GENERAL ASSEMBLY

CODING FOR BEGINNERS: HTML AND CSS

By Chen Hui Jing / @hj_chen

GENERAL ASSEMBLY

ABOUT WEB DEVELOPMENT

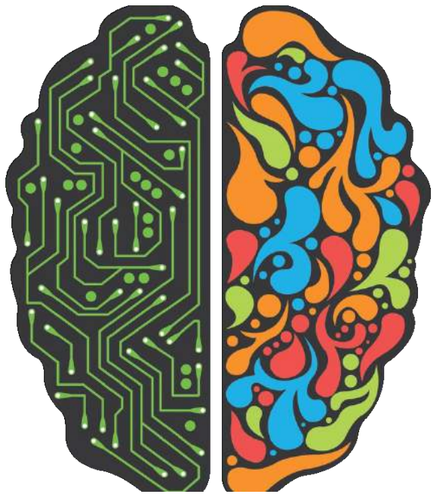
WHAT IS WEB DEVELOPMENT?

The process of building websites

HOW TO GET STARTED?

Your computer

A clear mind



CODE IS NUMBERS, LETTERS AND SYMBOLS

Regardless of what programming language you use, **all** code can be read in **any** text editor.

Javascript

```
close: function () {
  this.ul.setAttribute
  this.index = -1;

  $.fire(this.input, "
},
```

Code credit: Lea Verou

C

```
#include "8cc.h"
static int count_leadi
  for (int i = 7; i >=
      if ((c & (1 << i
          return 7 - i
  return 8;
}
```

Code credit: Rui Ueyama

Assembly

```
ctable   segment para
  db    9 dup(' ')
  db    9,10,' ',12,13
  db    13 dup(' ')
  db    27
  db    4 dup(' ')
  db    ' !"#$%&',39,'
  db    'ABCDEFGHIJKLM
```

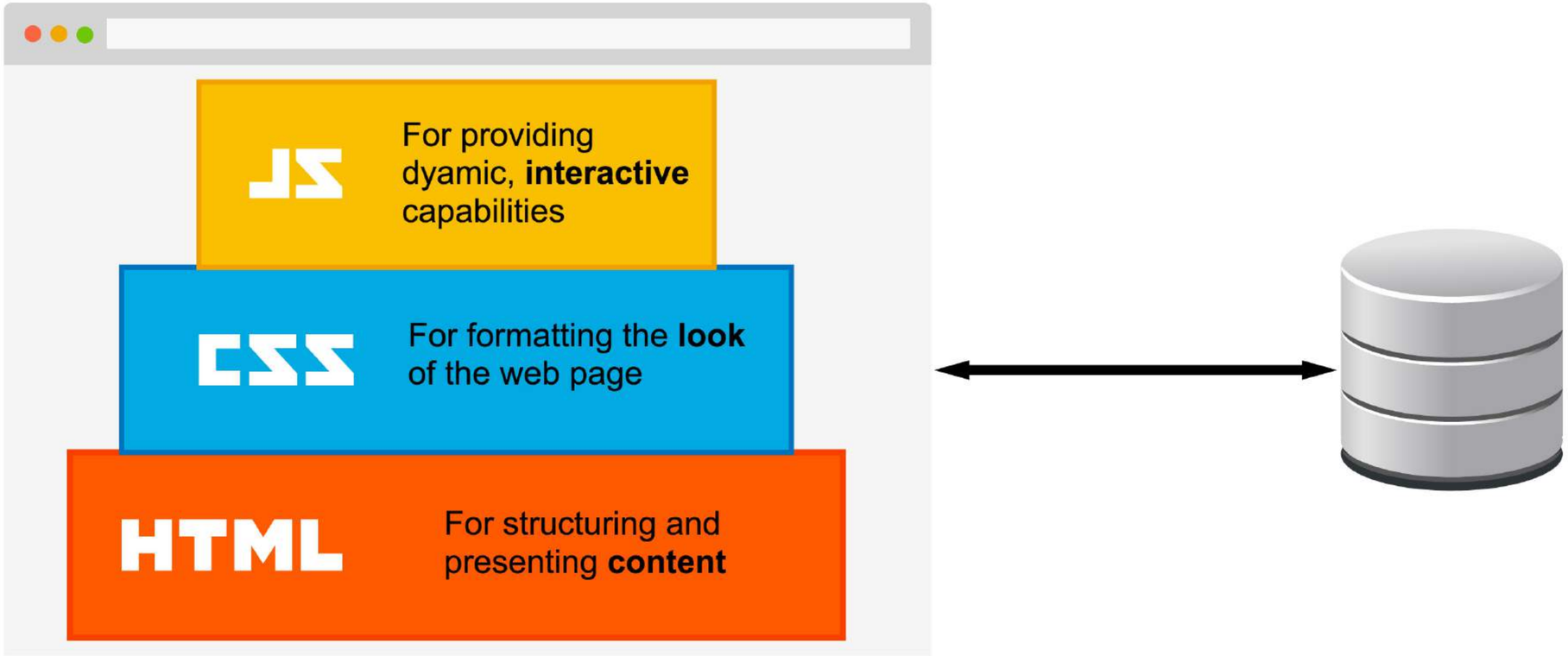
Code credit: Happy codings

HTML AND CSS ARE THE FOUNDATION OF THE WEB

HTML

CSS

BASIC TECHNOLOGY STACK



TOOLS WE'LL BE USING

- **Text editor**: Atom (open-source software)
- **Browser**: Chrome (excellent Dev Tools)

Note that these tools are just that, tools. You can choose to use other text editors and browsers as well. These were chosen because of some conveniences they afford.

