

TechLadies

in Penang

By Chen Hui Jing / @hj_chen

<http://bit.ly/tl-in-pen>



Surname

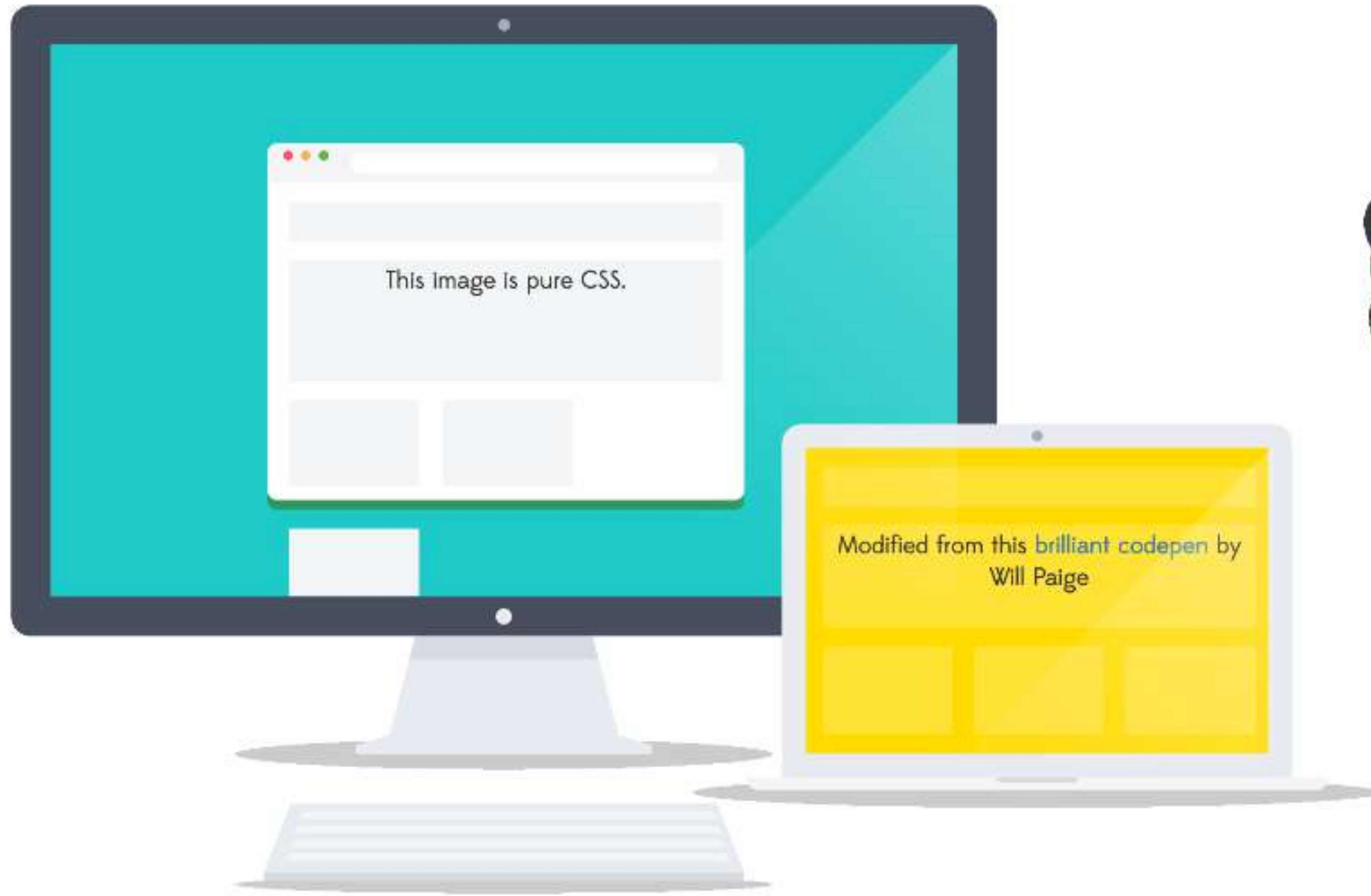
First name

陈 Chen	慧 晶 Hui Jing
-----------	-----------------

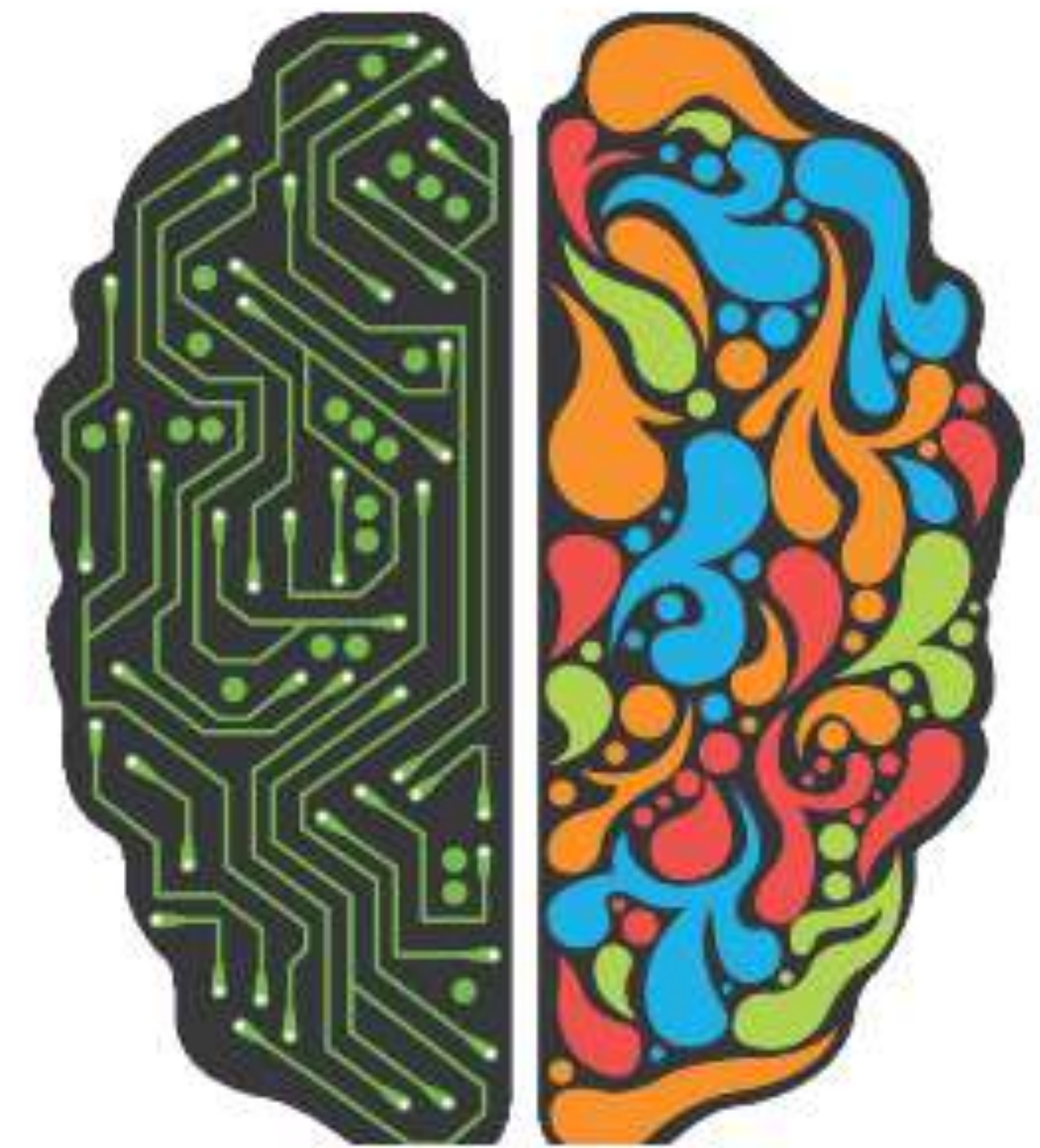


How to get started?

Your computer



A clear mind



HTML and CSS are the foundation of the web

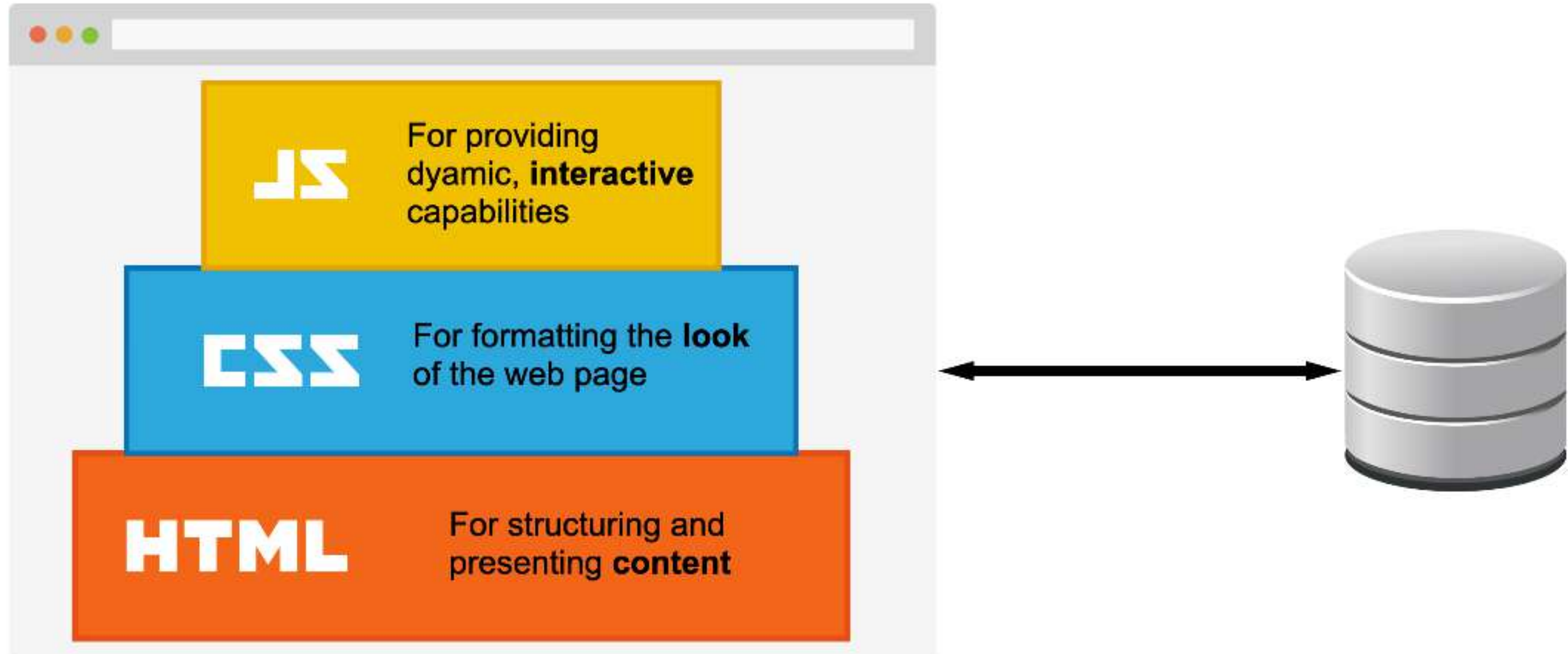
HTML



CSS



Basic technology stack



Tools we'll be using

- [Visual Studio Code](#) - Code editor
- [GitHub](#) - Git repository hosting service
- [Codeanywhere](#) - Online development environment
- [Heroku](#) - Application hosting service (may or may not cover this)

In most circumstances, we do development on our local machines, but given that setting up for development deserves its own workshop, we'll be using online tools for the second part of the workshop later.

