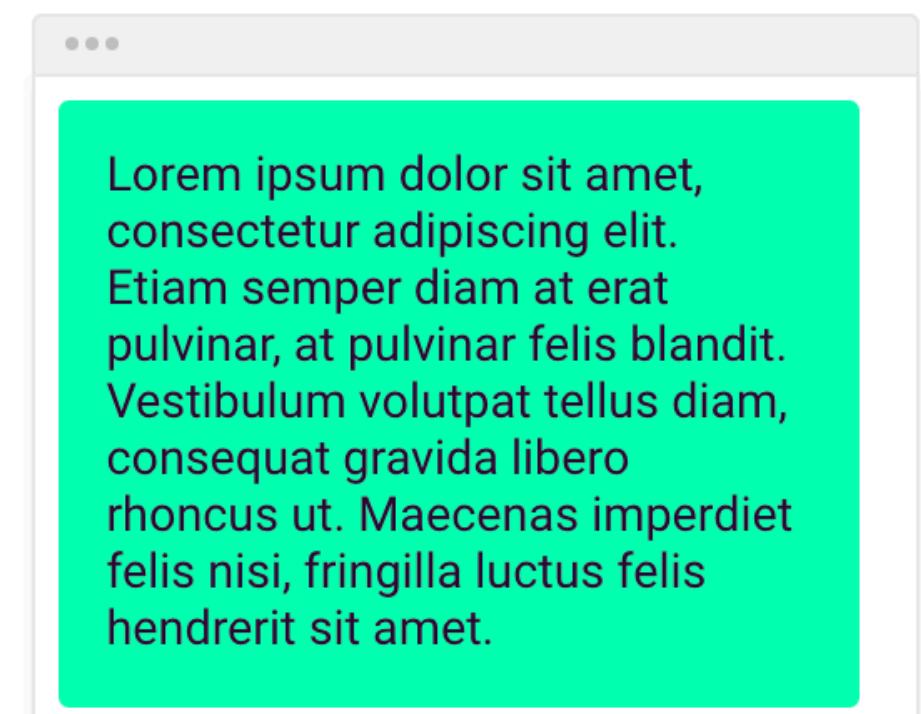
default

The element will **automatically** adjust its width to allow its content to be displayed correctly.



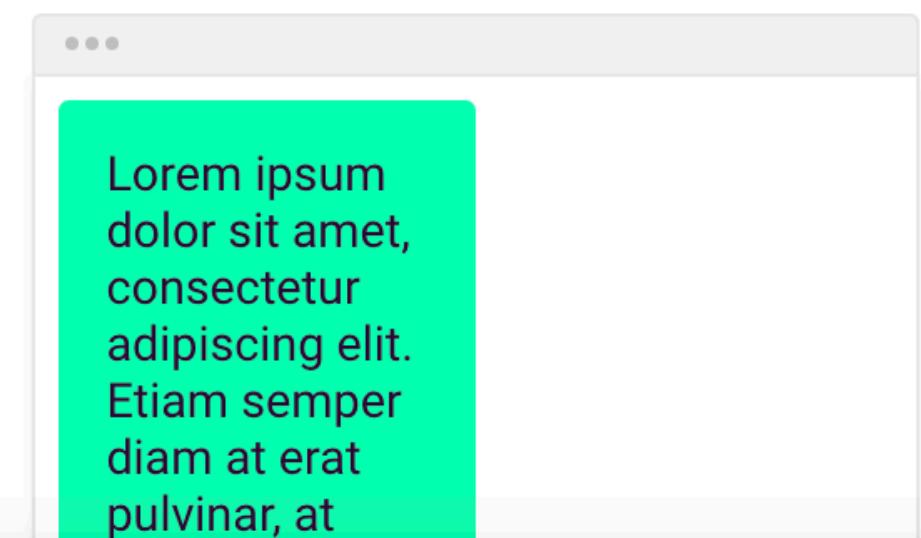
width: 240px;

You can use **numeric values** like **pixels, (r)em, percentages...**



width: 50%;

If you use **percentages**, the value is relative to the container's width.