GENERAL ASSEMBLY

INTERNET BASICS

ADVANCED RESEARCH PROJECTS AGENCY (ARPA)

- Set up in 1958 for R&D to expand the frontiers of technology and science
- Computers used to be monoliths which couldn't communicate with each other
- Best and brightest minds in the country came up with the concept of computer networking

PACKET SWITCHING TECHNOLOGY

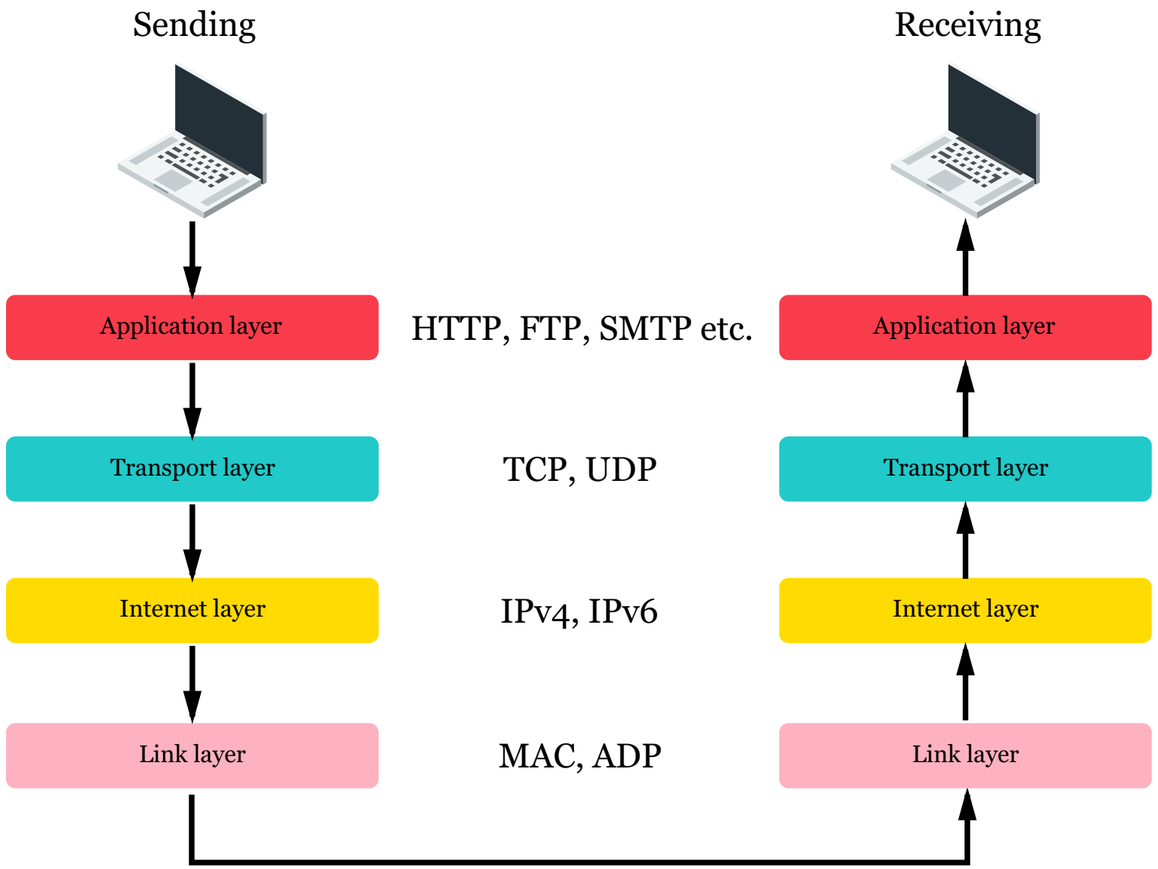
- Traditional communication networks used analog, circuit switching
- Circuit switching is like an MRT train running on tracks, while packet switching is like cars on the Expressways

PROTOCOLS

- Transmission Control Protocol (TCP) handles breaking up data into *packets* to be sent and reassembling them at their destination
- Internet Protocol (IP) handles the formatting and addressing of the data packets
- Every device connected to the internet needs a unique IP address

TCP/IP

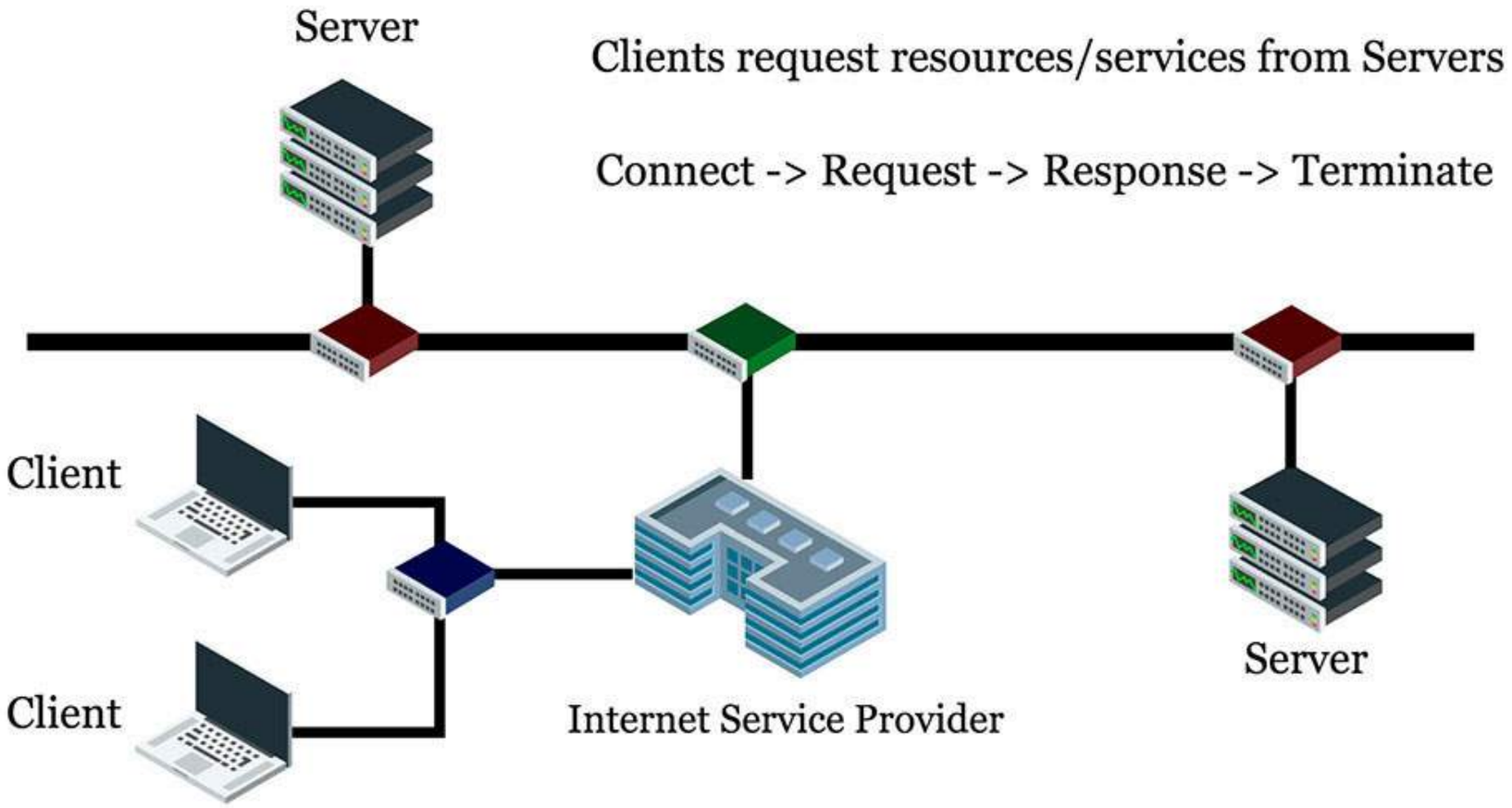
Transmission of data makes use of 4 layers