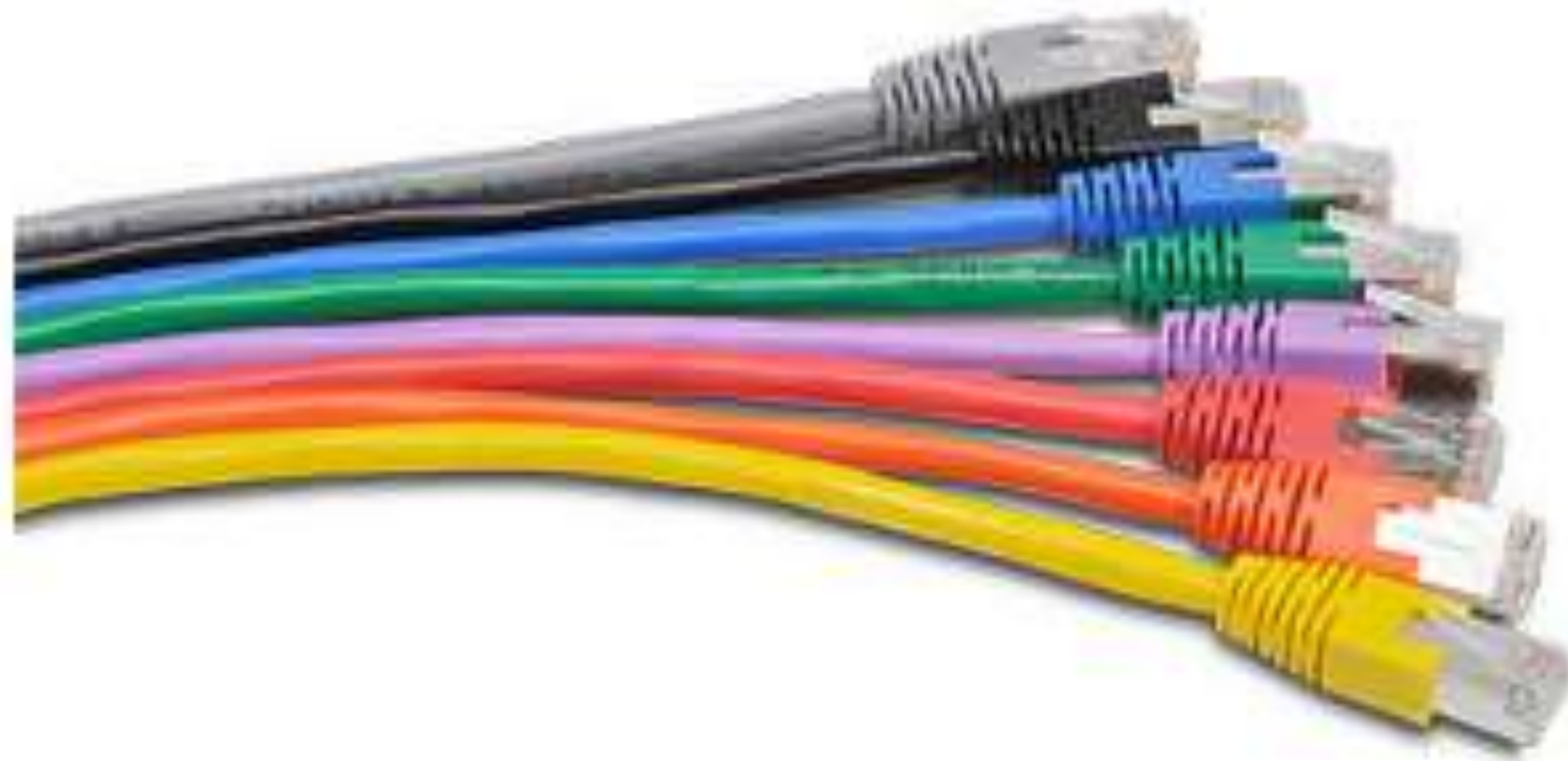


Internet basics

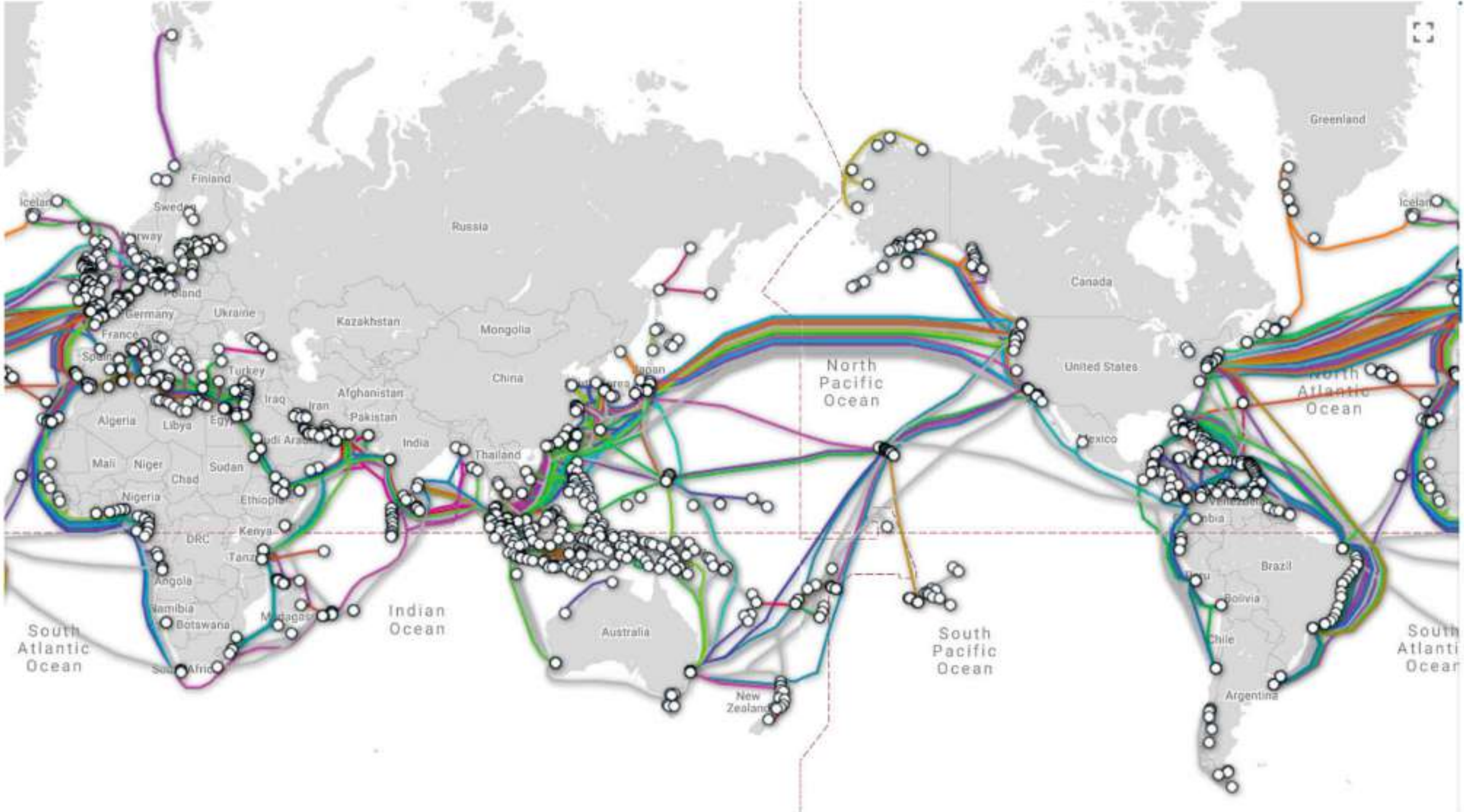


What is the internet?

The entire network of networks that connect all the world's devices to each other



