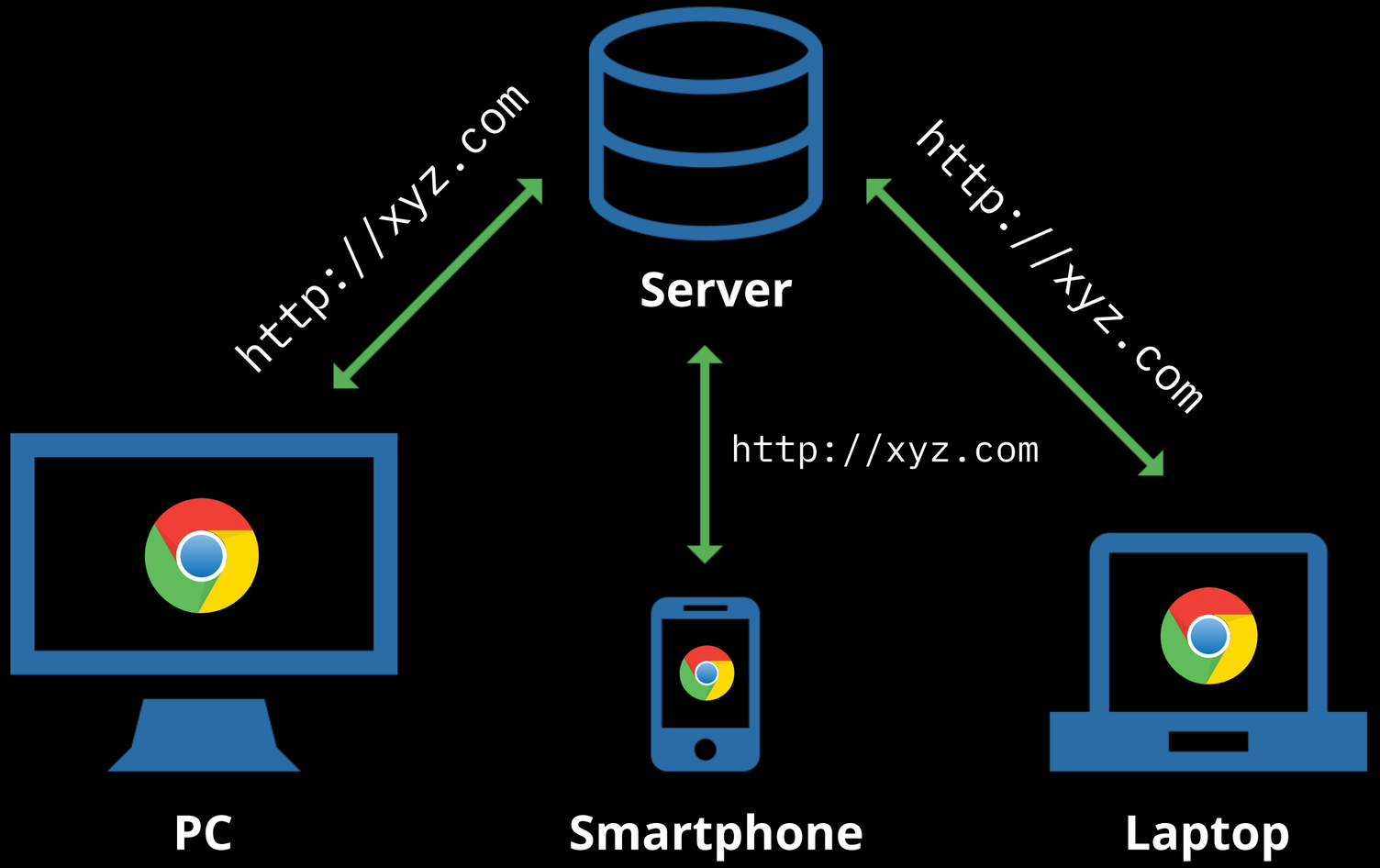
THE WEB

PART 1: INTRODUCTION

What we will Discuss?

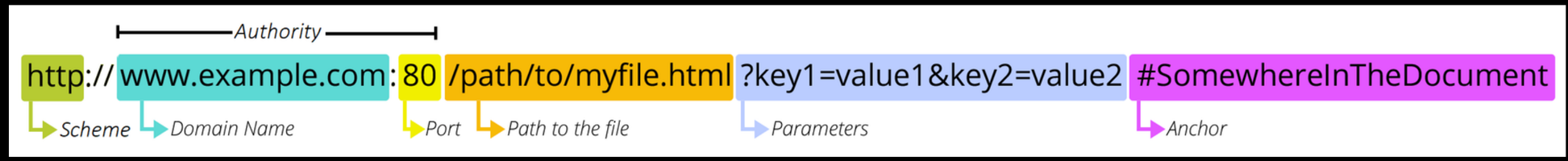
- How the web works.
  - Clients and Servers
  - URL
  - HTTP/HTTPS Protocol
- What is website.
  - HTML
  - CSS
  - JavaScript
  - Cookies
  - Local Storage and Database
- Chrome Dev-Tools
- Requests and Responses

What is Server and Client?