WORLD WIDE WEB

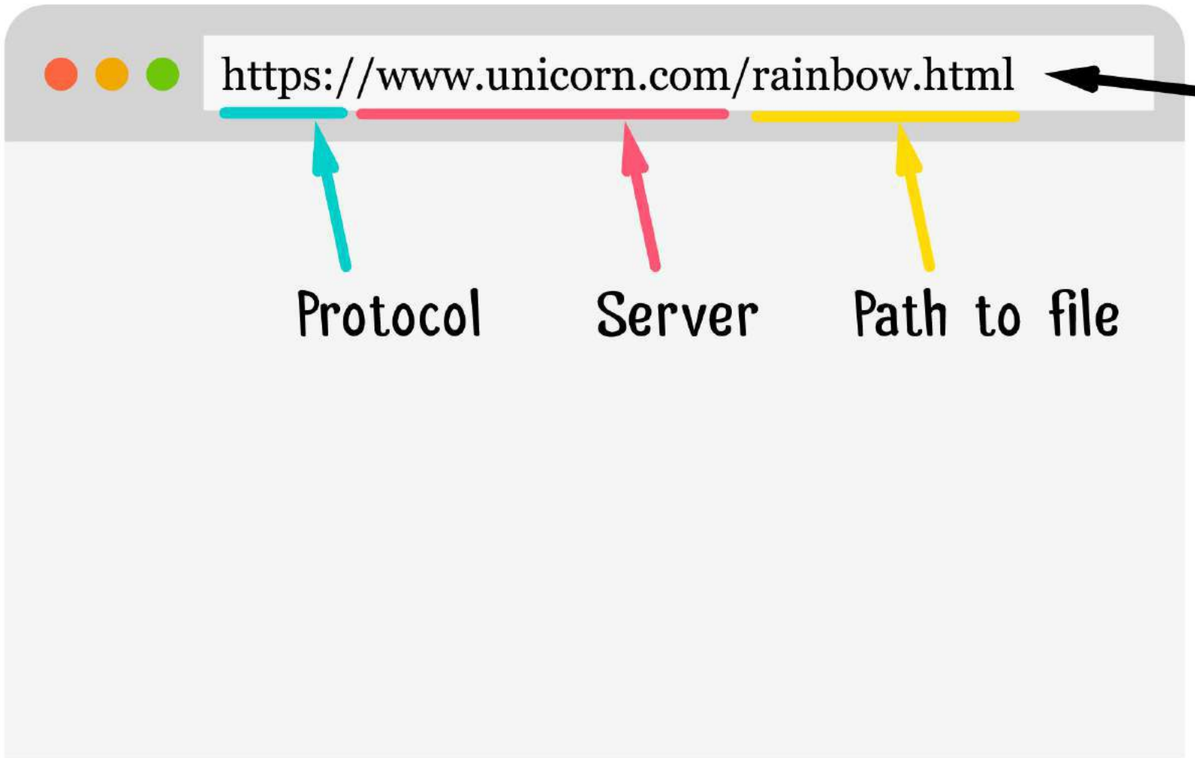
- Invented by **Tim Berners-Lee** in 1989
- Created the 3 essential technologies that power the World Wide Web:
  1. **Hypertext Transfer Protocol (HTTP)** for retrieving text from other documents via hypertext links
  2. **Uniform Resource Identifier (URI)** which is the unique identifier for every resource on the web
  3. **Hypertext Markup Language (HTML)** for structuring and presenting content on the web