Global Submarine Cable Map



Global Internet Exchange Map



From TeleGeography



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

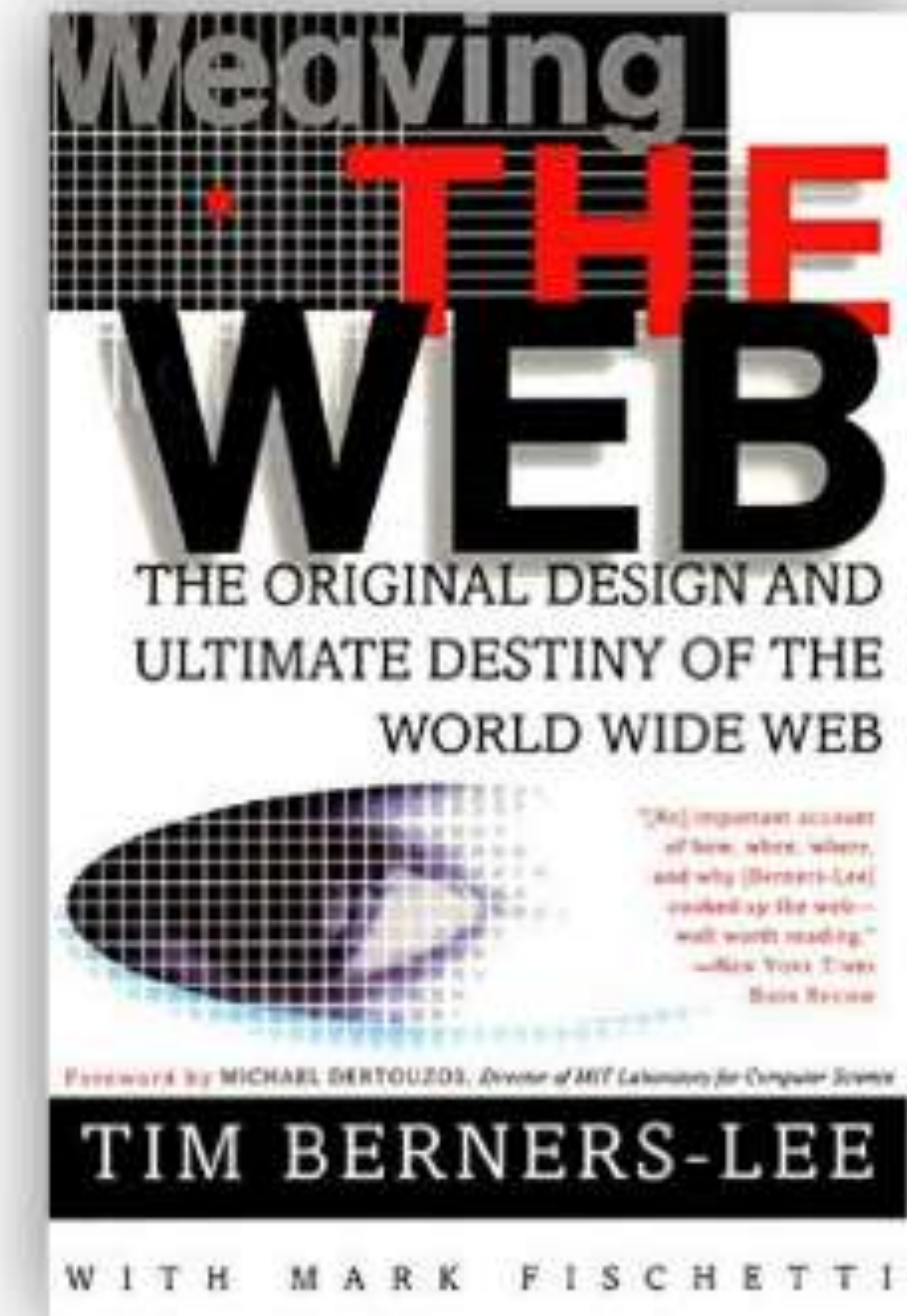
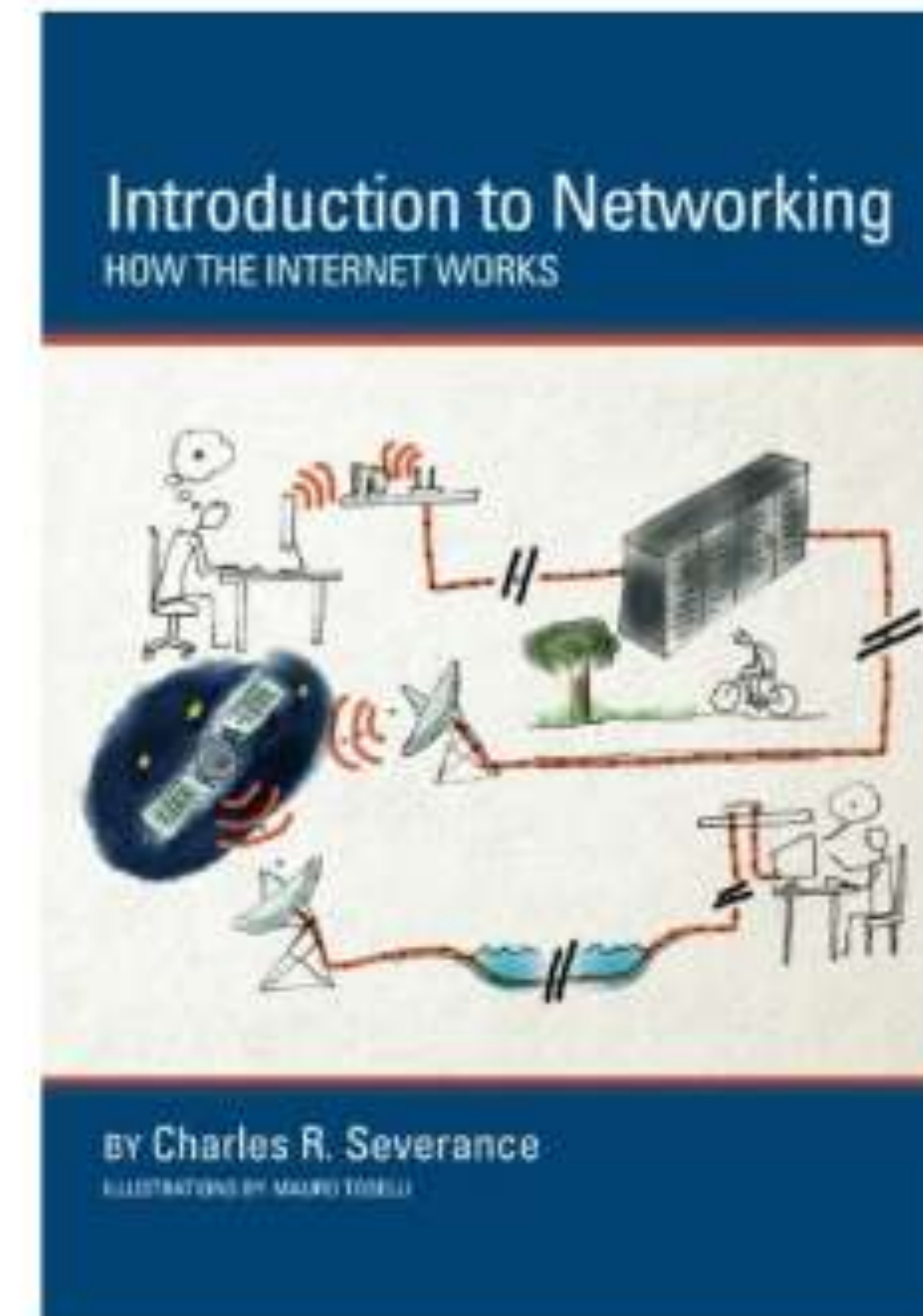
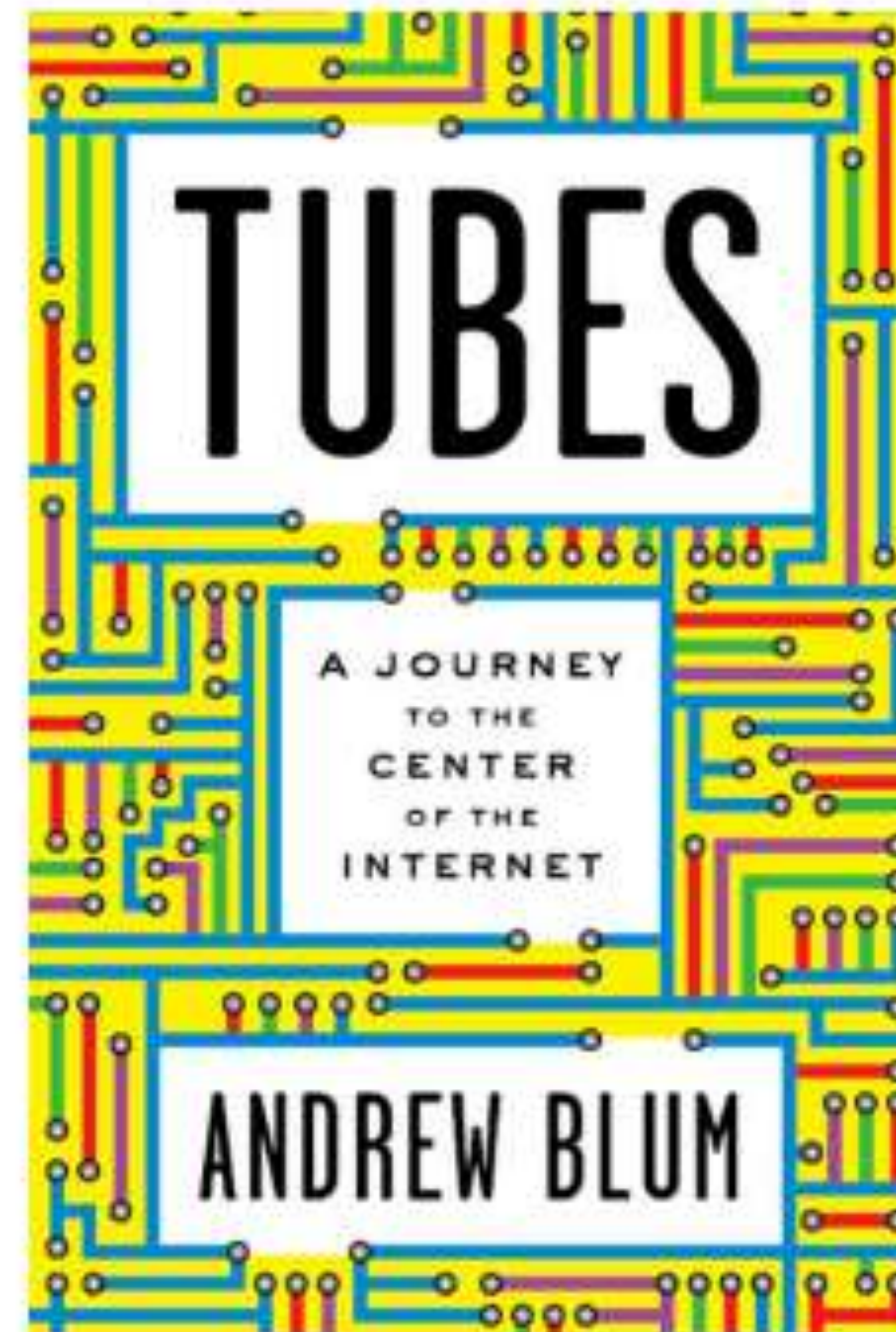
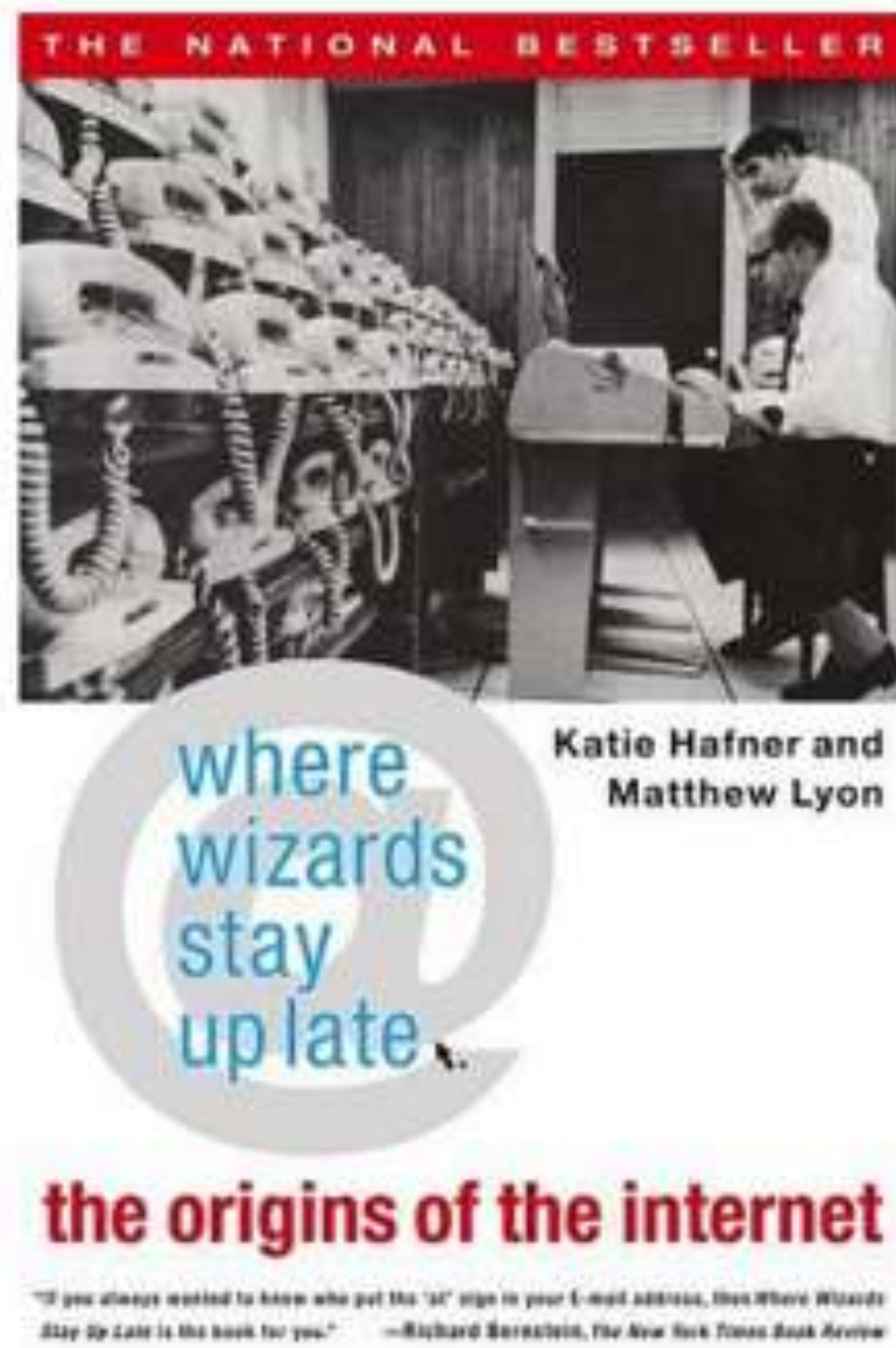


World Wide Web

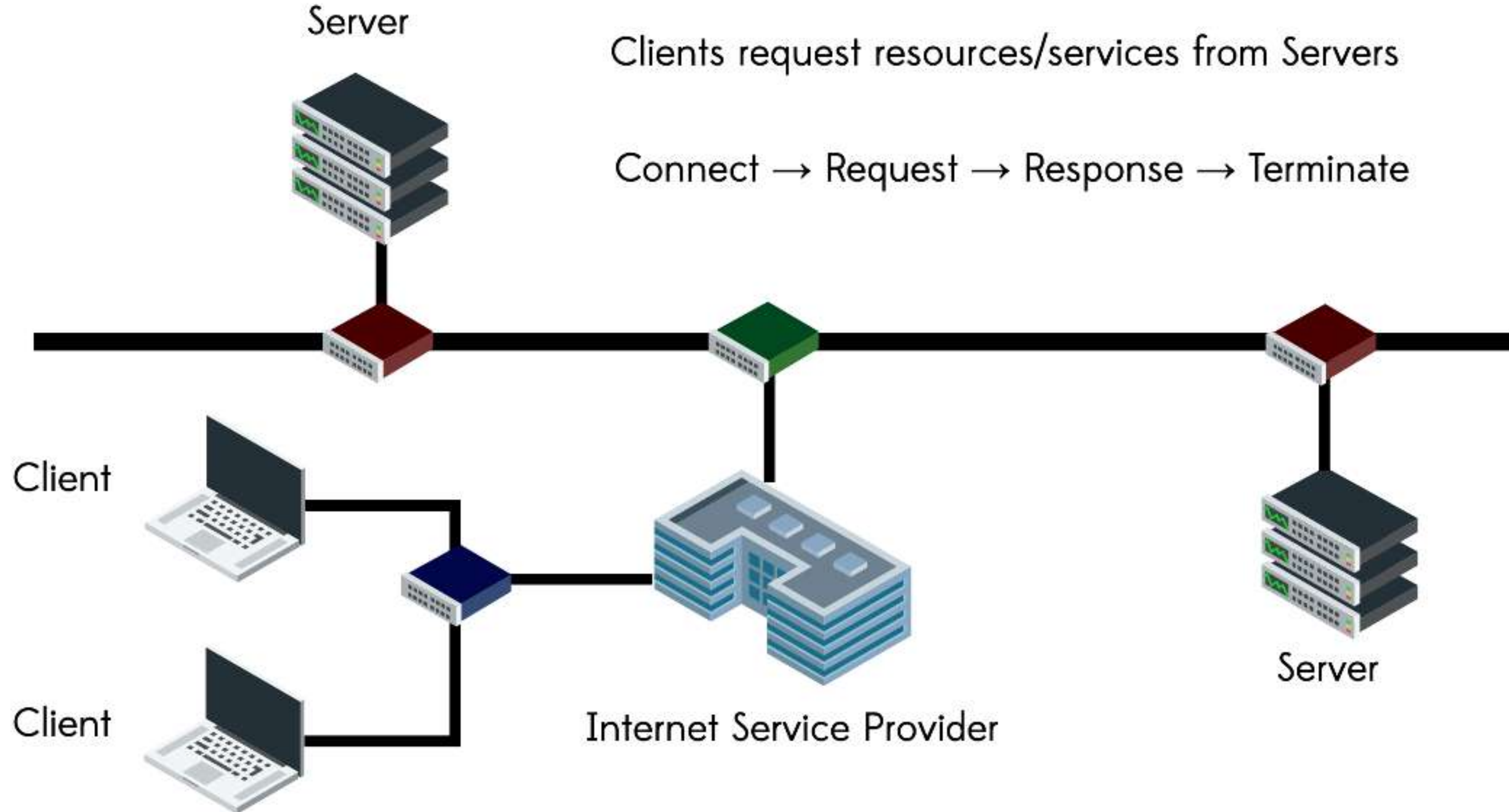
- Invented by **Tim Berners-Lee** in 1989
- Created the 3 essential technologies that power the World Wide Web:
 1. **Hypertext Transfer Protocol (HTTP)** for retrieving text from other documents via hypertext links
 2. **Uniform Resource Identifier (URI)** which is the unique identifier for every resource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