What is URL?

- URL stands for Uniform Resource Locator.
- A URL is nothing more than the address of a given unique resource on the Web.
- In theory, each valid URL points to a unique resource. Such resources can be an HTML page, a CSS document, an image, etc.



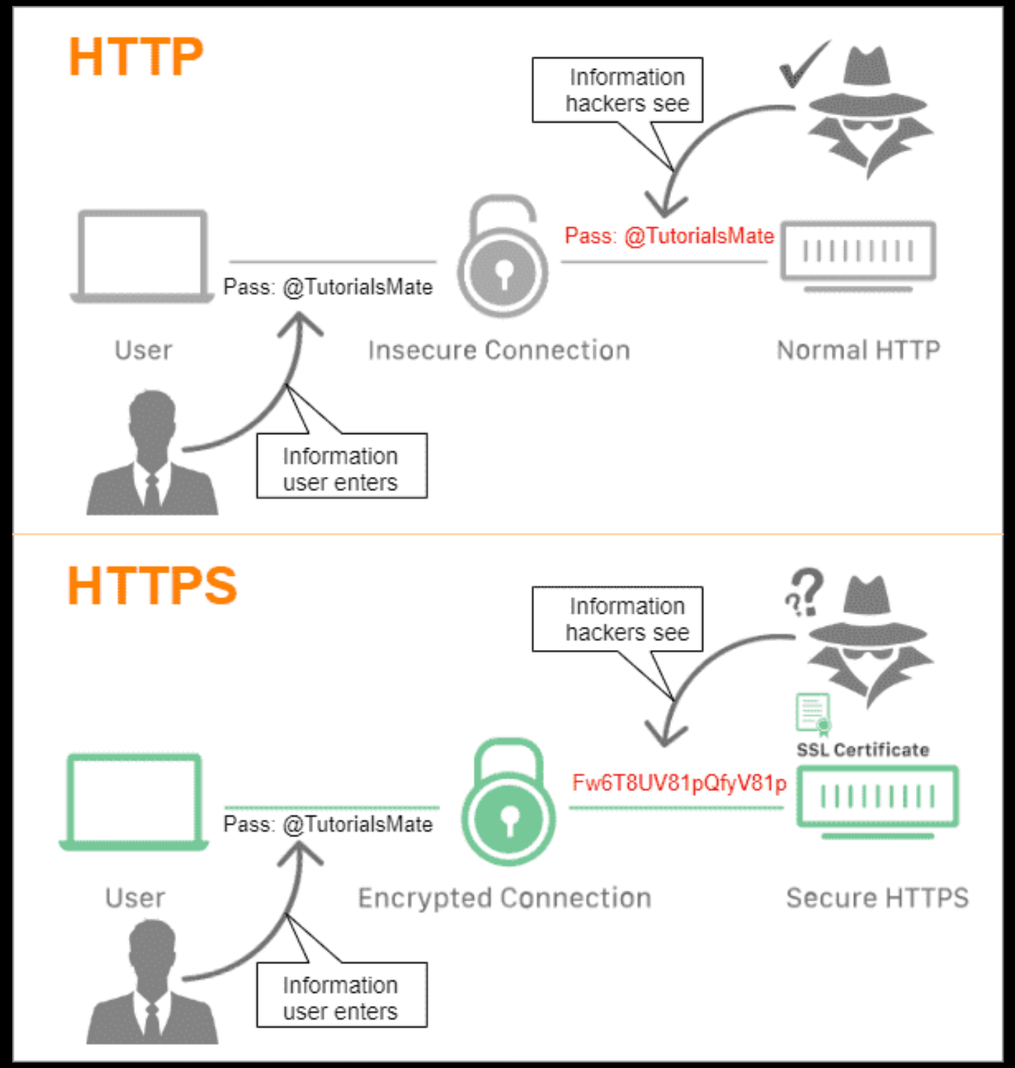
HTTP vs HTTPS

What is HTTP?

- HTTP stands for Hypertext Transfer Protocol.
- HTTP uses TCP (Transmission Control Protocol), generally over port 80.

What is HTTPS?