CLIENTS AND SERVERS



FROM SERVER TO YOUR BROWSER

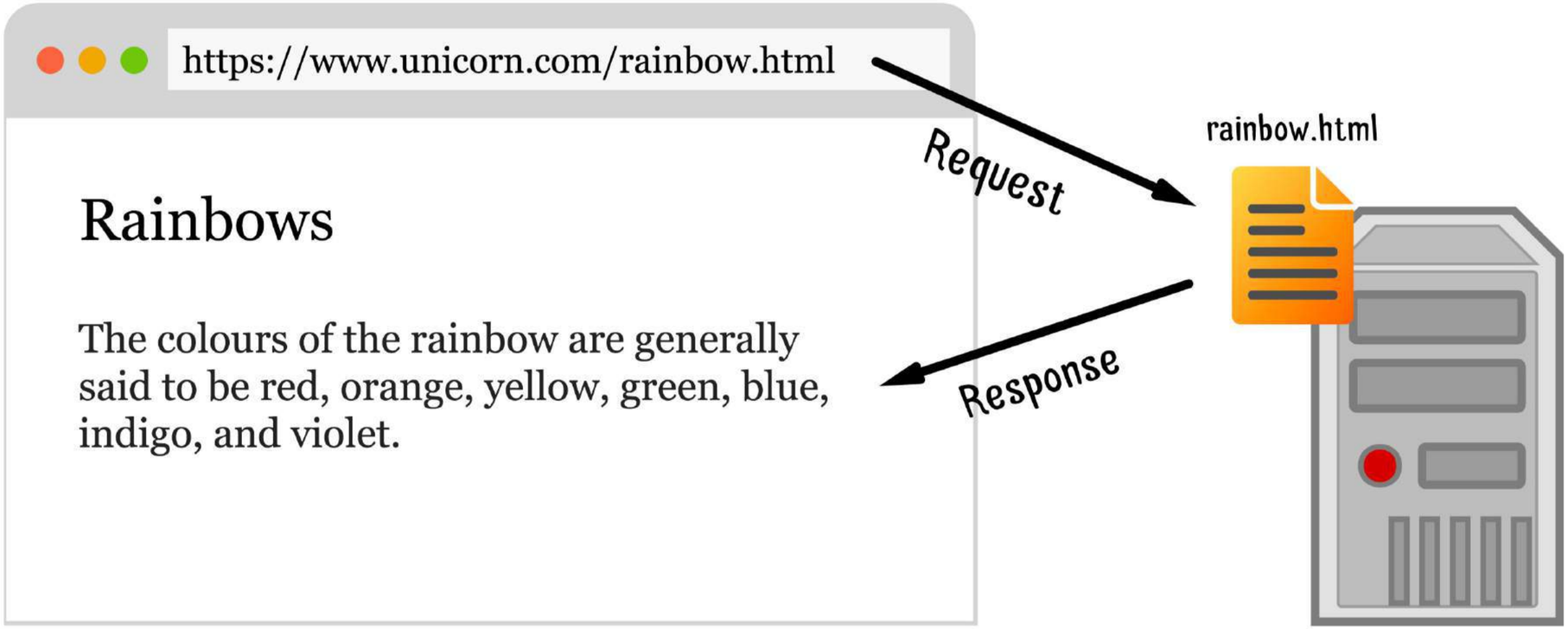
Enter a URL in the address bar



URL consists of the:
protocol,
server,
requested file

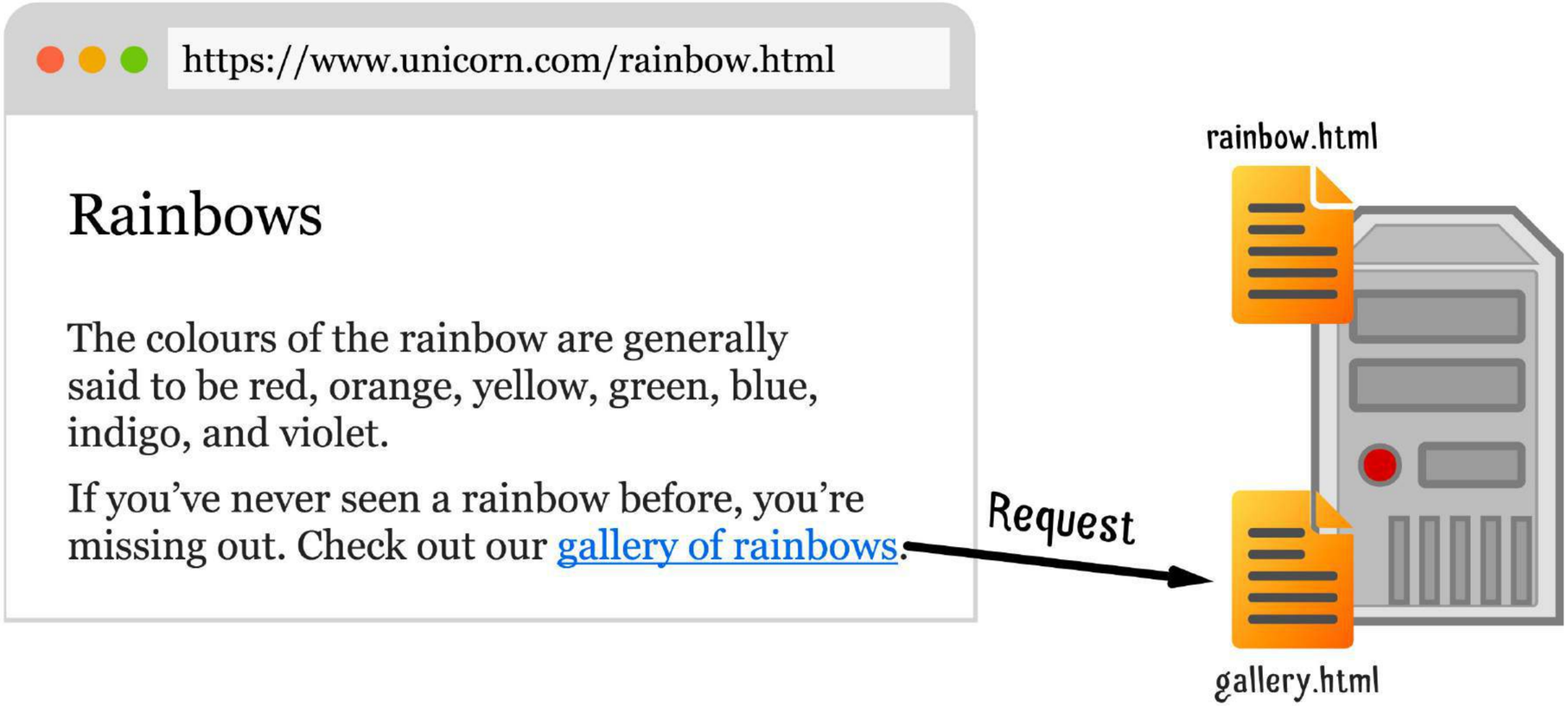
Browser sends request to server and server locates the requested file