If you like reading books...

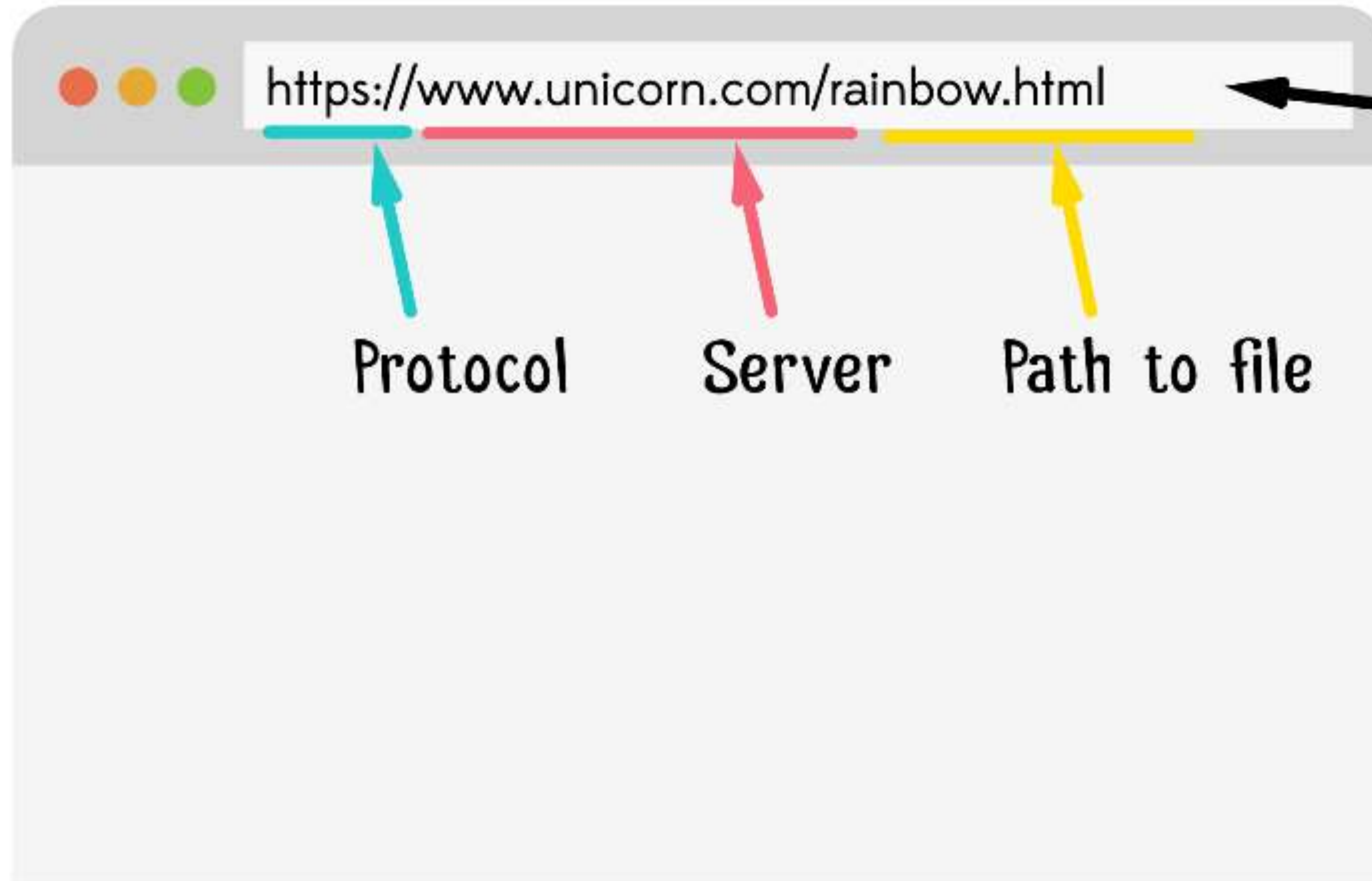


Clients and Servers



From server to your browser

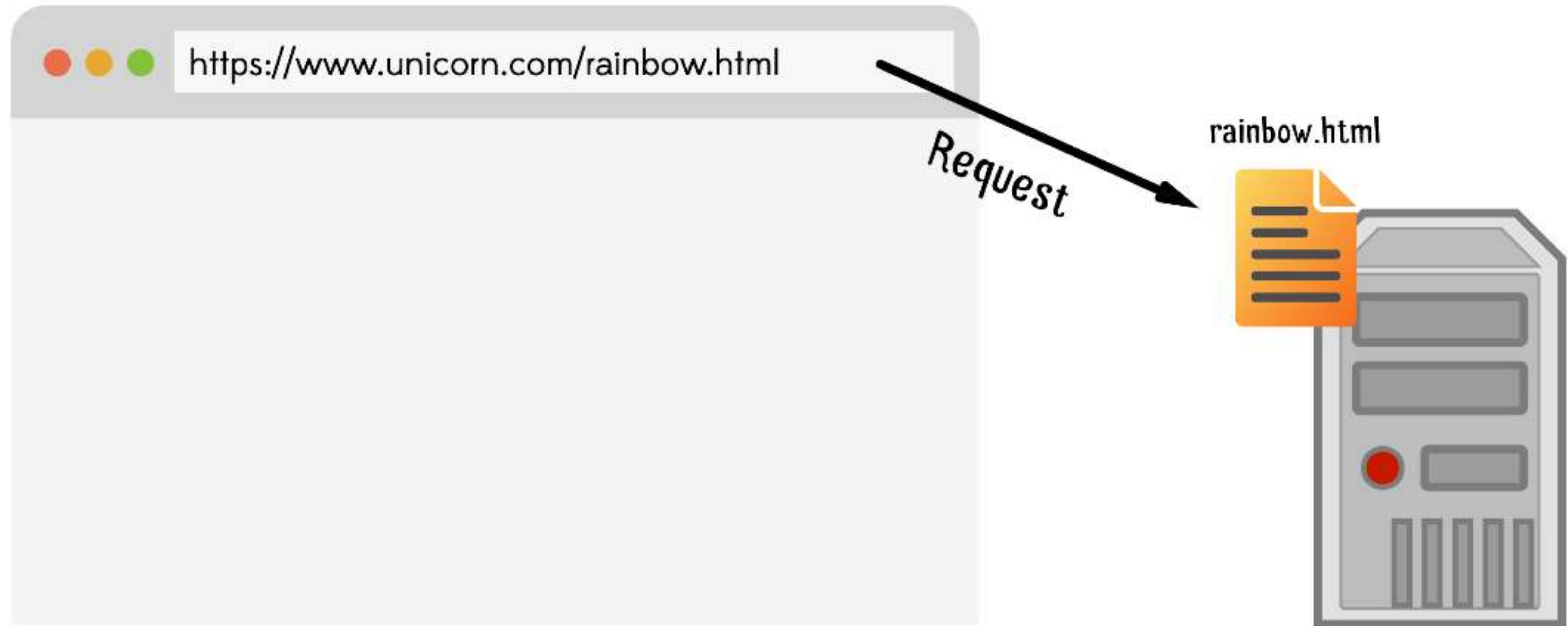
Enter a URL in the address bar



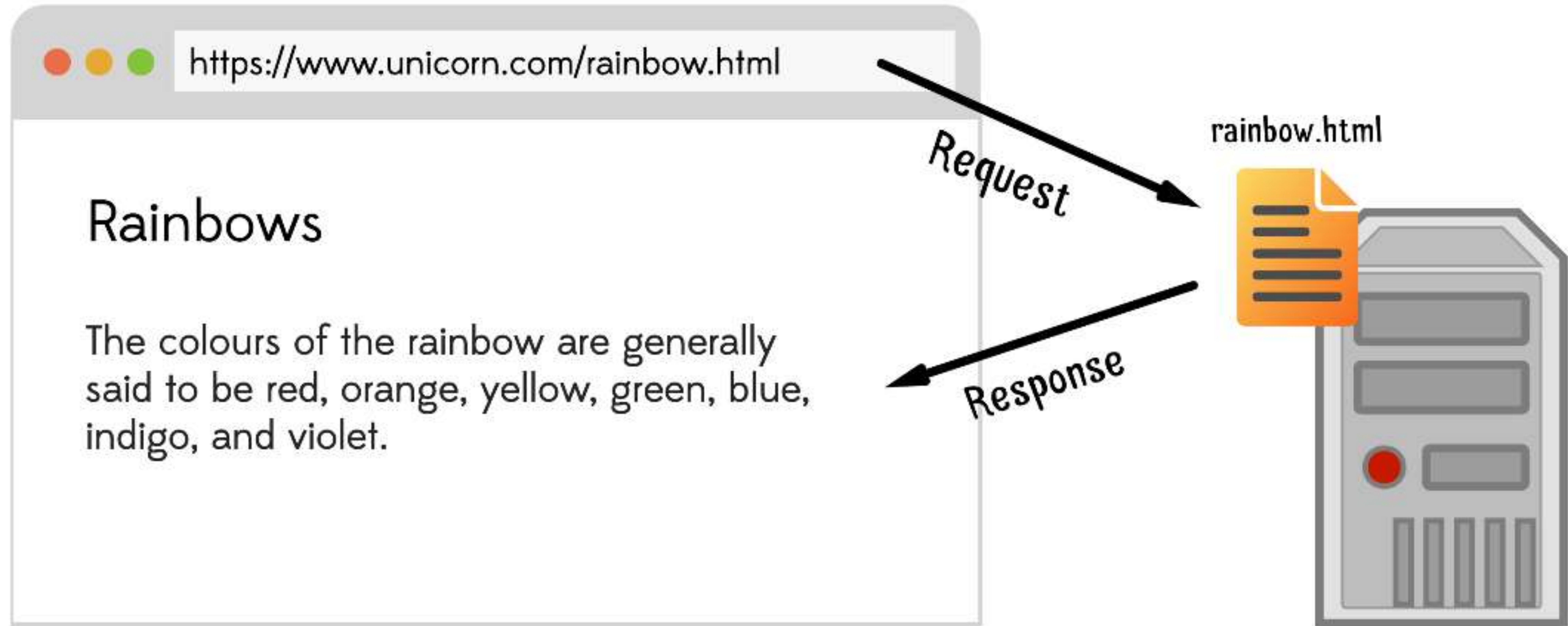
URL consists of the:
protocol,
server,
requested file