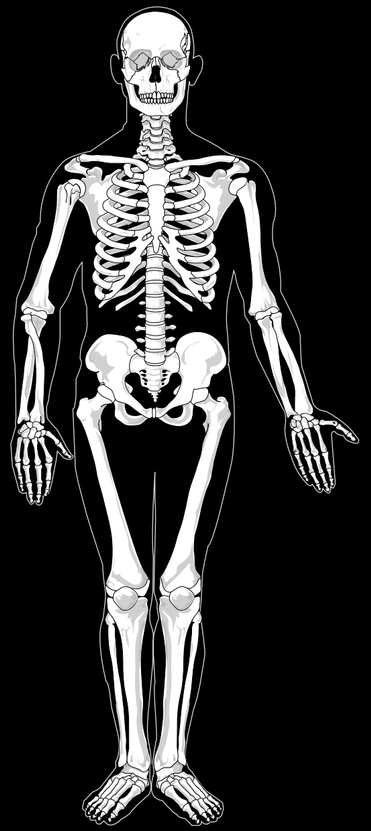
- HTTPS stands for Hypertext Transfer Protocol Secure (also referred to as HTTP over TLS or HTTP over SSL).
- HTTPS also uses TCP (Transmission Control Protocol) to send and receive data packets, but it does so over port 443.



What is the difference between HTTP and HTTPS?

- HTTP URL in your browser's address bar is http:// and the HTTPS URL is https://.
- HTTP is unsecured while HTTPS is secured.
- HTTP sends data over port 80 while HTTPS uses port 443.
- HTTP operates at application layer, while HTTPS operates at transport layer.
- No SSL certificates are required for HTTP, with HTTPS it is required that you have an SSL certificate and it is signed by a CA.
- HTTP doesn't require domain validation, where as HTTPS requires at least domain validation and certain certificates even require legal document validation.
- No encryption in HTTP, with HTTPS the data is encrypted before sending.

What is a "website"?

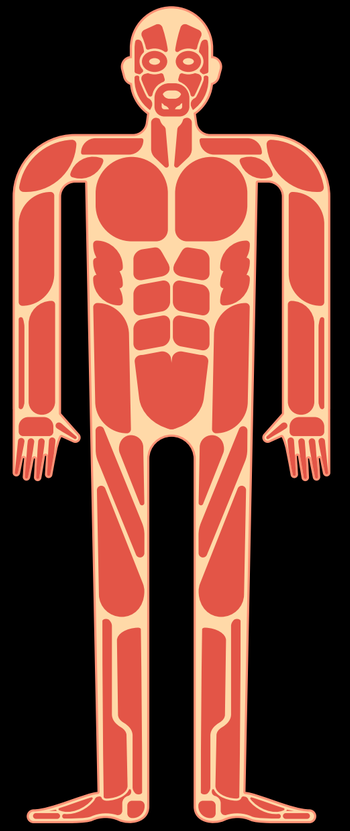


HTML

The Bones

CSS

The Skin



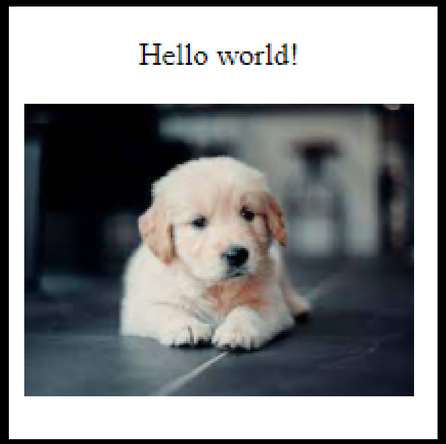
JAVASCRIPT

The Brains

HTML - The Bones

- It defines the layout of websites like images, buttons, and textboxes.
- It also defines where to load the javascript and css from

```
<body>
    <p>Hello world!</p>
    <img src="./dog.jpg">
    <script src="script.js"></script>
</body>
```



CSS - The Skin

- Defines what website elements should look like or position.
- Can be written in the HTML or loaded from external file.