Server returns the file to the browser which displays it

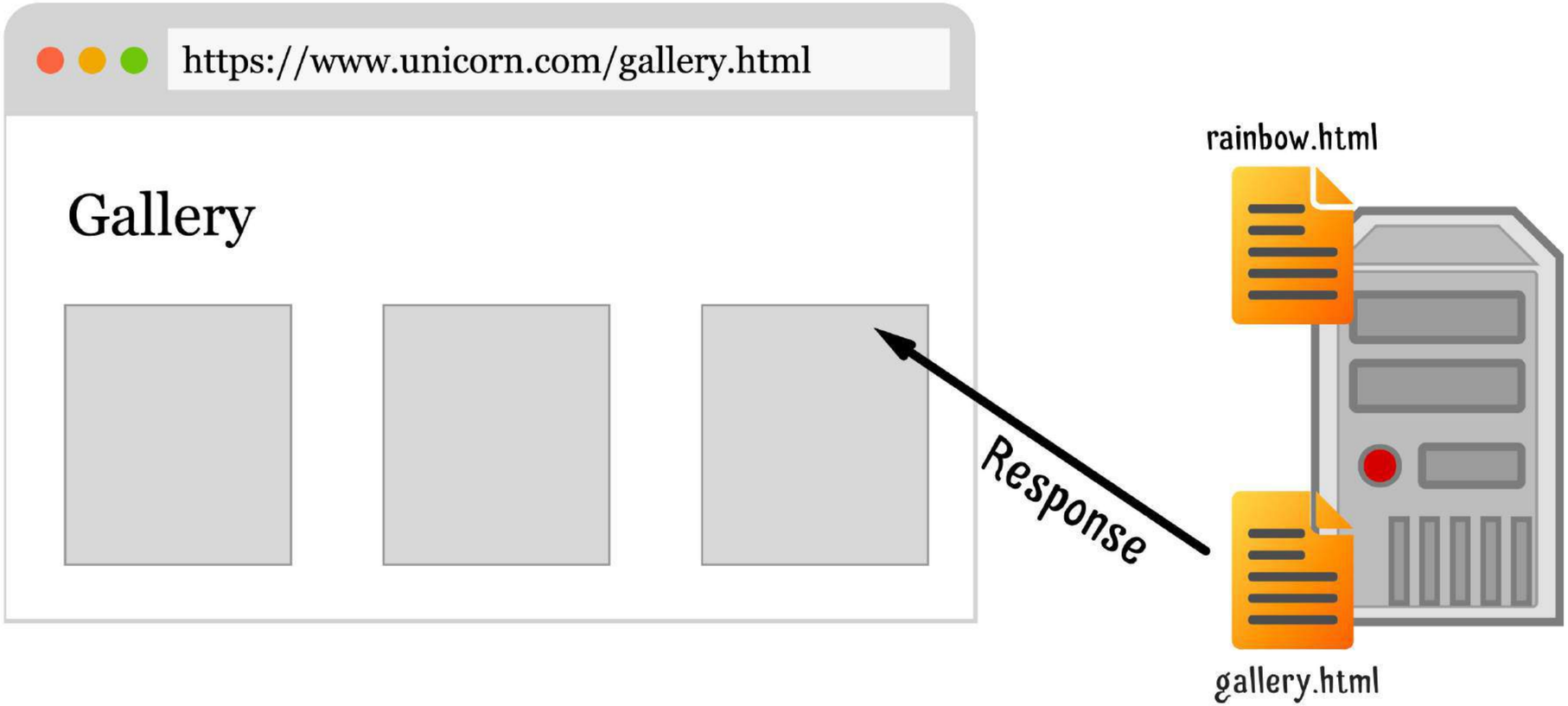


EVERYTHING IS CONNECTED BY LINKS

A link on a web page is a pre-entered URL. Clicking the link sends a request to the server.



The server sends the requested file back to the browser, which replaces the current page with the new file.



ABSOLUTE VS. RELATIVE LINKS

Absolute paths ask for a file from a **specific** location, which includes the protocol and server.

```
<a href="http://www.unicorn.com/gallery.html">Gallery<a>
```

Relative paths ask for a file without specifying a server.

```
<a href="gallery.html">Gallery<a>
```

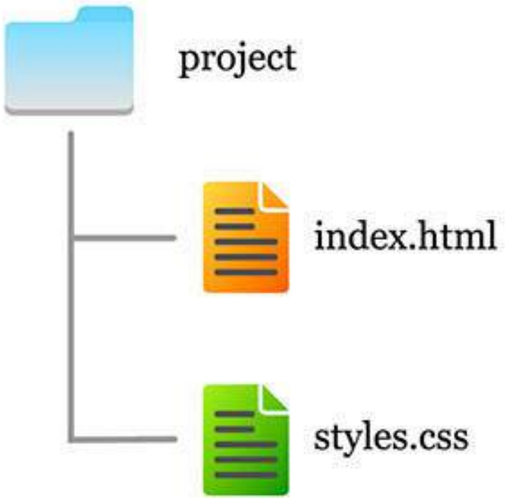
The browser will hence assume you're referring to the same server as the page you're on.

GENERAL ASSEMBLY

CREATE YOUR FIRST PROJECT

BASIC FOLDER STRUCTURE

1. Create a new folder, name it whatever you want
2. Create a file called index.html and another file called styles.css



GENERAL ASSEMBLY

HYPERTEXT MARK-UP LANGUAGE (HTML)

HYPERTEXT MARK-UP LANGUAGE (HTML)

- **Structures** the document and tells browsers what a certain element's function is
- Content is "marked-up" using tags
- Tags usually (but not always) come in pairs,

```
<p>This is an example of a paragraph element</p>
```

- The opening tag, closing tag and everything in between is a **HTML element**

STRUCTURE OF HTML DOCUMENT

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example page</title>
  </head>
  <body>
    <h1>Hello world</h1>
  </body>
</html>
```

DOCUMENT TYPE ELEMENT

```
<!DOCTYPE html>
```

- Appears just above the `<html>` tag
- Tells the browser to render the HTML in **standards** mode
- Let's validation software know which version of HTML to validate against
- Advised to use the **HTML5 doctype**

<html> ELEMENT

```
<html lang="en">
// HTML code for web page
</html>
```

- Represents the root of an HTML document
- Encouraged to specify a language attribute
- Language attribute aids speech synthesis (screen readers), translation tools and other language-related functionality

<head> ELEMENT

```
<head>
  <meta charset="utf-8">
  <title>Your site title</title>
  <meta name="description" content="A short description of your webs
  <meta name="author" content="Your name">

  <link rel="stylesheet" href="css/styles.css?v=1.0">
</head>
```

- Contains instructions for the browser and meta data for the website
- Title and description are what shows up on search engine results
- Stylesheets are also declared here

<body> ELEMENT