Browser sends request to server and server locates the requested file

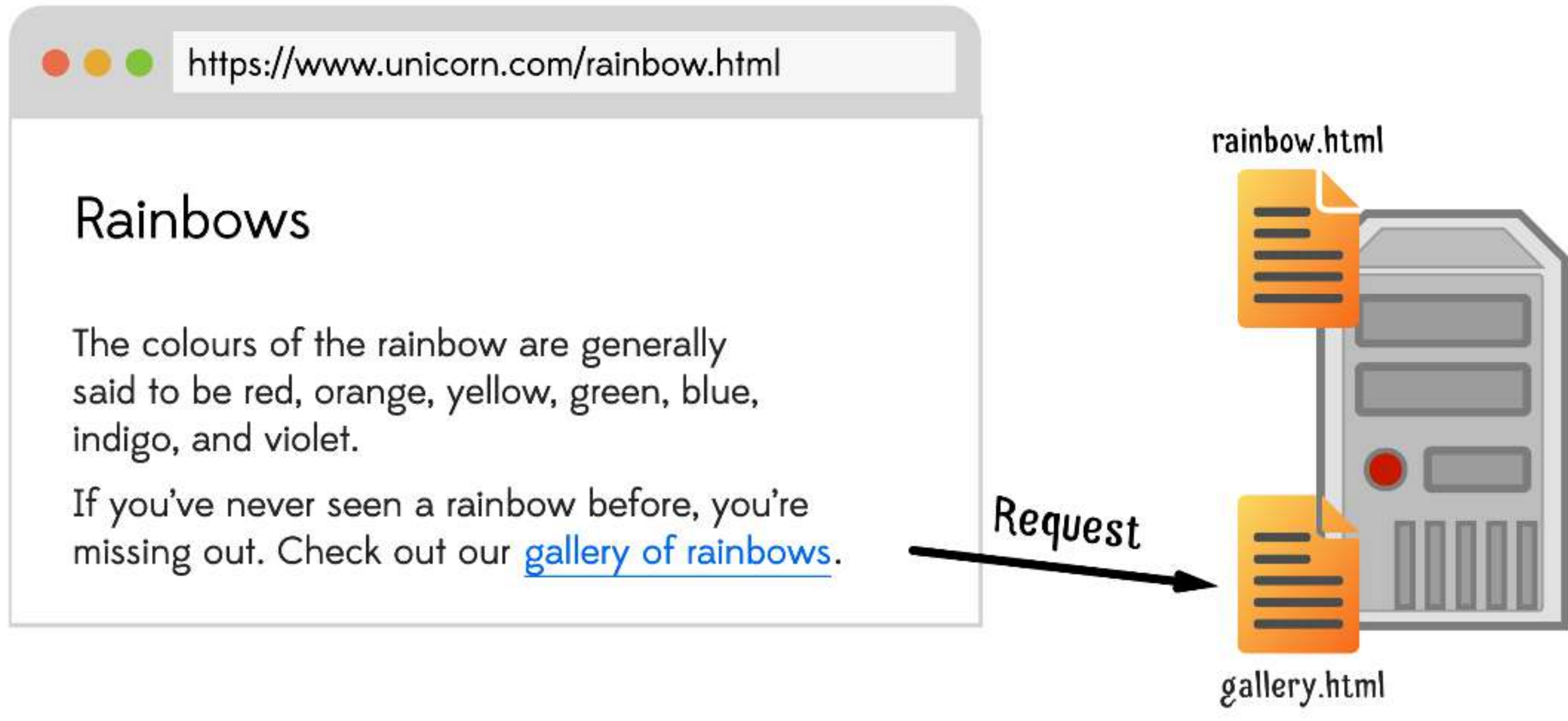


Server returns the file to the browser which displays it

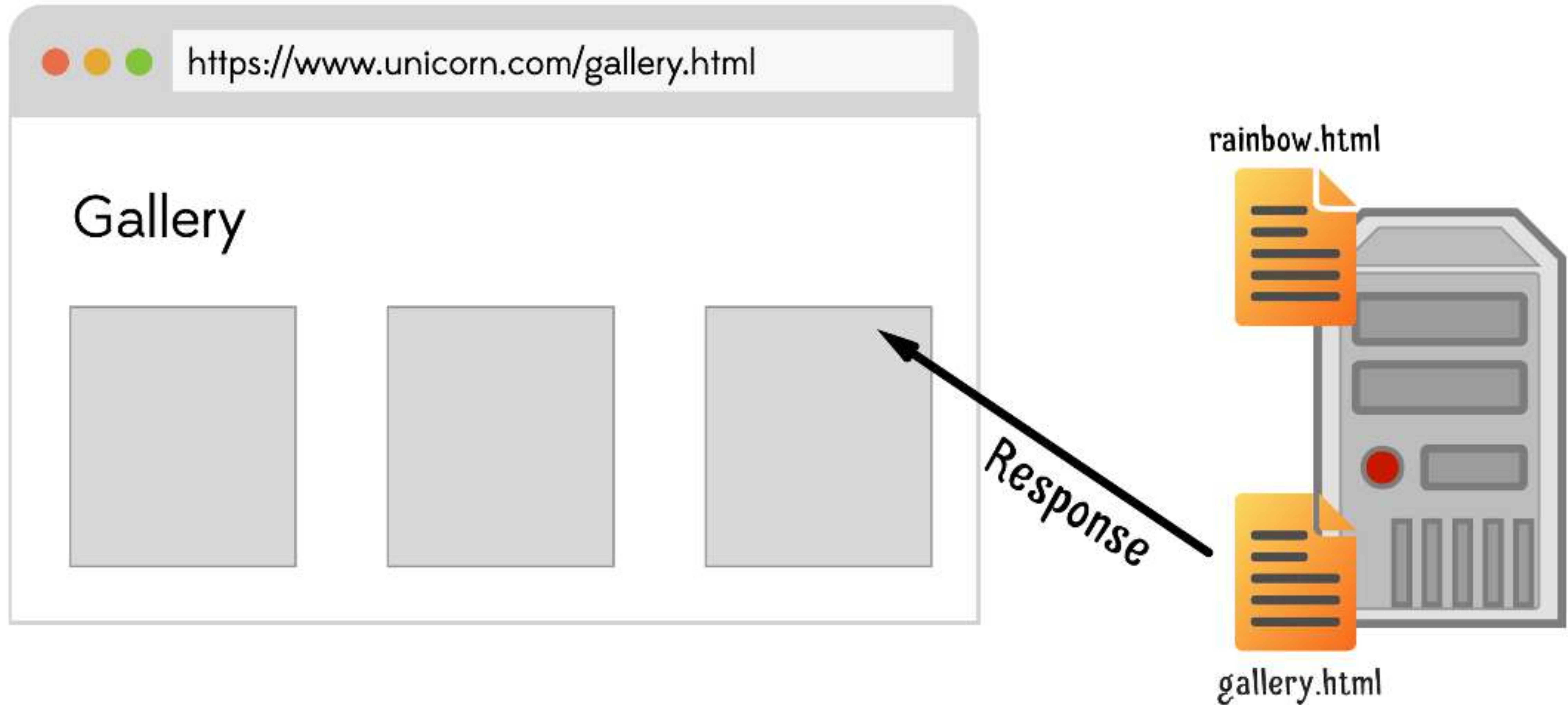


Everything is connected by links

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.



Hands-on Practice

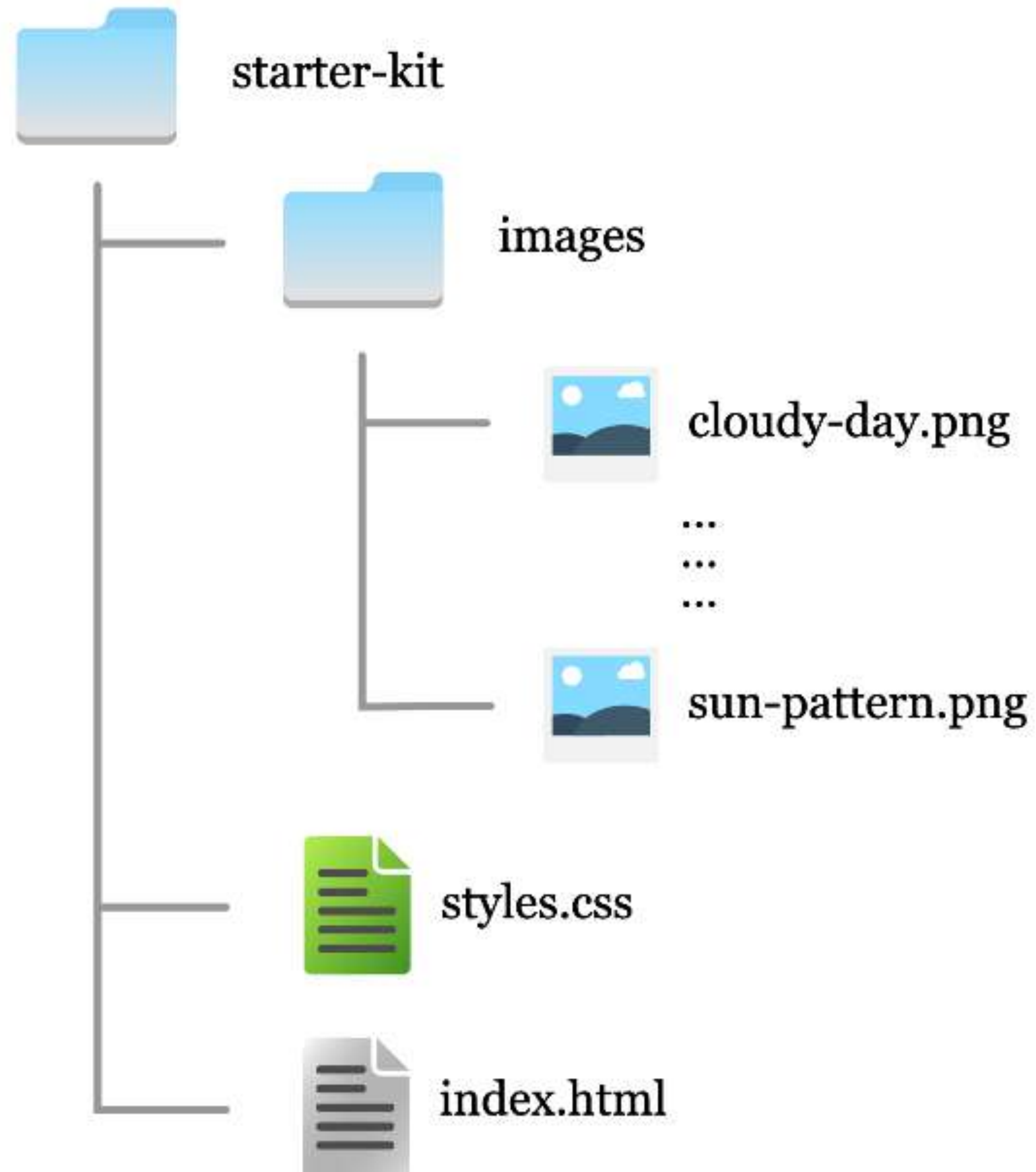


Download the starter-files

<http://bit.ly/tl-starter>



Starter kit files



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 website">
  <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>
    
    <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 roll se
  </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>`
- `<div>`
- `<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>`
- ``
- `<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>`
- ``
- `<ins>`