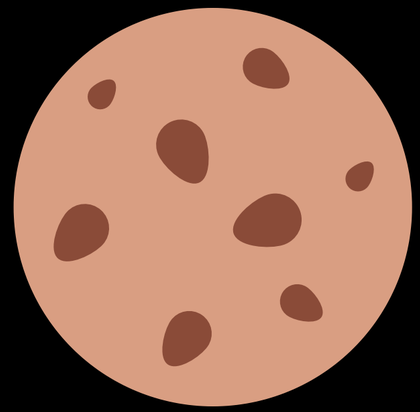
```
img {
    border-radius: 50%;
    filter: drop-shadow(0 0 0.75rem
black);
}
```

JavaScript - The Brains

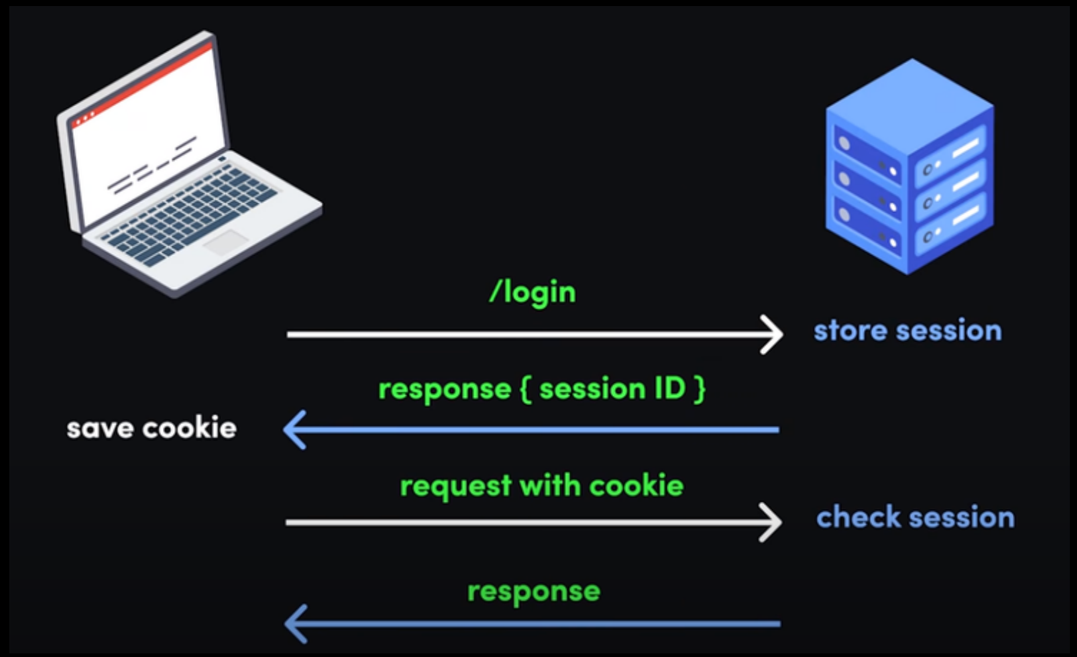
- Scripting language to make website do something
  - Do something when button is pressed
  - Animate things on webpage
  - Make requests to other endpoints

```
document.querySelector('img').addEventListener('click', () => {
    alert("! click on image !")
})
```

Cookies



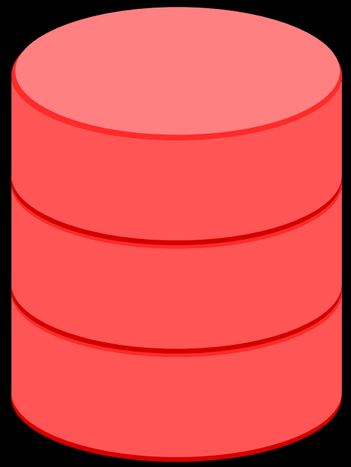
- Small pieces of information stored across visits to same web page.
- Maintained by browser, sent along with requests.
- Maintain a "session" after you log in to a site.



Local Storage

- Store key-value pairs like a cookie
- Not sent with requests to server
- Larger storage size limit (4KB vs 5MB)
- Can persist indefinitely

Database



- Organized collection of structured information, or data, typically stored electronically in a computer system
- Mainly two-types: No-SQL Database and SQL Database.

