Build Your First Web App

Build a full, working web application using the most widely available tools and technologies.

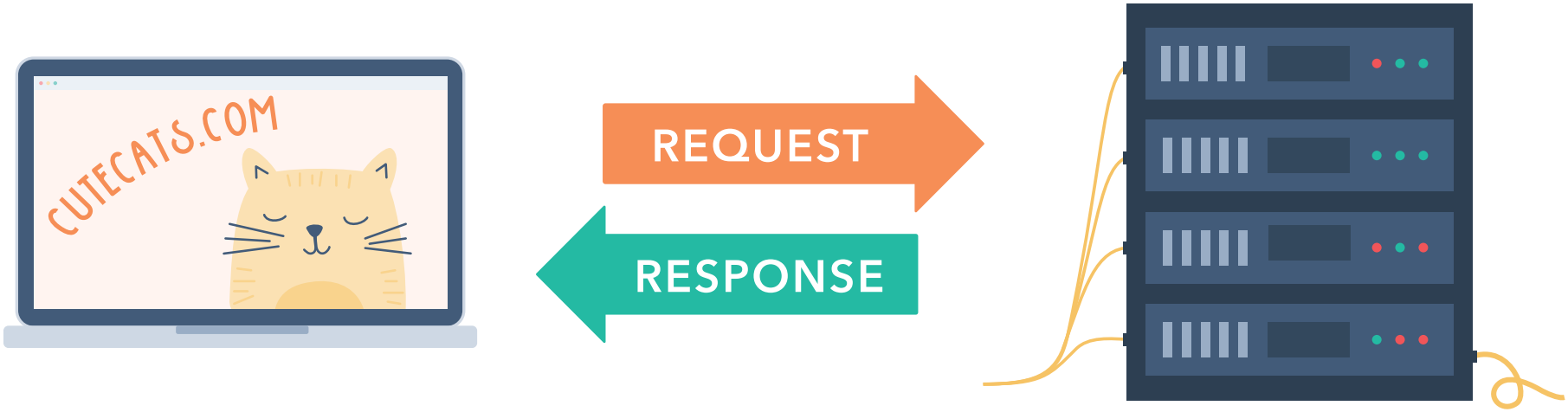
This course assumes a **basic working knowledge of programming concepts**.

Find instructions and links in the **Resources** section below.

Web Application

A collection of code that's stored on a remote server (not your computer) and delivered over the internet

Requests and Responses



Anatomy of a URL

Protocols

THE PROTOCOL

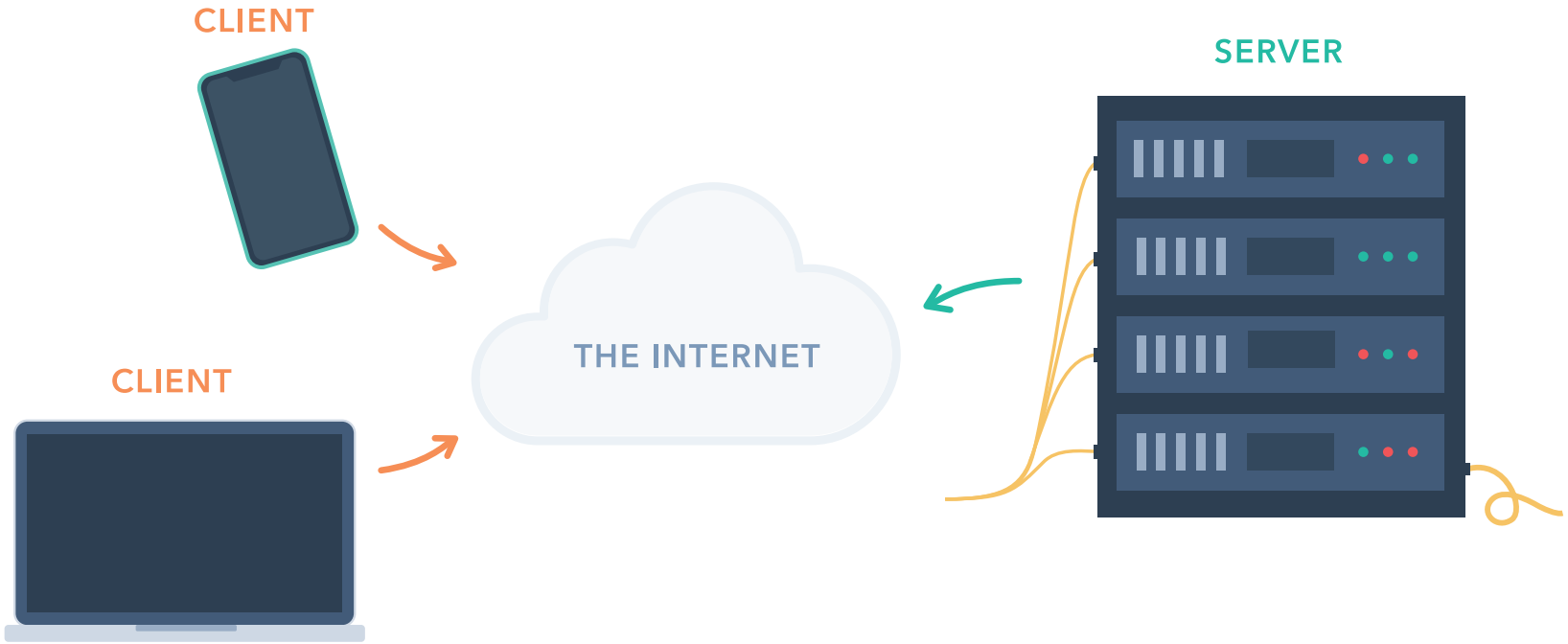
https://www.cutecats.com/5-cute-cats-to-brighten-your-day