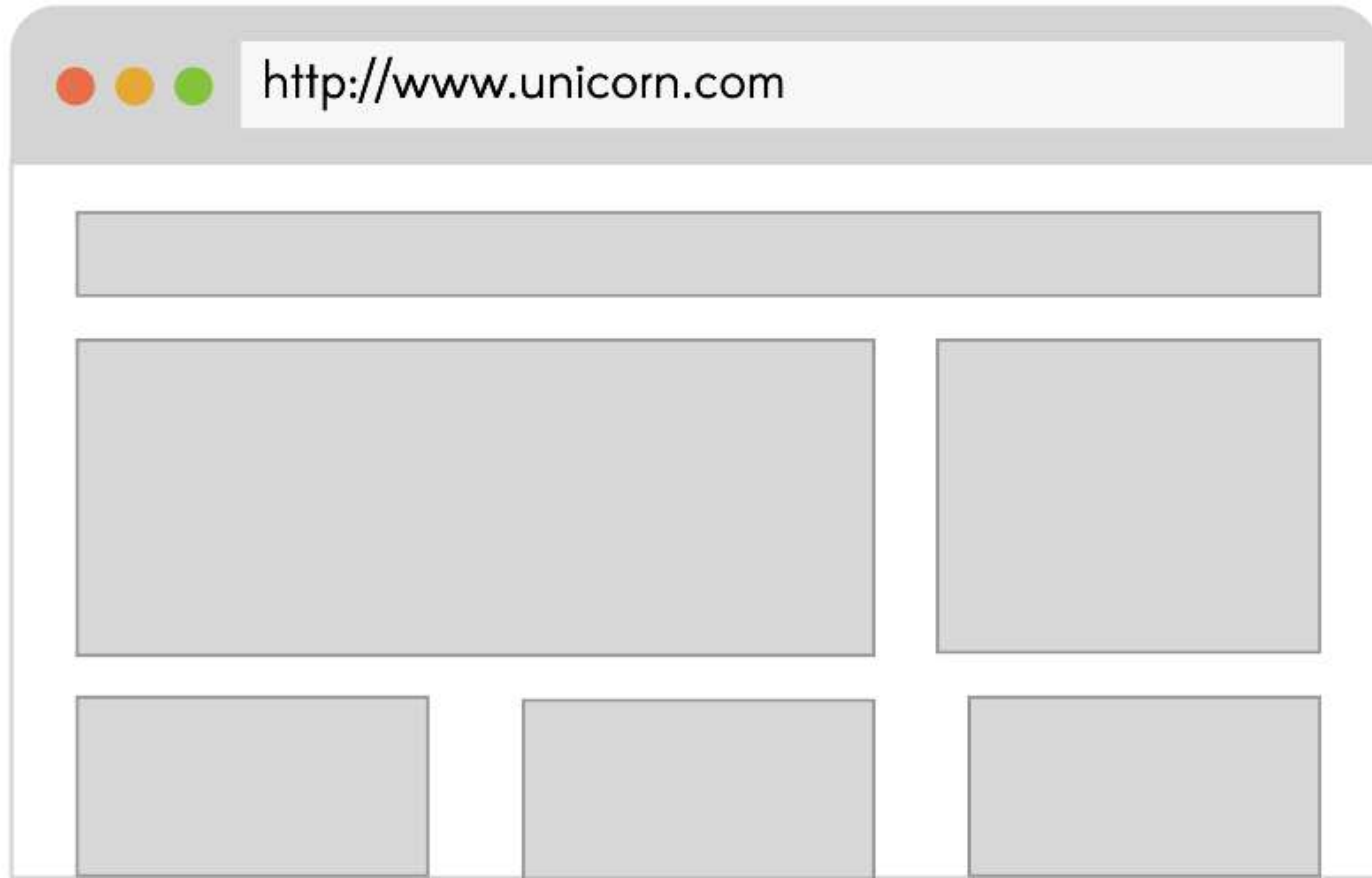
Basic HTML5 template

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>title</title>
    <meta name="description" content="A short description of your website">
    <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](https://en.wikipedia.org/wiki/Munsell_color_system) (a 20th-century system for numerically describing colours, based on equal steps for human visual perception) distinguishes 100 hues.

```
<p class="tl-example-4">Accordingly, <a href="https://en.wikipedia.org/wiki/Munsell_color_system">the Munsell colour system</a> (a 20th-century system for numerically describing colours, based on equal steps for human visual perception) distinguishes 100 hues.</p>
```

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

Full list of inline-level elements available [here](#).



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 pe
  <li>HEAT butter in large nonstic
  <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;  
}
```

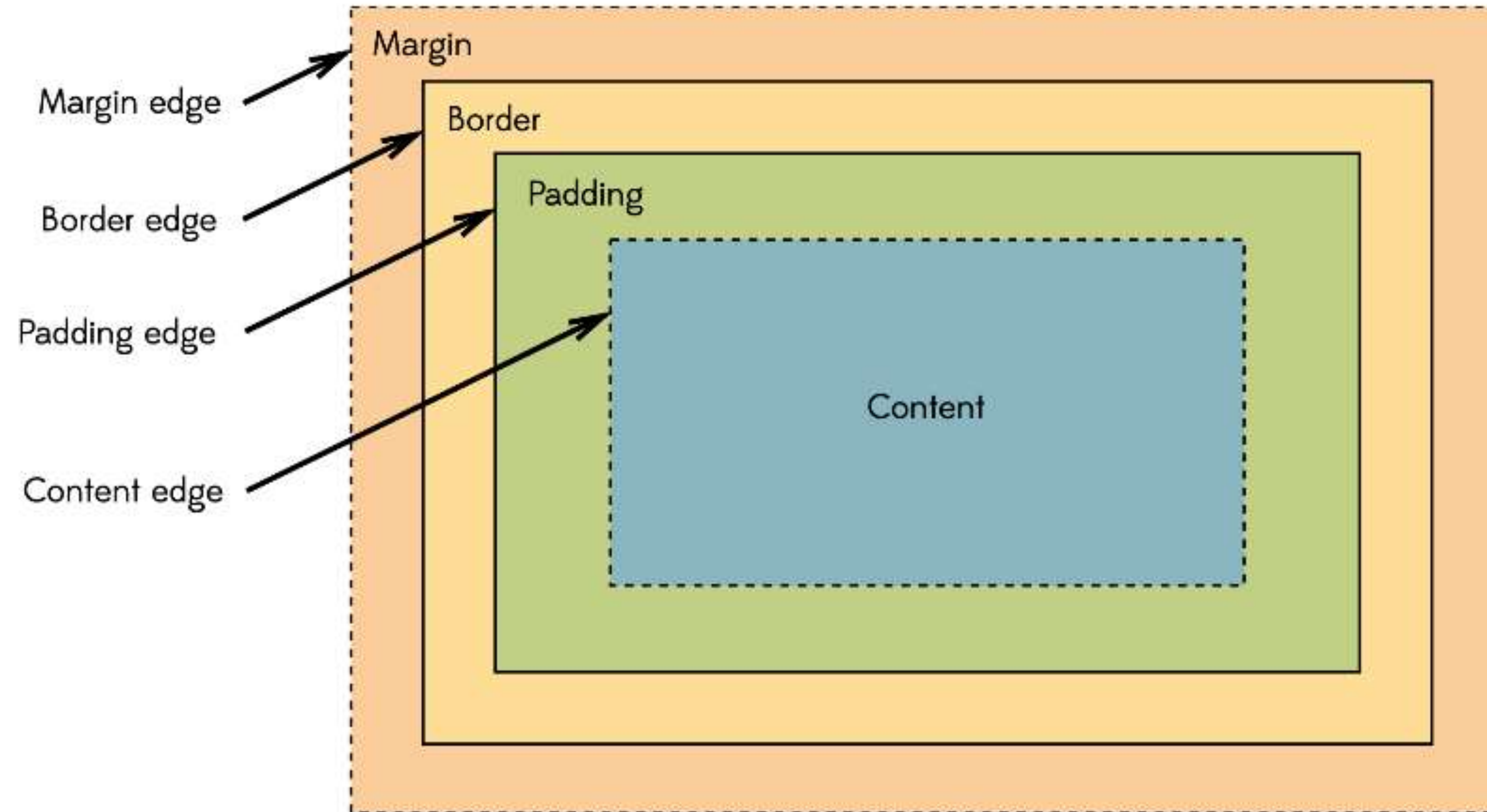
- [Link 1](#)
- [Link 2](#)
- [Link 3](#)

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:



- Content
- Padding
- Border
- Margin

The box model, visualised



When to use margin

Margin controls the space between elements.

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

```
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

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

$0-\infty$

Inline styles

$0-\infty$

IDs

$0-\infty$

Classes,
attributes and
pseudo-classes

$0-\infty$

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



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 `` tag

```

```

- 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*.

