

# WEB DEVELOPMENT: BREAKING IT DOWN

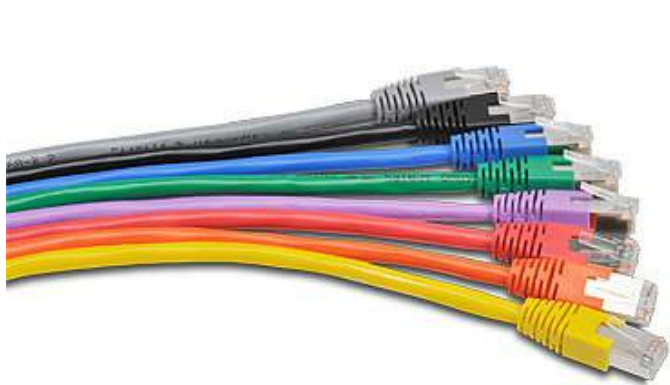
By [Chen Hui Jing](#) / [@hj\\_chen](#)



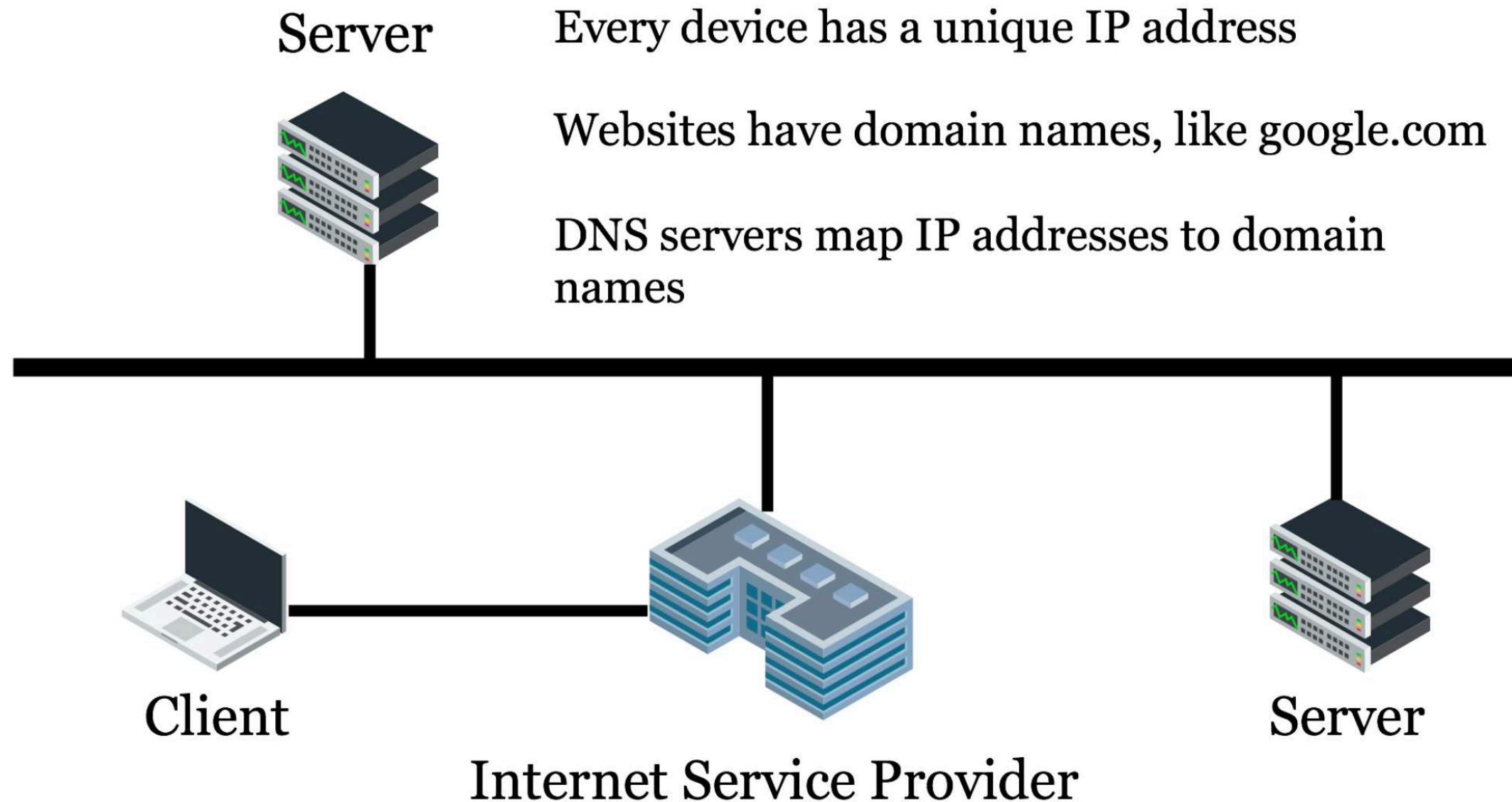
# INTERNET BASICS

# WHAT IS THE INTERNET?

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



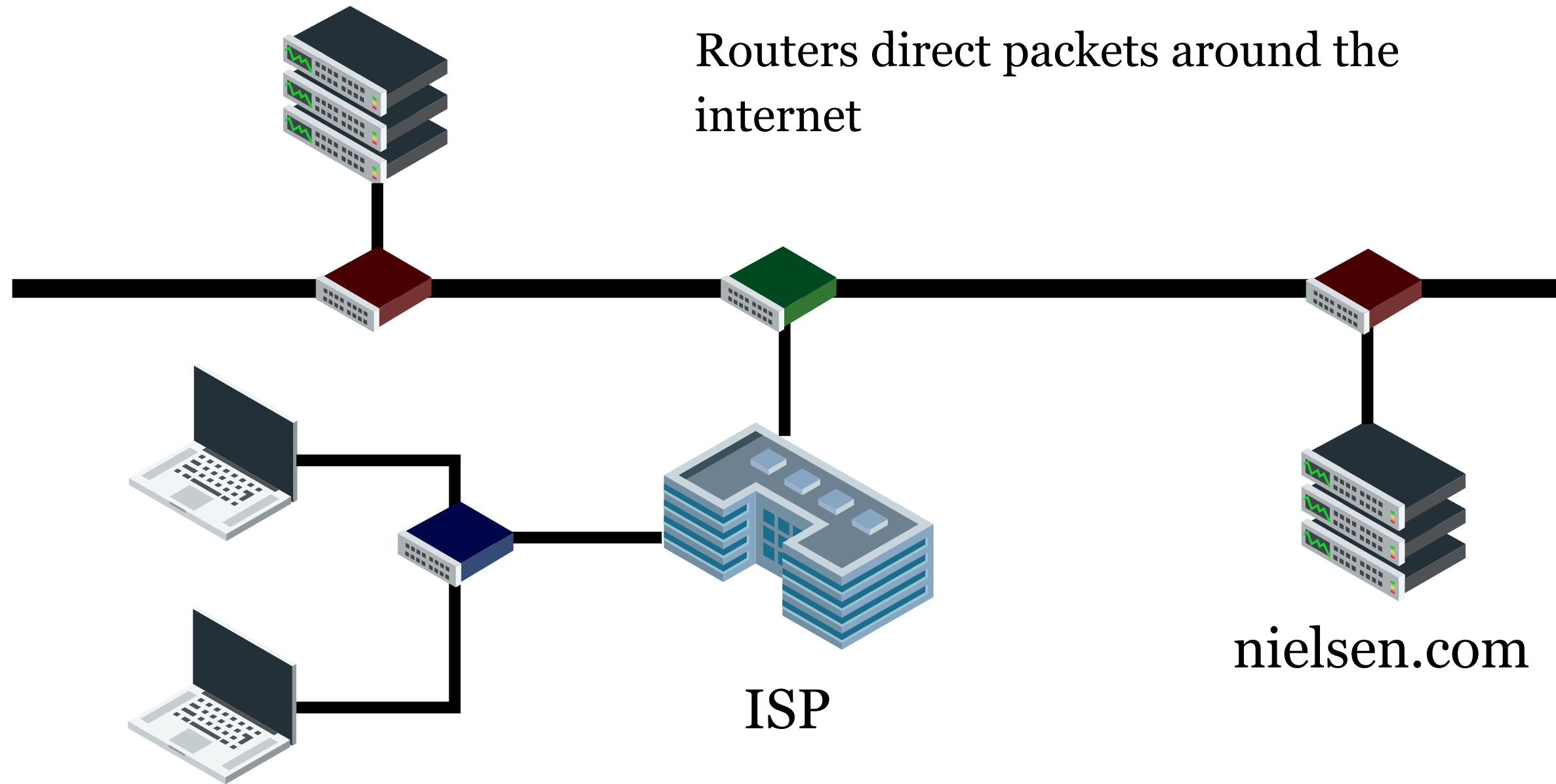
# HOW DOES THE INTERNET WORK?



facebook.com

Data is broken up into packets

Routers direct packets around the internet



# INTERNET ≠ WORLD WIDE WEB

	Internet	World wide web
<b>Estimated year of origin</b>	1969	1993
<b>Name of first version</b>	ARPAnet	NSFnet
<b>Comprises</b>	Network of devices, copper wires, fibre-optic cables and wireless networks	Files, folders & documents stored in various computers
<b>Governed by</b>	Internet Protocol (IP)	Hypertext Transfer Protocol (HTTP)
<b>Dependency</b>	None, everything else depends on it	The internet
<b>Nature</b>	Hardware	Software

Source: *Diffen*

# PROTOCOLS

A **protocol** is an agreed set of rules and standards that allow devices to communicate with each other.