- **Protocol:** Declares how your web browser should communicate with a web server when sending or fetching a web page or document
- **HTTP (Hyper Text Transfer Protocol):** The most common protocol
- **HTTPS:** Security-focused counterpart of HTTP, where the browser encrypts the data it's sending

Servers and URLs

- **Server:** Stores and serves information (like your application)
- **Clients:** All the devices requesting that information (like your laptop or phone)

Client-Server Relationships



Environments

- **Production:** Stable, secure, and ready for traffic
- **QA (or Quality Assurance):** Environment for testing an application on the server before it goes to users
- **Local:** A server on your computer, often used when writing code

Heroku

- Production server host
- Cloud platform
- Free service
- Resources available

Front-end

The user interface, or what you see on the screen

Written in HTML, CSS, JavaScript

Back-end

Stores, transforms, and serves data

Can use many languages; we're using JavaScript

Front-end Anatomy

HTML

The skeleton, or how the page is structured

JavaScript

The muscles, or how the page changes

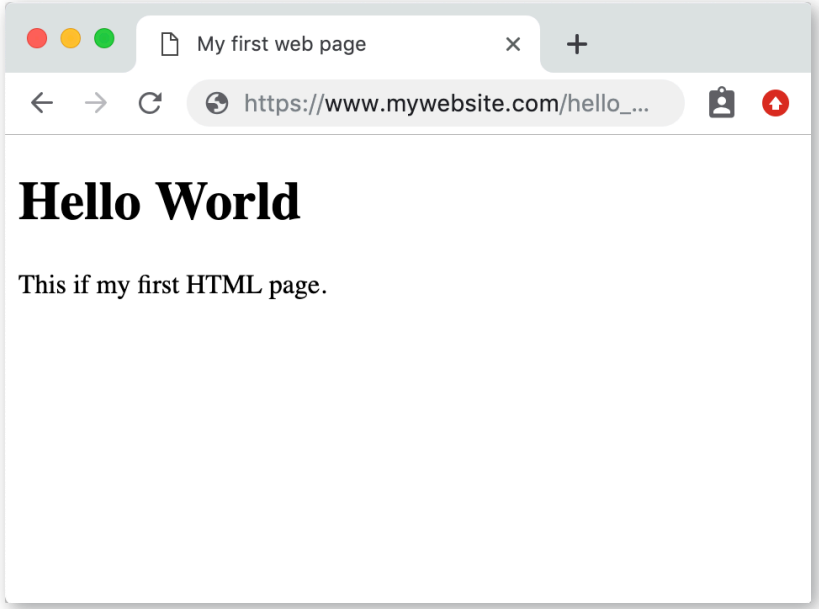
CSS

The skin, or how the page looks

HTML (The Skeleton)

hello_world.html

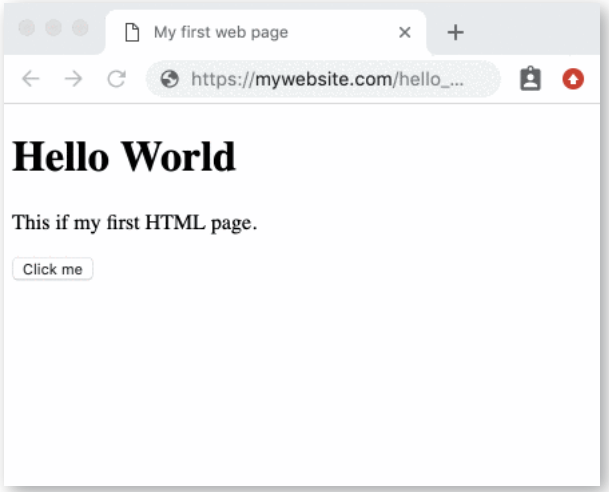
```
<!DOCTYPE html>
<html>
<head>
    <title>My first web page</title>
</head>

<body>
    <h1>Hello World</h1>
    <p>This if my first HTML page.</p>
</body>
</html>