MySQL®

mongoDB

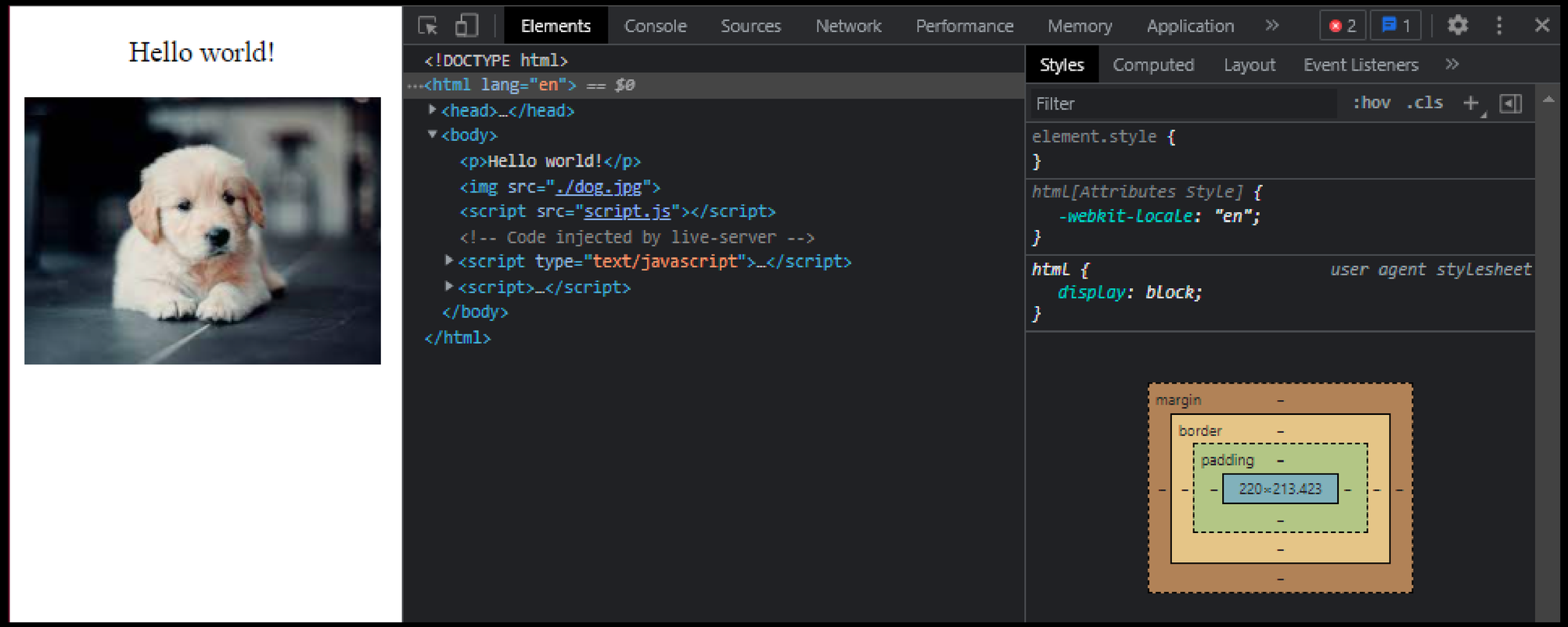
DevTools

- Inspect Element
- Console
- Network
- Application

Note: - It open via **F12, Ctrl + shift + I** etc

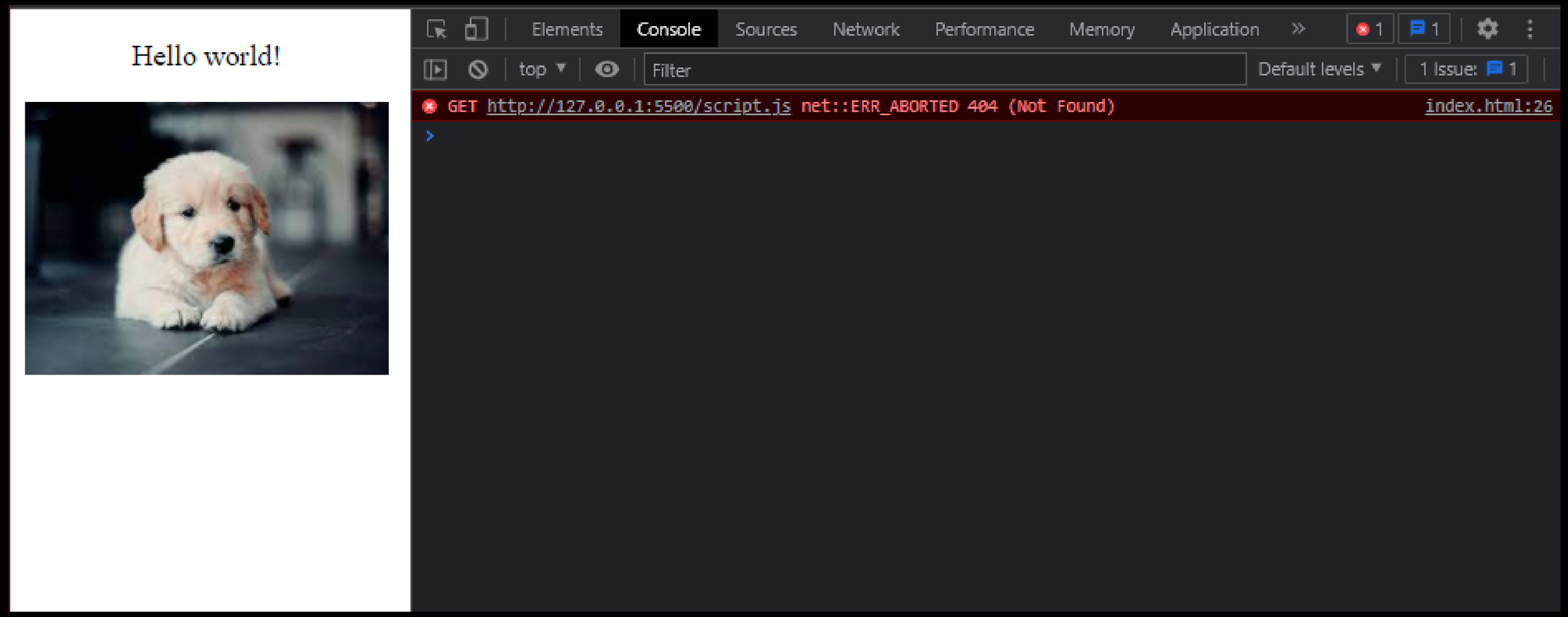
DevTools - Inspect Element

- Inspect HTML of page
- Delete or add elements
- View event listeners and styles (css property)