Multiple properties & Multiple selectors



```
p {  
  color: red;  
  width: 500px;  
  border: 1px solid black;  
}
```



```
p, li, h1 {  
  color: red;  
}
```

Multiple properties & Multiple selectors



```
● ● ●  
p {  
  color: red;  
  width: 500px;  
  border: 1px solid black;  
}
```

```
● ● ●  
p, li, h1 {  
  color: red;  
}
```

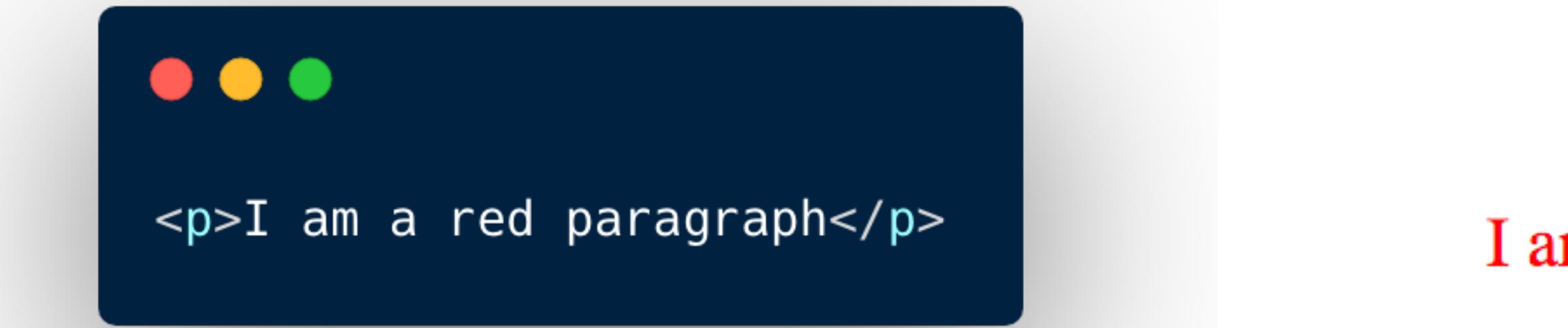
Multiple properties & selectors

```
● ● ●  
p {  
  color: red;  
  width: 500px;  
  border: 1px solid black;  
}
```

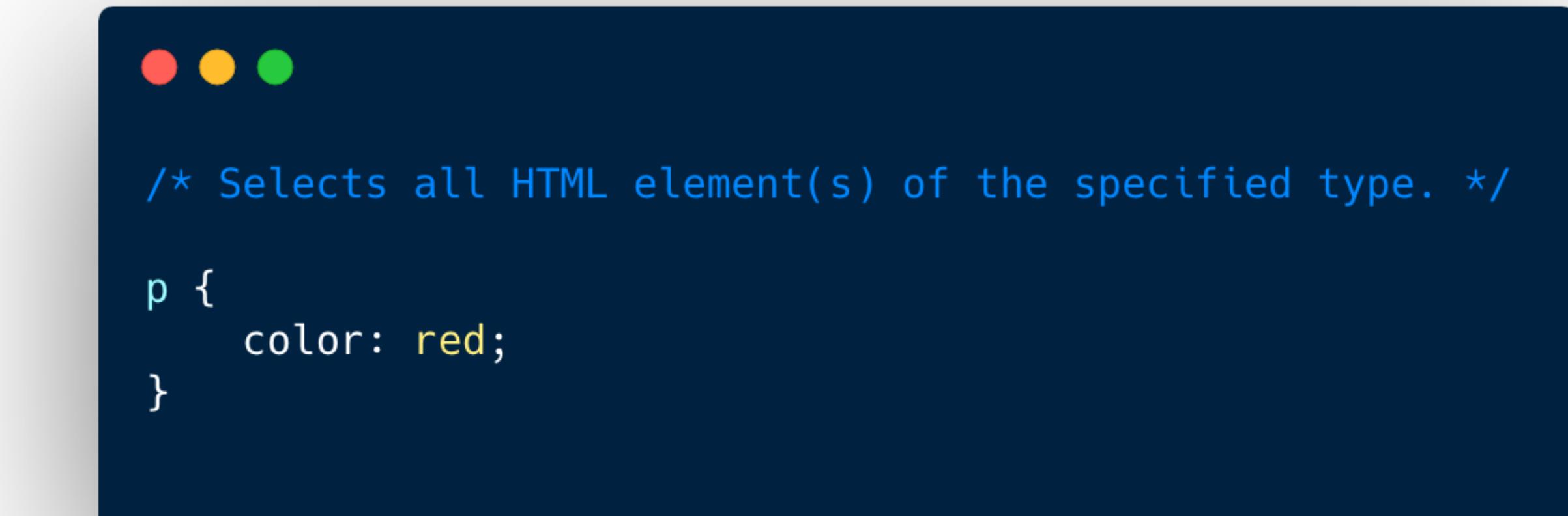
```
● ● ●  
p, li, h1 {  
  color: red;  
}
```

Element selectors (div, p, section, h1, etc...)

1. Element
2. ID
3. Class



A screenshot of a web browser window. At the top, there are three colored dots: red, yellow, and green. Below them, the HTML code <p>I am a red paragraph</p> is displayed in a dark blue box. To the right of the box, the text "I am a red paragraph" is shown in red.