```
<body>
  <header>
    <img src="img/logo.png" alt="Site logo">
    <nav>
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <h1>Page header</h1>
    <p>Some content in a paragraph. Brownie tiramisu toffee sweet r
  <main>
</body>
```

- Represents the **main content** of the document
- Should only be one <body> element on a web page

FORMATTING YOUR WEB PAGE

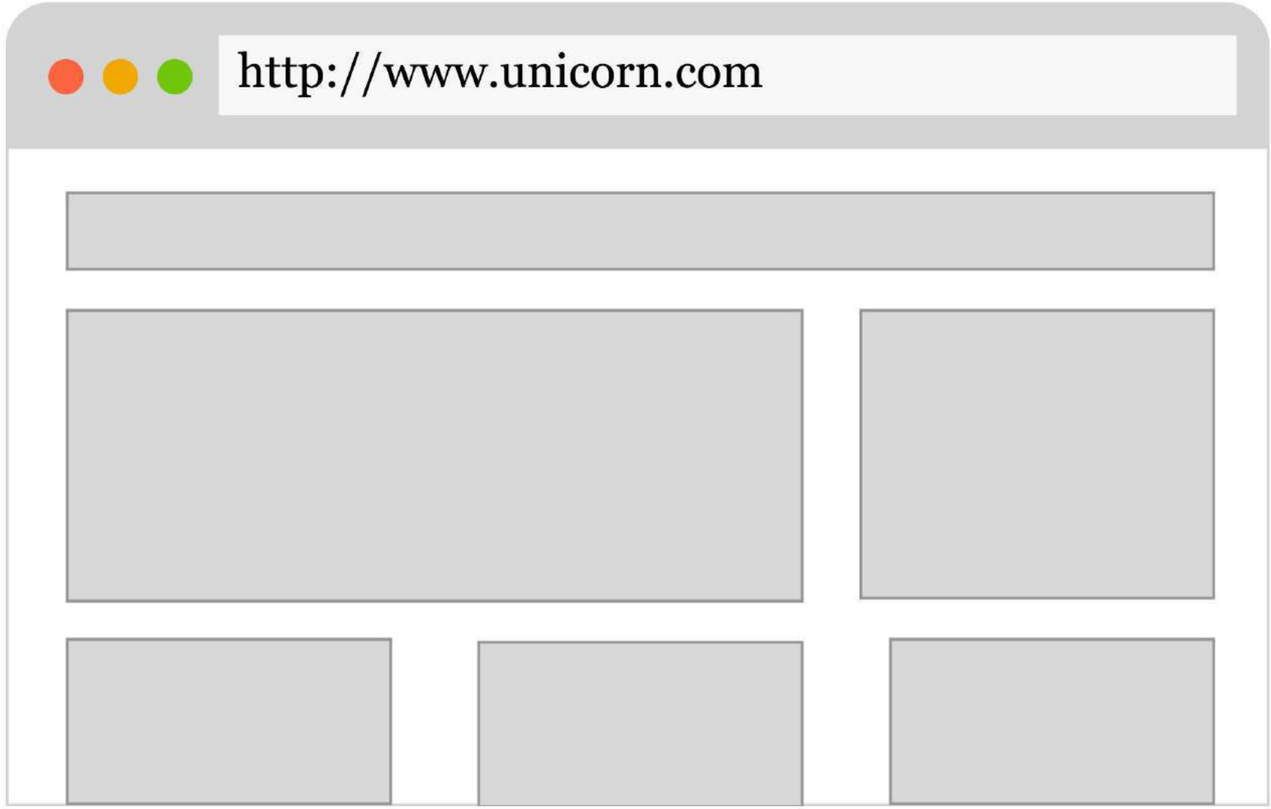
- <address>
- <article>
- <footer>
- <header>
- <h1>
- <h2>
- <h3>
- <h4>
- <h5>
- <h6>
- <hgroup>
- <nav>
- <section>
- <dd>
-
- <dl>
- <dt>
- <figcaption>
- <figure>
- <hr>
-
- <main>
-
- <p>
- <pre>
-
- <caption>
- <col>
- <colgroup>
- <table>
- <tbody>
- <td>
- <tfoot>
- <th>
- <thead>
- <tr>
- <button>
- <datalist>
- <fieldset>
- <form>
- <input>
- <keygen>
- <label>
- <legend>
- <meter>
- <optgroup>
- <option>
- <output>
- <progress>
- <select>
- <details>
- <dialog>
- <menu>
- <menuitem>
- <summary>
- <abbr>
- <b>
- <bdi>
- <bdo>
-
- <cite>
- <code>
- <data>
- <dfn>
-
- <i>
- <kbd>
- <mark>
- <q>
- <rp>
- <rt>
- <rtc>
- <ruby>
- <s>
- <samp>
- <small>
-
-
- <sub>
- <sup>
- <time>
- <u>
- <var>
- <wbr>
- <area>
- <audio>
- <map>
- <track>
- <video>
- <embed>
- <object>
- <param>
- <source>
- <canvas>
- <noscript>
- <script>
- <del>
- <ins>

BASIC HTML5 TEMPLATE

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>title</title>
    <meta name="description" content="A short description of your web
    <meta name="author" content="Your name">
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <!-- page content -->
    <script src="script.js"></script>
  </body>
</html>
```

TOP TO BOTTOM, LEFT TO RIGHT

- Web pages are made up of rectangular boxes
- These boxes are placed from top to bottom, left to right



BLOCK-LEVEL ELEMENTS

Block-level elements take up the entire width of the container.

RAINBOWS

A rainbow is a meteorological phenomenon that is caused by reflection, refraction and dispersion of light in water droplets resulting in a spectrum of light appearing in the sky.

COLOURS

- Red
- Orange
- Yellow
- Green
- Blue
- Indigo
- Violet

The block-level tags shown in this example are `h1`, `h2`, `p`, `ul` and `li`.

You can refer to the full list of block-level elements here.

INLINE-LEVEL ELEMENTS

If an element is *NOT* block-level, it is inline.

Accordingly, the Munsell colour system (a 20th-century system for numerically describing colours, based on equal steps for human visual perception) distinguishes 100 hues.

```
<p class="ga-example-4">Accordingly, <a href="https://en.wikipedia.o
```

Commonly used inline-level tags include `a`, `input`, `label`, `img` and so on.