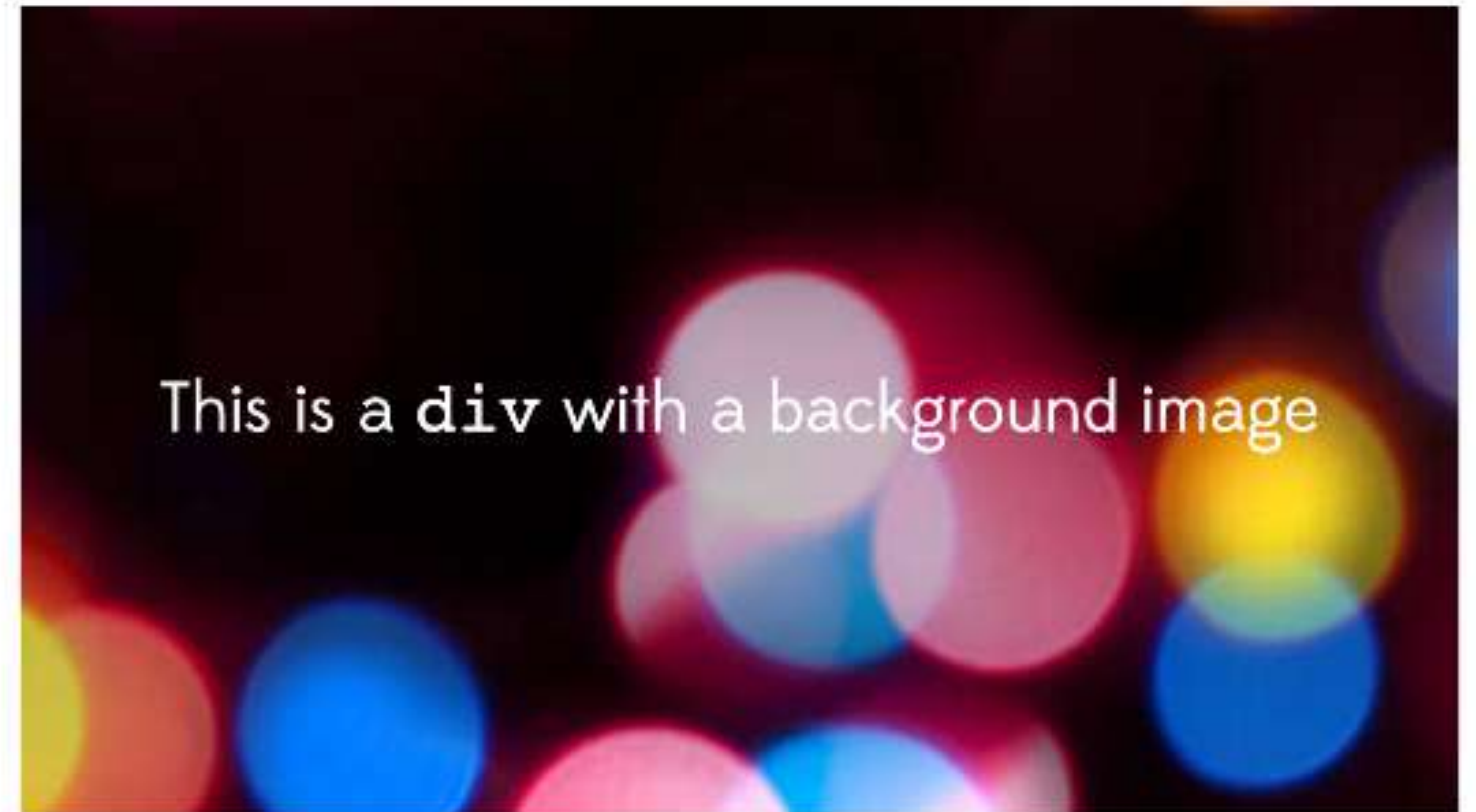


Setting background images

`background-image` can use relative or absolute paths

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

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

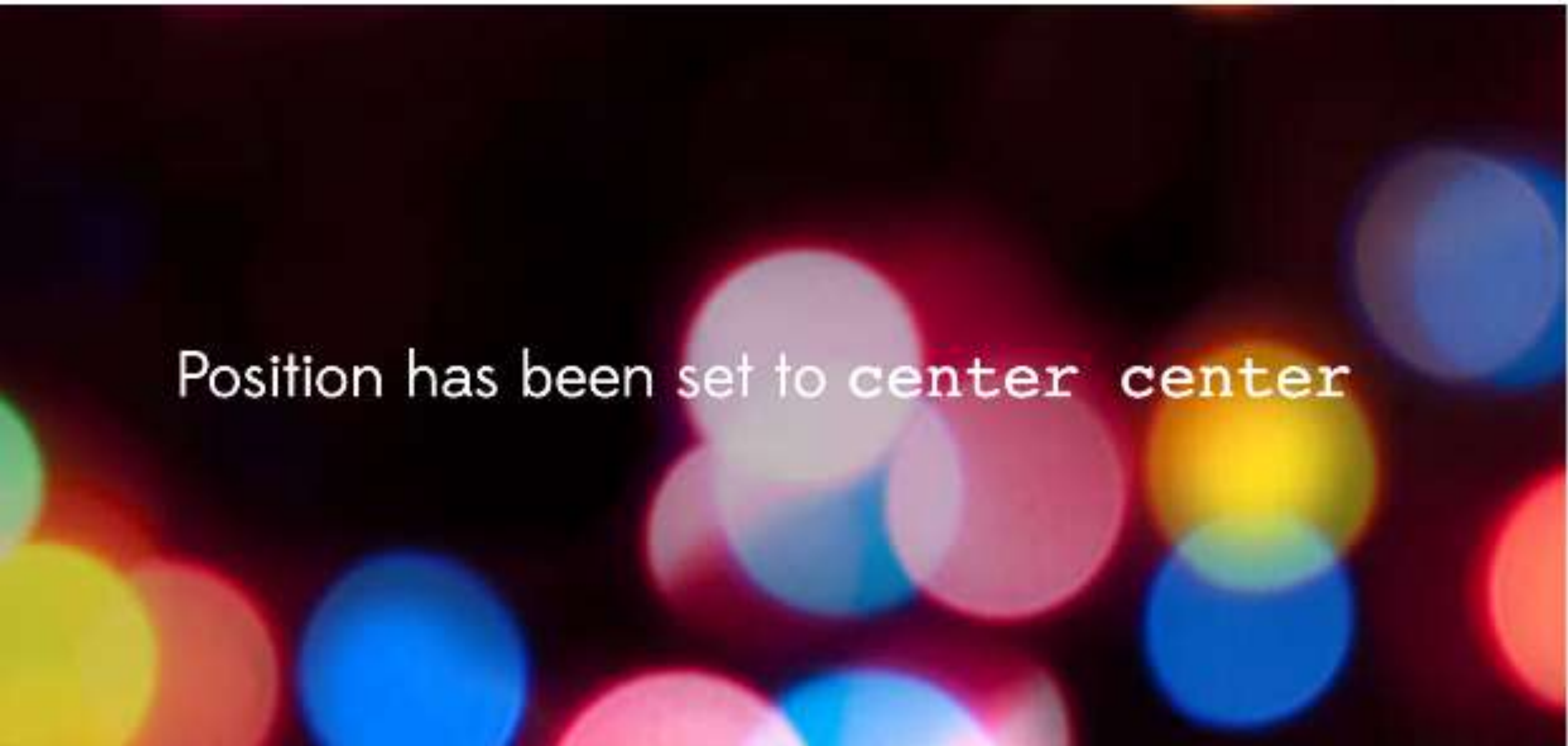


background-position

This is used to set the position of the image

```
div {  
  background-color: #170104;  
  background-image: url('img/background.jpg');  
  background-position: center center;  
}
```

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

A dark background with a pattern of colorful, out-of-focus bokeh lights in shades of blue, yellow, pink, and green. The lights are distributed evenly across the frame.

Position has been set to center center

A dark background with a pattern of colorful, out-of-focus bokeh lights in shades of blue, yellow, pink, and green. The lights are concentrated in the bottom-left corner, with the rest of the frame being mostly dark.

Position has been set to 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;  
}
```

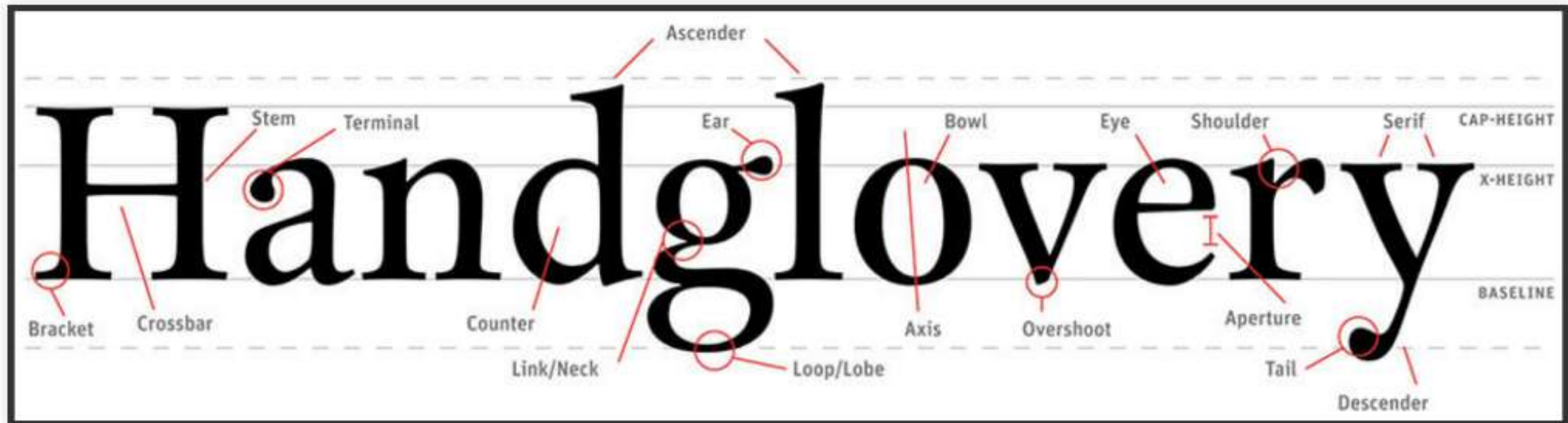


Web typography



Typography terminology

Typography is, quite simply, the art and technique of arranging type.



General guidelines

- Default font-size is 16px. Don't go smaller than that for body copy.
- Adjust the measure to around 45-75 characters per line.
- Have a line-height of around 140%.
- Ensure sufficient contrast between the text and the background.
- Create a visual hierarchy, through size or text styles.



Web fonts

Don't go crazy with the number of fonts used. Usually 2 is enough.

Serve the font files yourself using `@font-face` or use a hosted service, like Google Fonts, using `@import`