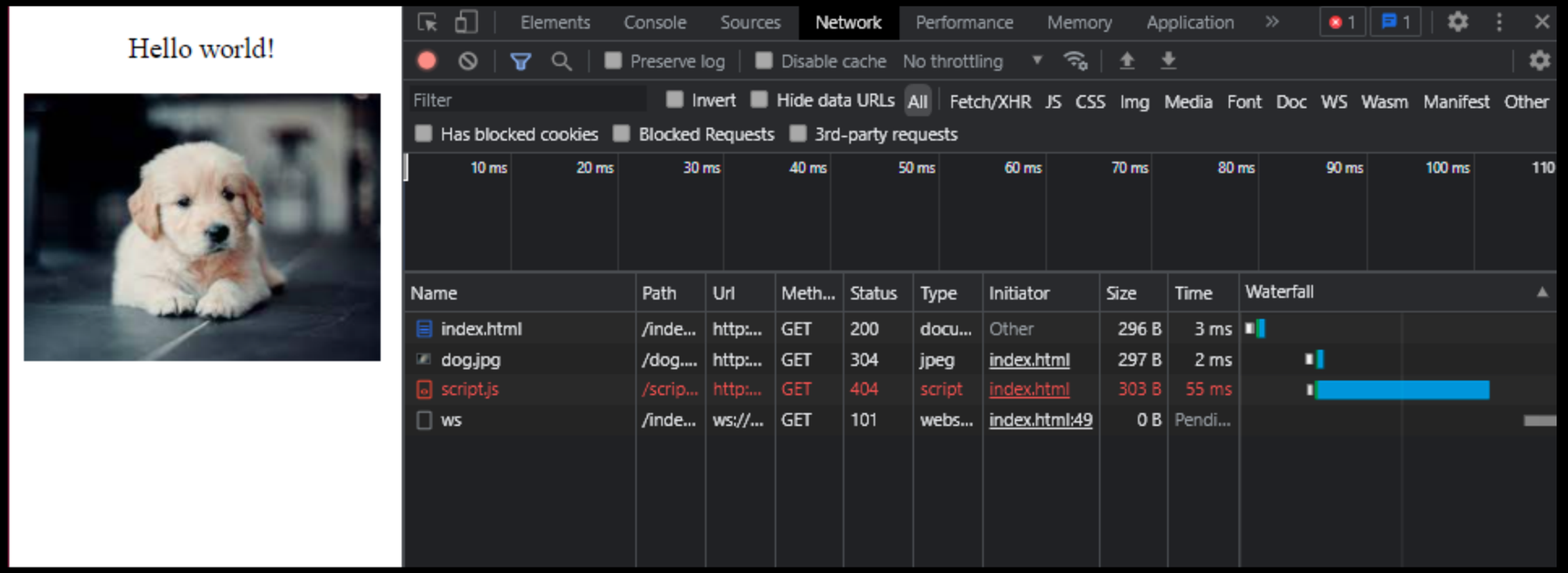
DevTools - Console

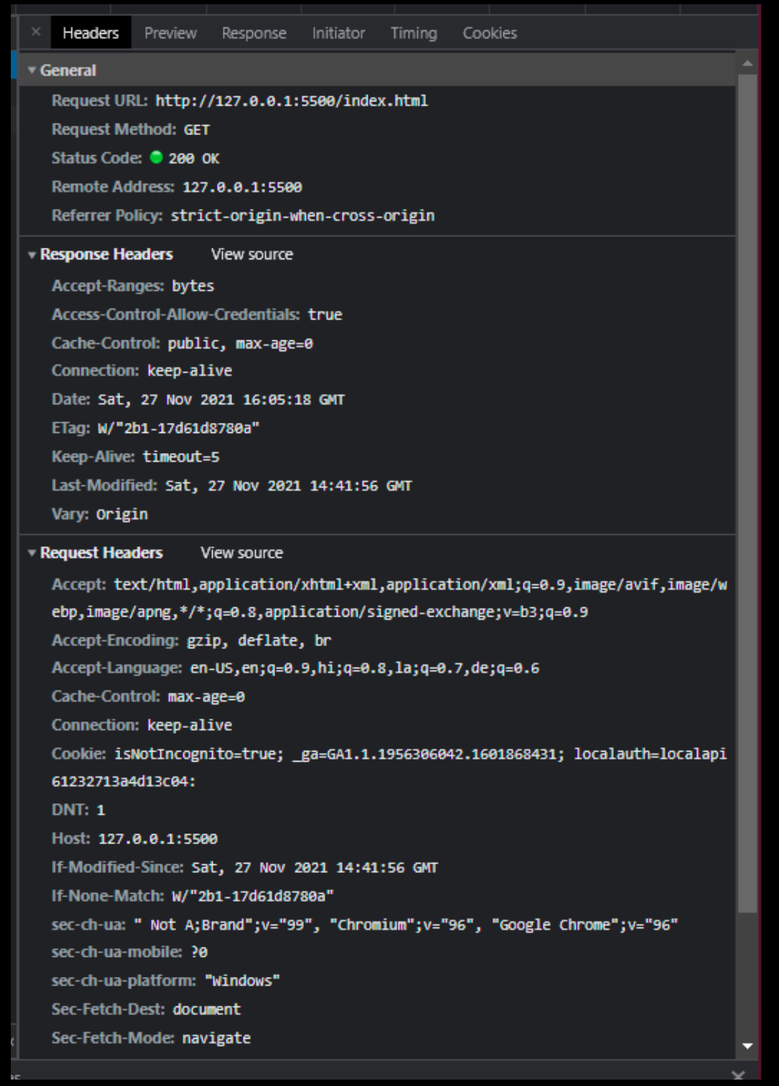
- View errors
- Execute your own javascript to interact with page and existing javascript