```



JavaScript (The Muscles)

JS functions.js

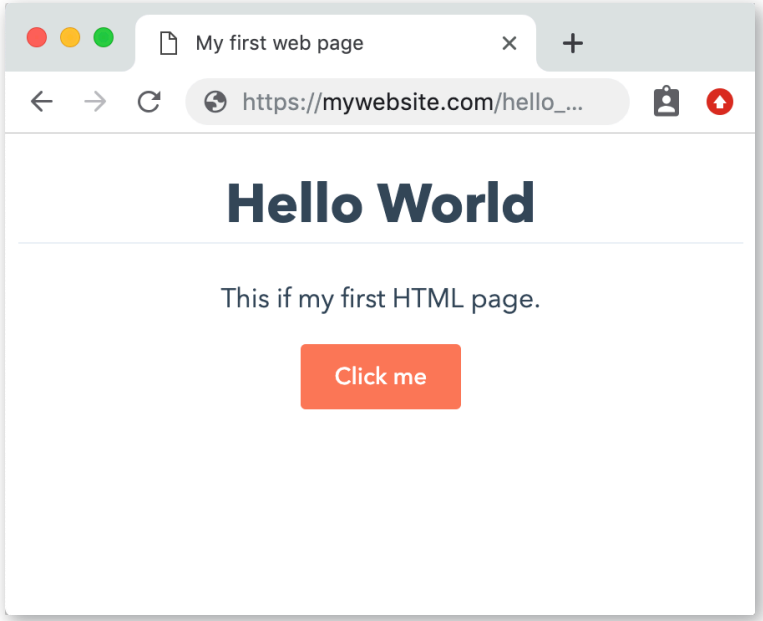
```
1   function changeText() {
2       document.getElementById("demo").innerHTML
3       = "This web page is interactive!";
4   }
5
```



CSS (The Skin)

styles.css

```css
body, button {
    font-family:"Avenir Next", sans-serif;
    text-align: center;
    color: #33475b;
}
h1 {
    border-bottom: 1px solid #eaf0f6;
}
button {
    background-color: #ff7a59;
    color: white;
    border: none;
    border-radius: 3px;
    padding: 10px 20px;
    font-size: 14px;
    font-weight: 500;
}
```



Database

Used for storing your app's dynamic data

Song	Artist	Album
The Mother	Brandi Carlile	By The Way, I Forgive You
Wilhelm Scream	The Bamboos	Medicine Man
Fireworks	First Aid Kit	Ruins
The Chain	Fleetwood Mac	Rumors
Navy Blues	Aisha Badru	Pendulum
Light On	Maggie Rogers	Heard It In A Past Life
A Case of You	Joni Mitchell	Blue

APIs

- Stands for Application Programming Interface
- How the front-end of your app communicates with the back-end
- Lets you retrieve and modify data in your database

Requests

```
HTTP GET /weather/new_york_city
```

REQUEST

RESPONSE

```
{
    {"new_york_city": {
        "temperature":"63",
         "willRain":"true"
        }
}
```