```
@import url(https://fonts.googleapis.com/css?family=Fjalla+One|Average);
```

```
body {  
  font-family: "Average", serif;  
}  
  
h1 {  
  font-family: "Fjalla One", sans-serif;  
}
```



Browser developer tools



What are devtools?

Every modern browser has a powerful suite of developer tools.

- Inspect currently loaded HTML, CSS and Javascript
- Check which assets were loaded
- See how long it took for your assets to load
- Provides details for each HTTP request/response
- And a lot more...



How to use devtools?

- Keyboard shortcut
 - Windows: Ctrl + Shift + I
 - Mac: Option + Command + I
- Menu bar
 - Firefox: Tools › Web Developer › Toggle Tools
 - Chrome: View › Developer › Developer Tools
 - Safari: Develop › Show Web Inspector
 - Opera: Developer › Web Inspector
- Context menu
 - Right-click on an element in the browser and select Inspect

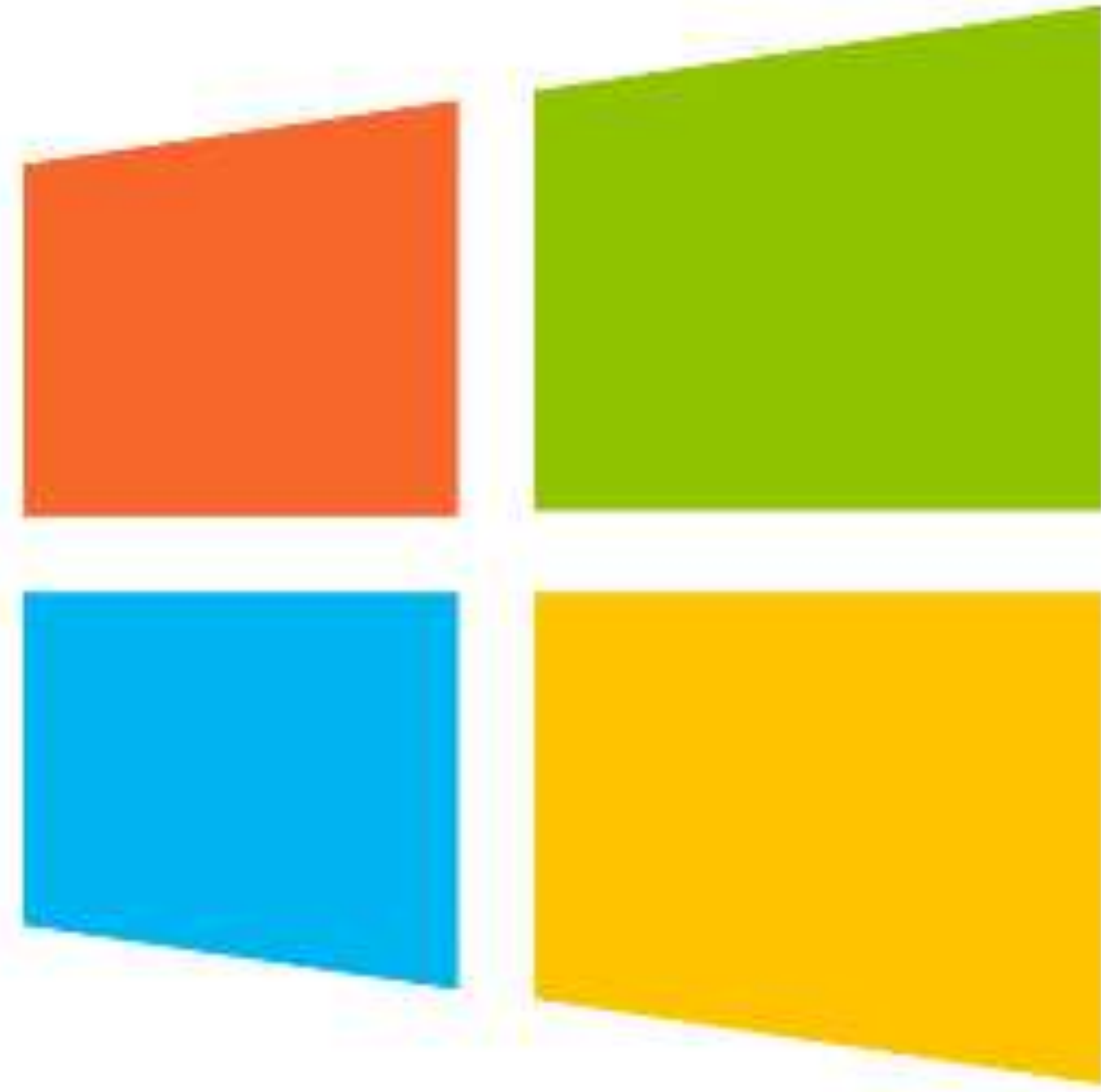


**Let's talk about
computers**



Operating systems

Your computer's personal manager

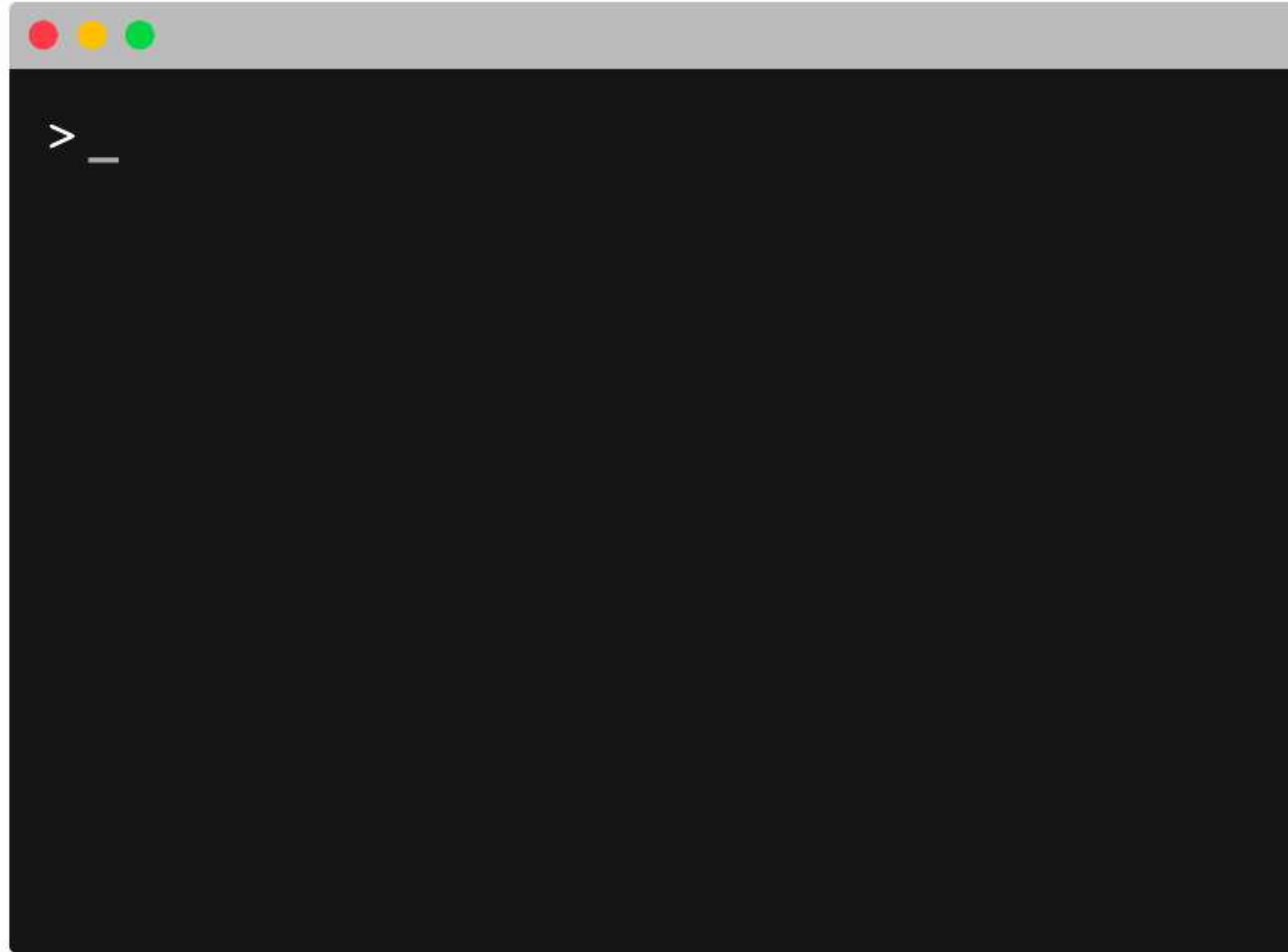


File systems

How computers store information

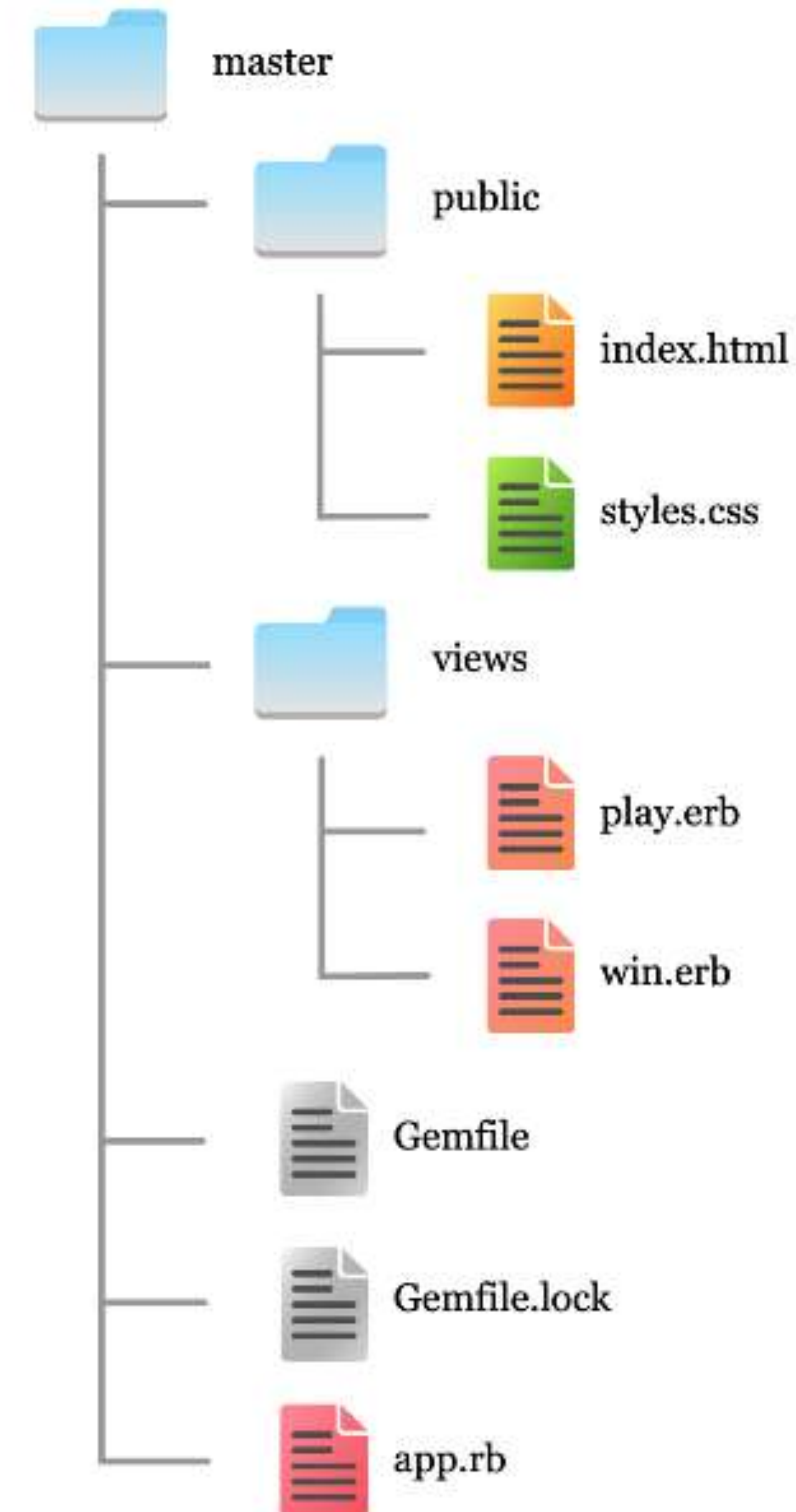


Command line interface



Guess The Number

1. Fork the repository on [Github](#)
2. Follow the instructions [here](#)
3. `.erb` files behave in exactly the same way as HTML
4. Gemfile handles ruby gem dependencies



Deploying your website



Web hosting



- Renting server space for your website
- Shared hosting, Dedicated hosting, VPS (Virtual Private Server)
- Managed vs Unmanaged



Domain name registration

Domain names map to your **server's IP address**

Provides a human-readable name for your site

Top-level domains, e.g. .com, .io, .org, .edu

Register domain names with a **domain name registrar**



Getting your site online

Upload the files onto your server, **Git** or **FTP**

Servers usually run on **Linux**

Type of hosting determines level of control

Must be running a HTTP server, common ones are **Apache** or **Nginx**



Basic Git Terminology

Git is a **version control system**

Repository: A place that stores all your project files

Commit: Used to take a "snapshot" of the state of your project

Branch: Used to develop features without disrupting the main code base

Fork: A copy of a repository, allows you to freely experiment with changes without affecting the original project.



Commit your work

Make sure you're in your project directory, otherwise the following commands will not work.

Stage the files you've been working on

```
git add .
```

Write a sensible commit message

```
git commit -m "Add styling to Guess the Number app"
```

Push your changes to the remote repository

```
git push origin master
```



What is Heroku?

A place to host your application

Connects to GitHub for easier deployment

<https://www.heroku.com/>

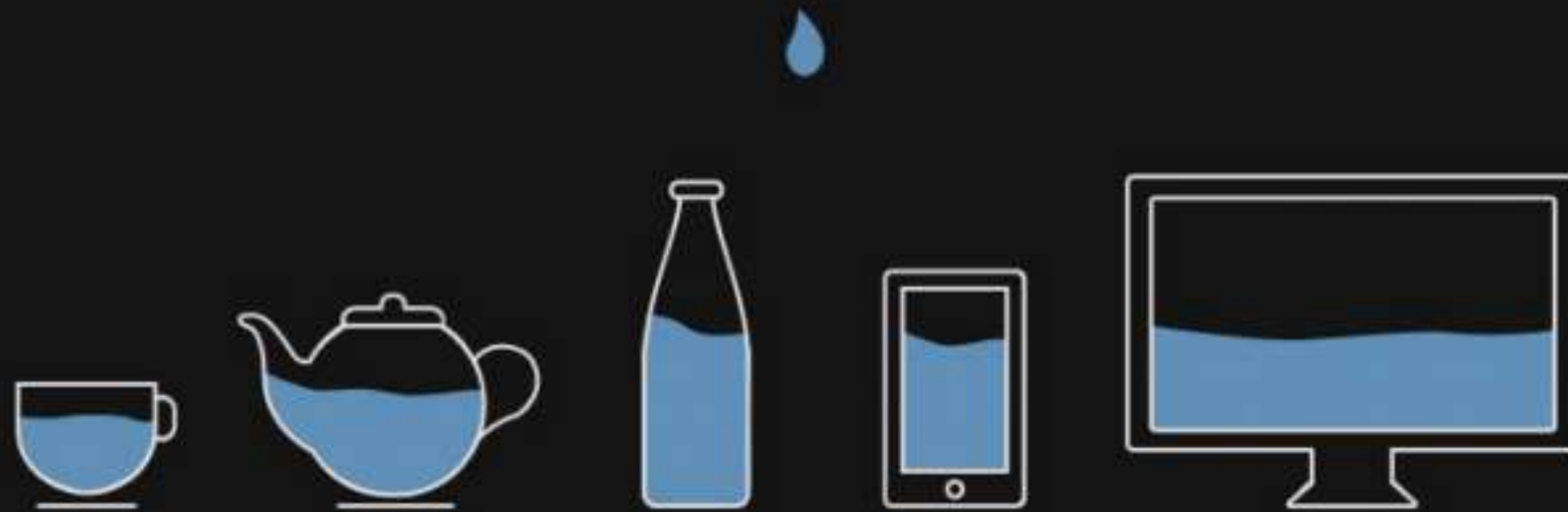


Responsive web design



The web is fluid

CONTENT IS LIKE WATER



“ You put water into a cup it becomes the cup.
You put water into a bottle it becomes the bottle.
You put it in a teapot, it becomes the teapot. ”

Josh Clark (originally Bruce Lee) - Seven deadly mobile myths
Illustration by Stephanie Walter

Image by [Stephanie Walter](#)



Browsers are responsive by default

Our problems with websites are self-created

Satirical but true example

*"The control which designers know in the print medium, and often desire in the web medium, is simply a function of the limitation of the printed page."
- John Allsop, A Dao of Web Design (2000)*



Media queries

Tell the browser to display the site's content in a particular way when certain conditions are true.

Basic syntax as follows:

```
@media <media-type> (<media-feature>);
```

- **media-type:** e.g. screen, print
- **media-feature:** e.g. width, height



Mobile-first media queries

```
.o-flex3__item {  
  flex: 0 0 100%  
}  
  
@media all and (min-width: 30em) and (max-width: 47.9375em) {  
  .o-flex3__item {  
    flex: 0 0 49.15254%  
  }  
}  
  
@media all and (min-width: 48em) {  
  .o-flex3__item {  
    flex: 0 0 32.20339%  
  }  
}
```

- Default styles are for small screen sizes
- Add styles as the screen size gets bigger
- Can chain additional media features



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



Resources



To find out more...

- [HTML & CSS is hard \(but it doesn't have to be\) \(online course\)](#)
- [Getting started with the Web \(website\)](#)
- [HTML & CSS: Design and Build Web Sites by Jon Duckett \(book\)](#)
- [Designing with Web Standards by Jeffrey Zeldman \(book\)](#)



The End



<https://www.chenhuijing.com>



@hj_chen



@hj_chen



@huijing

Font used is [Edmond Sans](#), by James T. Edmondson