DevTools - Network

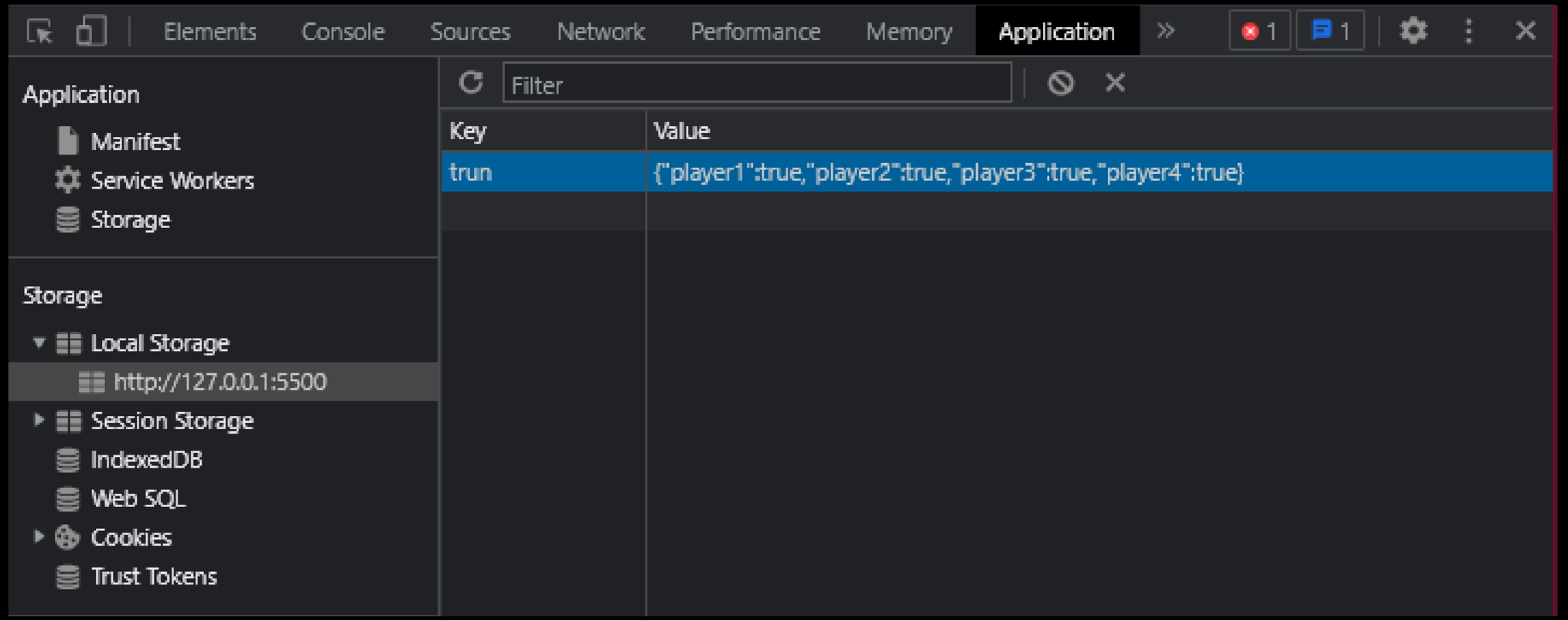
- View requests sent from your browser
  - Resources requested from server
  - Login forms
  - File uploads