Full list of inline-level elements available here.

GENERAL ASSEMBLY

CASCADING STYLE SHEETS (CSS)

CASCADING STYLE SHEETS (CSS)

- Tells the browser how to **display** a certain element
- Follows the general ruleset:
  1. Select the HTML element to be styled
  2. Specify the properties of the element to be styled
  3. Give the values we want each property to have

STRUCTURE OF A CSS RULE

```
selector {
  property1: value;
  property2: value;
  property3: value;
}
```

- The **selector** identifies which HTML elements the rule will be applied to
- The **curly braces** contain the property-value pairs, separated with semi-colons
- The **properties** define the style of the selected element
- The **values** are dependent on the property, and indicate the value of the properties to be set

TYPES OF CSS SELECTORS

- **Element**: matches all the elements of that name on the page

```
p {}
```

- **Class**: matches all the elements with the specified class attribute, e.g. `<div class="example">`

```
.example {}
```

- **ID**: matches the element with the specified id attribute, e.g. `<div id="example">`

```
#example {}
```

DESCENDENT SELECTORS

Used to select tags that are children of other tags

```
<ul>
  <li>4 large eggs</li>
  <li>1/4 cup milk</li>
  <li>2 tsp. butter</li>
</ul>
<ol>
  <li>BEAT eggs, milk, salt and p
  <li>HEAT butter in large nonsti
  <li>CONTINUE cooking – pulling,
</ol>
```

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

- 4 large eggs
- 1/4 cup milk
- 2 tsp. butter

1. BEAT eggs, milk, salt and pepper in medium bowl until blended.
2. HEAT butter in large nonstick skillet over medium heat until hot. POUR IN egg mixture. As eggs begin to set, GENTLY PULL the eggs across the pan with a spatula, forming large soft curds.
3. CONTINUE cooking – pulling, lifting and folding eggs – until thickened and no visible liquid egg remains. Do not stir constantly. REMOVE from heat. SERVE immediately.

Selector list is read from right-to-left, with the left-most being the parent.

PSEUDO-SELECTORS

Applies to selectors when certain conditions occur

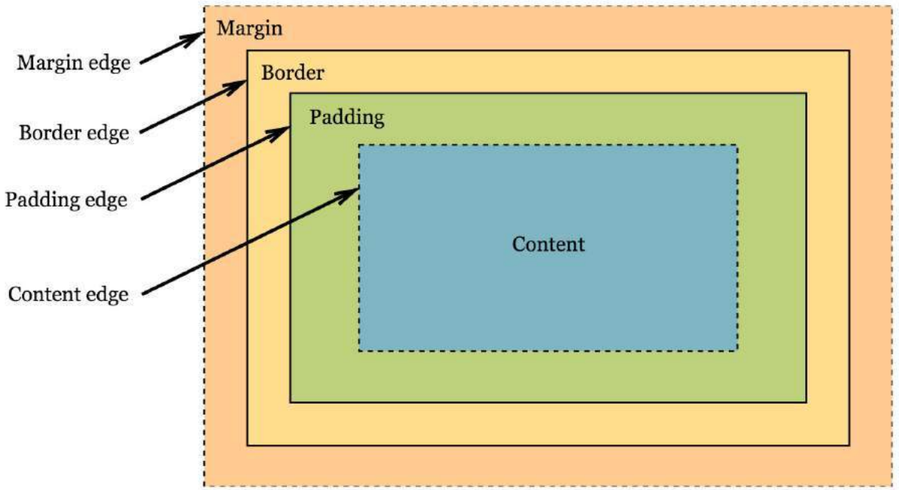
```
a {
  /* removes underlines from
     all text links */
  text-decoration: none;
}
a:hover {
 /* adds an underline and makes
    the font green when hovered */
  text-decoration: underline;
  color: green;
}
```

- Link 1
- Link 2
- Link 3

There are many other pseudo-selectors you can use as well. The full list is available here.

THE BOX MODEL

The model is made up of four boxes, from inside to outside:



The box model, visualised

- Content
- Padding
- Border
- Margin

WHEN TO USE MARGIN

Margin controls the space between elements.

```
h2 {
  margin: 5px 0 5px 0;
}
```

RAINBOWS

A rainbow is a meteorological phenomenon that is caused by reflection, refraction and dispersion of light.

COLOURS

- Red
- Orange
- Yellow

```
h2 {
  margin: 20px 0 20px 0;
}
```

RAINBOWS

A rainbow is a meteorological phenomenon that is caused by reflection, refraction and dispersion of light.

COLOURS

- Red
- Orange
- Yellow

WHEN TO USE PADDING

Padding controls the size of the box without adjusting the size of the content within it.

```
h2 {
  padding: 0;
}
```

RAINBOWS

A rainbow is a meteorological phenomenon that is caused by reflection, refraction and dispersion of light.

COLOURS

- Red
- Orange
- Yellow

```
h2 {
  padding: 20px 0 20px 0;
}
```

RAINBOWS

A rainbow is a meteorological phenomenon that is caused by reflection, refraction and dispersion of light.

COLOURS

- Red
- Orange
- Yellow

WHERE TO WRITE YOUR STYLES

Browsers will pick up your CSS if they are between a `<style>` tags which is a child of the `<head>` tag.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <style type="text/css">
      h1 {
        font-size: 2rem;
      }
      a {
        text-decoration: none;
      }
      a:hover {
        text-decoration: underline;
        color: darkred;
      }
    </style>
  </head>
```

USE A SEPARATE CSS FILE

As your site grows, you'll have many more styles, so it's better to move them all into a separate file.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link type="text/css" rel="stylesheet" href="main.css">
  </head>
```

In this example, we are using `main.css` but you can name the file anything. This file will hold all your CSS and be linked in the `<head>` of every page.