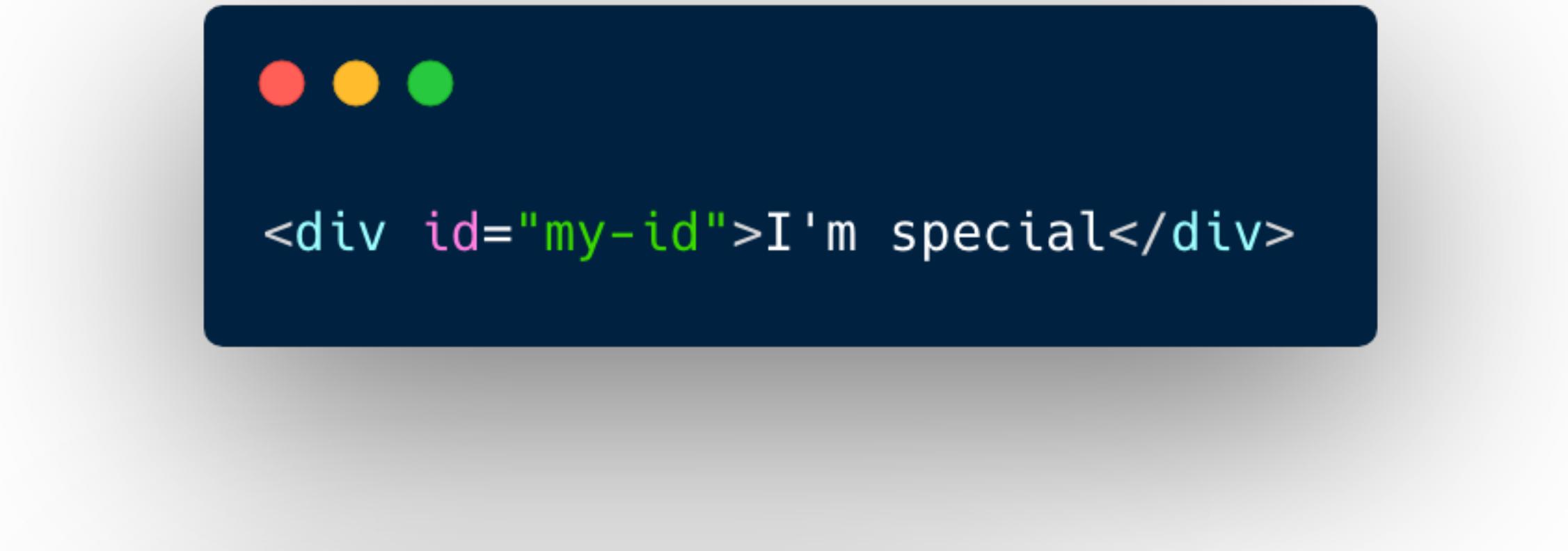
A screenshot of a web browser window. At the top, there are three colored dots: red, yellow, and green. Below them, a snippet of CSS code is shown in a dark blue box:

```
/* Selects all HTML element(s) of the specified type. */

p {
    color: red;
}
```

IDs — target one unique element with a particular ID

1. Element
2. ID
3. Class



I'm special

Classes - group a bunch of things together

1. Element
2. ID
3. Class

```
● ● ●  
  
<div class="my-class">I'm red</div>  
<div class="my-class">I'm red</div>  
<div>I'm not red</div>  
<div class="my-class">I'm red</div>
```

I'm red
I'm red
I'm not red
I'm red

```
● ● ●  
  
.my-class {  
    color: red;  
}
```

Demo

<https://codepen.io>

Your Task

Make the most fun/unique webpage you can on codepen.io in the next 15 minutes!

Requirements

1. At least 3 different HTML elements
2. At least 3 different CSS declarations
3. Submit your pen here:

<https://workshop-submissions.now.sh>

Helpful links

[htmlreference.io](#)

[cssreference.io](#)

[unsplash.com](#)

[google.com](#)

Q & A

Keep learning! Keep building!

More Resources

<https://htmlreference.io/>

<https://cssreference.io/>

<https://css-tricks.com/>

<http://www.dontfeartheinternet.com>

<https://internetingishard.com>

<https://www.freecodecamp.org/>

<https://www.codecademy.com>

<https://www.theodinproject.com>

<https://egghead.io>

There is no one tutorial or course that will teach you everything you need to learn — the best thing you can do to learn is to actually build things!

Thanks!

chandra.praputraa@gmail.com

github/chandrainf

Sources

<https://internetingishard.com>

<https://unsplash.com>

<https://internetingishard.com/>

<https://developer.mozilla.org>

<https://htmlreference.io/>

<https://cssreference.io/>

...and many more!