<b>Internet Protocol (IP)</b>	To route information to the proper address
<b>Hypertext Transfer Protocol (HTTP)</b>	For the transmission of web pages over the internet
<b>Transmission Control Protocol (TCP)</b>	To create packets, reassemble them, make sure none were lost in transmission
<b>Simple Mail Transfer Protocol (SMTP)</b>	For computers to send email
<b>Post Office Protocol (POP)</b>	For computers to receive email
<b>File Transfer Protocol (FTP)</b>	For copying files over a network from one computer to another

# **ABOUT WEB DEVELOPMENT**

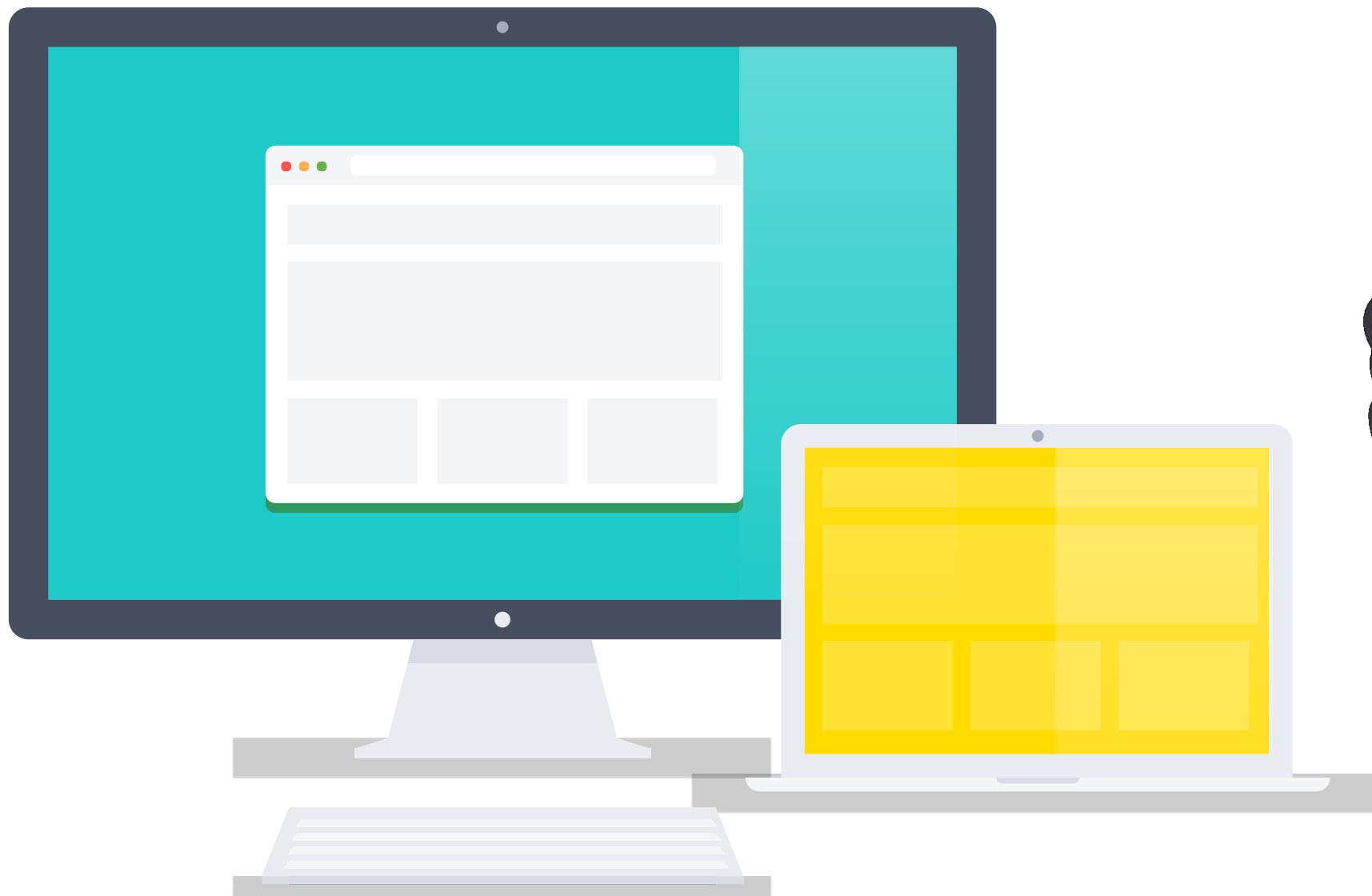


# **WHAT IS WEB DEVELOPMENT?**

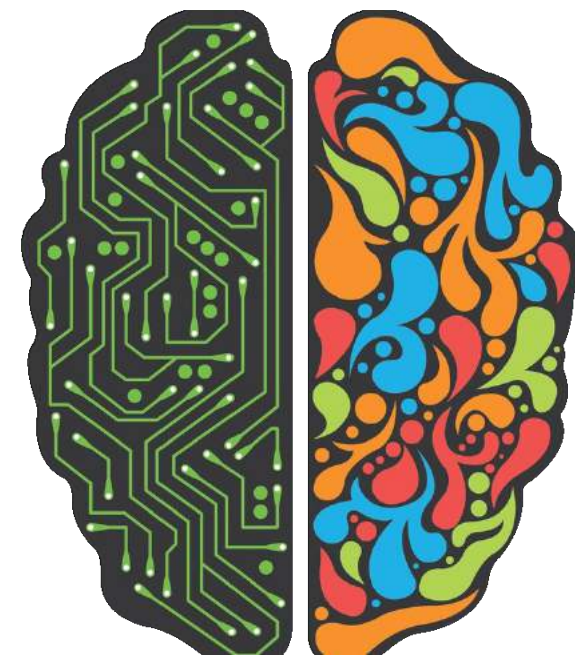
The process of building websites

# 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](#)



# **PROGRAMMING FOR THE WEB**

# FRONT-END



Client-side



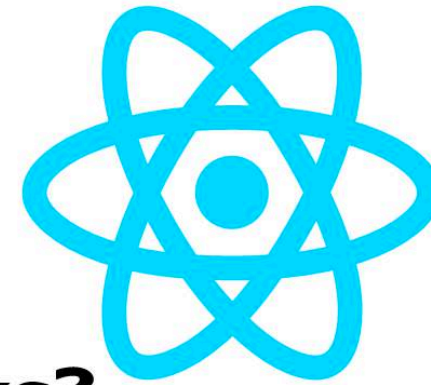
# BACK-END



Server-side



# FEELING OVERWHELMED?



Frameworks?



Libraries?



Tools?