Node.js

- Open-source server environment
- Written in JavaScript
- Easy to use

Express.js

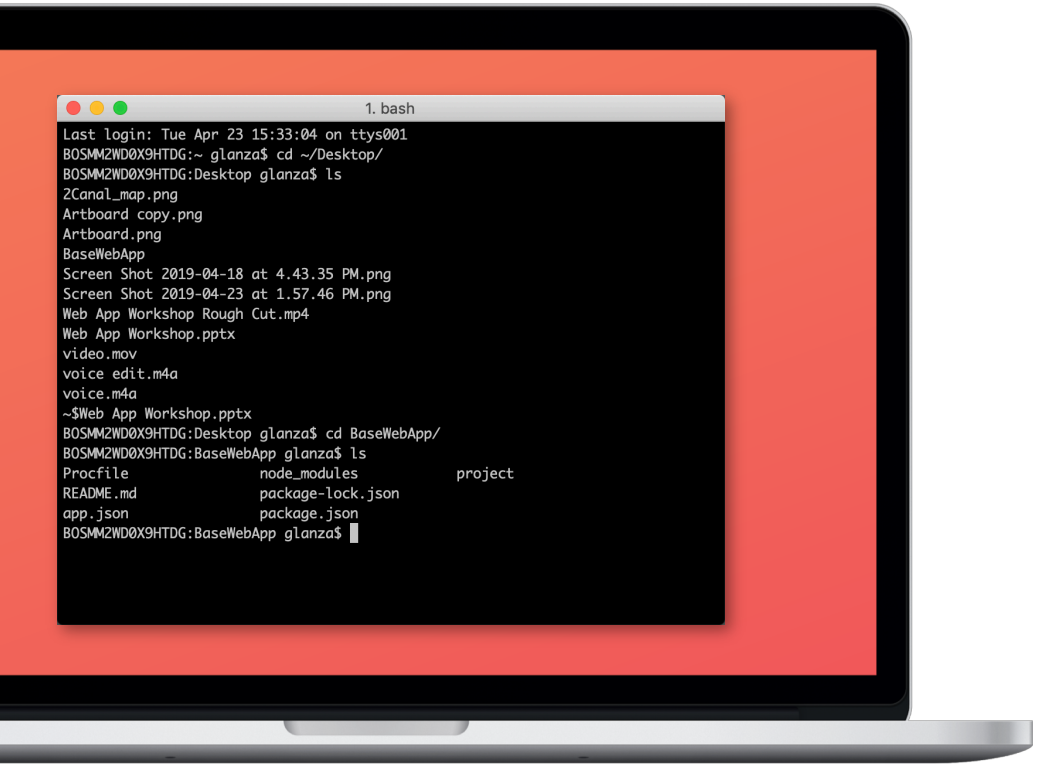
- Framework for creating web and mobile applications
- Popular and quick to set up

git

- A free and open source distributed version control system
- Makes it easy to track changes to files
- Great for working with multiple people and tracking progress

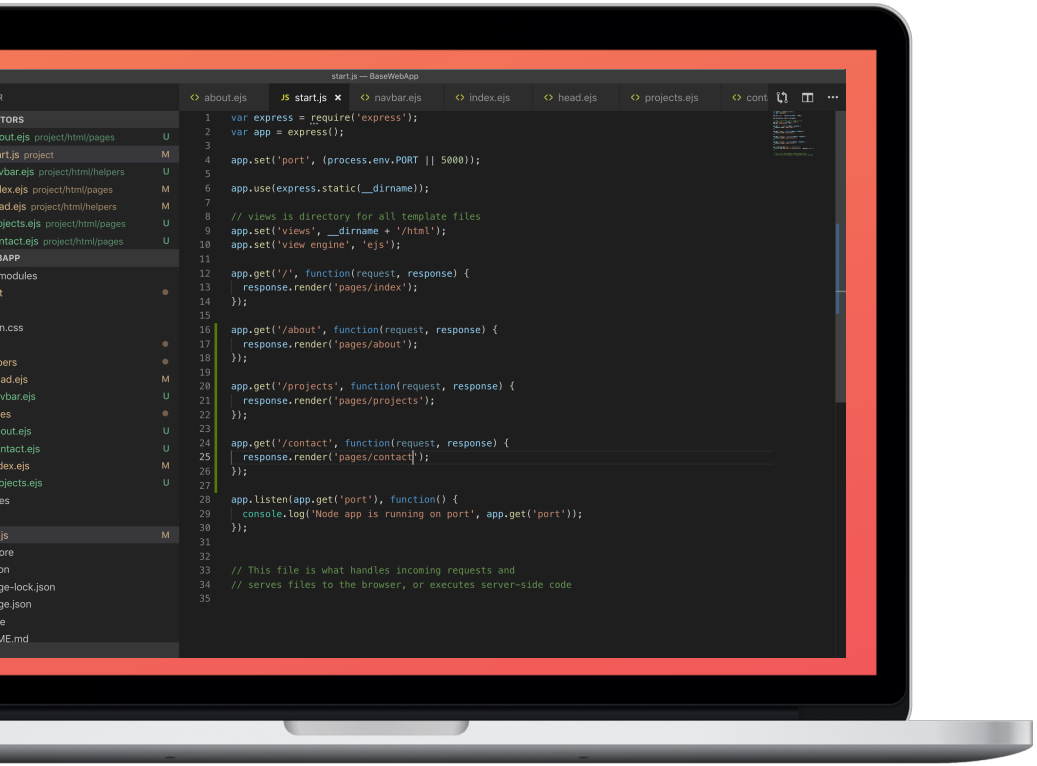
GitHub

- A web-based git repository hosting service
- Makes it easy to share code and contribute to others' projects



Terminal

- Basic way to communicate with applications
- Simple lines of text to run specific commands



Code Editor

- Used when writing or modifying your application's code
- Visual Studio Code

What to expect:

1. Build your web app.
2. Build out additional functionality.
3. Integrate your app with an external API.
4. Set up a database.

Thank you