CSS SPECIFICITY

0-∞	0-∞	0-∞	0-∞
Inline styles	IDs	Classes, attributes and pseudo-classes	Elements and pseudo-elements

```
ul {
  // CSS properties
}
```

0, 0, 0, 1

```
.class-1 .class-2 p {
  // CSS properties
}
```

0, 0, 2, 1

```
#id-1 .class-3 div {
  // CSS properties
}
```

0, 1, 1, 1

GENERAL GUIDELINES FOR WRITING CSS

- Declare your styles from lowest specificity then move up
- Keep your specificity as low as possible
- Name your classes sensibly
- Never style IDs
- Don't write inline styles

GENERAL ASSEMBLY

USING IMAGES

TYPES OF IMAGES

- Content images
  - contain relevant information
  - help the user understand the content
- Background images
  - decorative in nature
  - contribute to the overall look and feel of the site

CONTENT IMAGES

Content images are created using the <img> tag

```
<img src="path/to/image" alt="Description of the image">
```

- Doesn't need a closing tag.
- Requires a <src> attribute to tell the browser where to find the image file
- Requires an <alt> attribute which describes the image or its purpose

BACKGROUND IMAGES

Background images are set via CSS

There are several properties related to backgrounds:

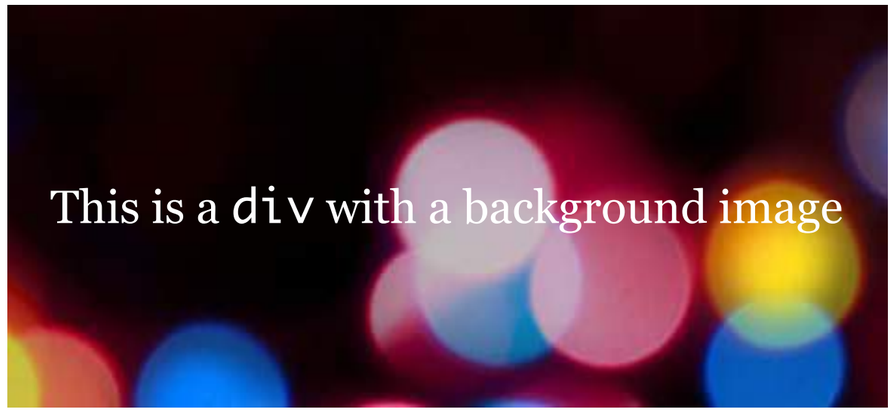
```
background-image: none
background-position: 0% 0%
background-size: auto auto
background-repeat: repeat
background-origin: padding-box
background-clip: border-box
background-attachment: scroll
background-color: transparent
```

`background` is one of many CSS properties that can be written in *shorthand*.

SETTING BACKGROUND IMAGES

`background-image` can use relative or absolute paths

```
div {
  background-color: #170104;
  background-image: url('img/back
}
```



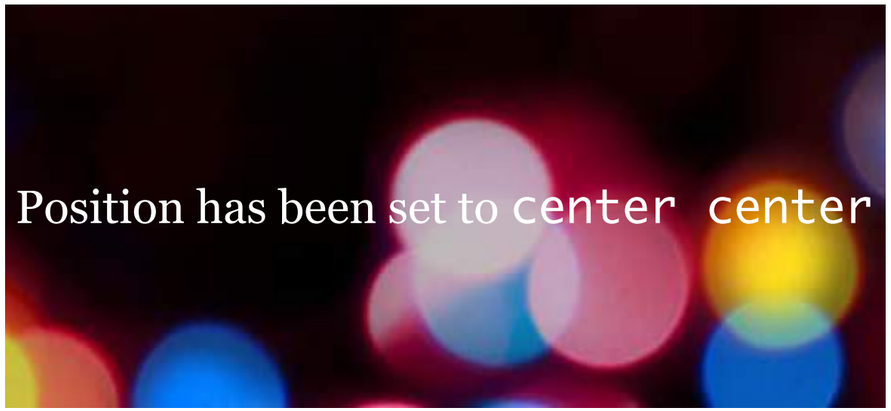
It's advisable to set a `background-color` as a fallback for the background image

BACKGROUND-POSITION

This is used to set the position of the image

```
div {
  background-color: #170104;
  background-image: url('img/backg
  background-position: center cent
}
```

```
div {
  background-color: #170104;
  background-image: url('img/backg
  background-position: left bottom
}
```



BACKGROUND-REPEAT

Used for tiling patterned backgrounds

Takes the following values:

- `repeat-x`: tiles the image horizontally
- `repeat-y`: tiles the image vertically
- `no-repeat`: don't tile or repeat anything

```
div {
  background-color: #EBEBEB;
  background-image: url('img/sativa.jpg');
  background-repeat: repeat;
}
```

GENERAL ASSEMBLY

WEB ACCESSIBILITY

SEMANTICS AND ACCESSIBILITY

- To make the web easier to use and access, and available to everyone
- Encompasses all disabilities, including visual, auditory, physical, speech, cognitive and neurological disabilities
- Benefits people *without* disabilities as well
- Accessible websites benefit from search engine optimisation (SEO)

BASIC ACCESSIBILITY CHECKLIST (1/2)

- **Page title**: To adequately and briefly describe the content of the page
- **Image text alternatives**: To make visual information accessible
- **Headings**: To provide meaningful hierarchy for facilitation of navigation
- **Contrast ratio**: To have sufficient luminance contrast ratio, for people with different requirements
- **Resize text**: To ensure visibility and usability as text size increases

BASIC ACCESSIBILITY CHECKLIST (2/2)

- **Keyboard access & visual focus**: To provide full functionality through a keyboard, and visible focus with logical order
- **Forms, labels & errors**: To have proper labels, keyboard access, clear instructions, and effective error handling
- **Multimedia alternatives**: To have alternative formats for audio and visual impaired

Visit Web Accessibility Initiative (WAI) to understand more about this important aspect of the web

GENERAL ASSEMBLY

RESOURCES

TO FIND OUT MORE...

- Dash (online course)
- Codeacademy (online course)
- Bento (online resources)
- Mozilla Developer Network (MDN) (website)
- HTML & CSS: Design and Build Web Sites by Jon Duckett (book)
- Designing with Web Standards by Jeffrey Zeldman (book)

SIGN UP FOR GA COURSES

- Web Development Immersive (full-time)
- Front-end development (part-time)
- Javascript development (part-time)
- Back-end development (part-time)
- Various classes and workshops

THE END

 http://www.chenhuijing.com

 @hj_chen

 @hj_chen

 @huijing