# HTML AND CSS ARE THE FOUNDATION OF THE WEB

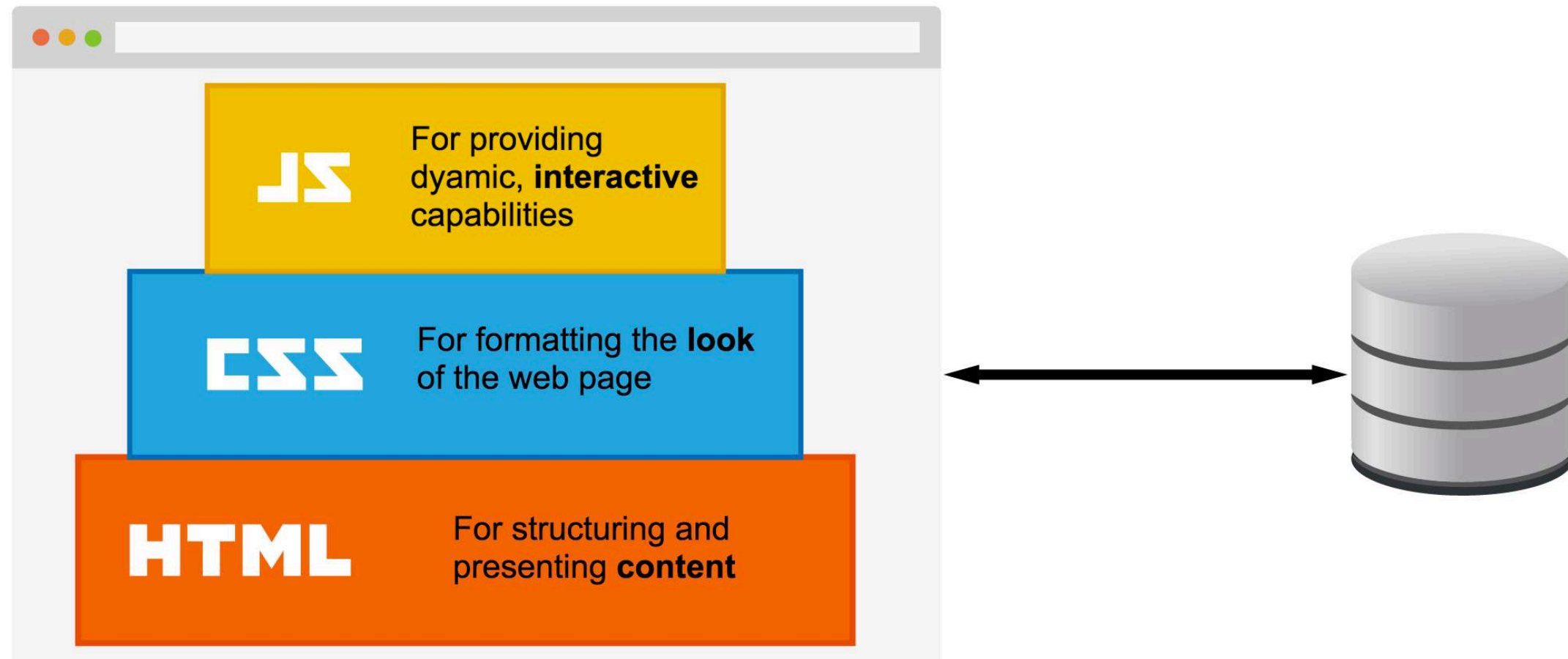
**HTML**



**CSS**



# BASIC TECHNOLOGY STACK







---

**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**

# HISTORY OF HTML

- Invented by **Tim Berners-Lee** in 1989 as a means for easy linking of references in research papers using hypertext
- Created the **Hypertext Transfer Protocol (HTTP)** for retrieving text from other documents via hypertext links
- Specifications were introduced as a guide for standardisation among browser vendors

# 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>
    
    <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>
- <div>
- <dl>
- <dt>
- <figcaption>
- <figure>
- <hr>
- <li>
- <main>
- <ol>
- <p>
- <pre>
- <ul>
- <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>
- <br>
- <cite>
- <code>
- <data>
- <dfn>
- <em>
- <i>
- <kbd>
- <mark>
- <q>
- <rp>
- <rt>
- <rtc>
- <ruby>
- <s>
- <samp>
- <small>
- <span>
- <strong>
- <sub>
- <sup>
- <time>
- <u>
- <var>
- <wbr>
- <area>
- <audio>
- <map>
- <track>
- <video>
- <embed>
- <object>
- <param>
- <source>
- <canvas>
- <noscript>
- <script>
- <del>
- <ins>

# 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

# **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 {}
```



# 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



# JAVASCRIPT

# JAVASCRIPT

- A cross-platform, object-oriented scripting language
- Can run directly in the browser, does not need to be compiled first
- Allows greater control of web page behaviour
- Is a full-fledged programming language, with operators, control structures and statements

# **BRIEF HISTORY OF JAVASCRIPT**

- Invented in 10 days by **Brendan Eich** of Netscape
- Was taken to the **European Computer Manufacturers Association (ECMA)** for standardisation in 1996
- Currently on the 6th edition, known as **ECMAScript 2015**, or **ES6**

# CLIENT-SIDE VS SERVER-SIDE

## Client-side Javascript

- Provides a means of controlling the browser and its Document Object Model
- Examples are form validation, load more buttons, image galleries and so on

## Server-side Javascript

- Usually refers to **node.js**
- Provides a customised response based on user (client) requests
- For example, displaying your profile information after you logged in

# WHAT IS AN API?

- Stands for **Application Programming Interface**
- A means for your web page to access data from another source
- Usually a URL, that when accessed, returns data in a standardised format (often, JSON)

# EXAMPLE WEATHER API

```
// 20160211134054
// http://api.openweathermap.org/data/2.5/weather?id=1880251&appid=

{
  "coord": {
    "lon": 103.8,
    "lat": 1.37
  },
  "weather": [
    {
      "id": 803,
      "main": "Clouds",
      "description": "broken clouds",
      "icon": "04d"
    }
  ]
}
```

# WEB APIS

- Give the browser and server the ability to perform many tasks
- DOM is actually an API that allows access and modification of the web page
- Device APIs let web developers access hardware features
- Communication APIs let devices, applications and web pages talk to each other, e.g. web notifications, push etc.





# RESOURCES

# TO FIND OUT MORE...

- [Dash](#) (online course)
- [Codecademy](#) (online course)
- [Mozilla Developer Network \(MDN\)](#) (website)
- [HTML & CSS: Design and Build Web Sites](#) by [Jon Duckett](#) (book)
- [Eloquent Javascript](#) by [Marijn Haverbeke](#) (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