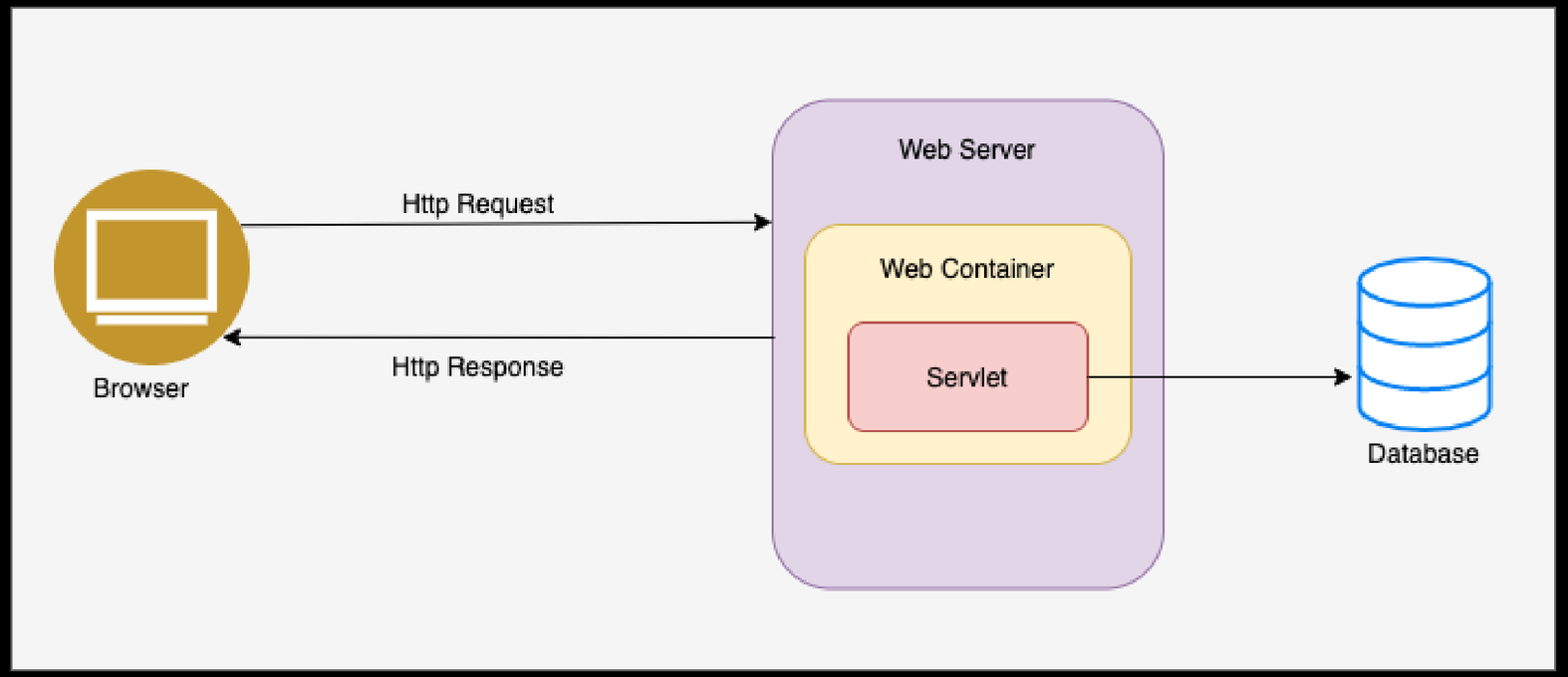
DevTools - Application

- View cookies and local storage for a website
- Modify contents to mess with web service



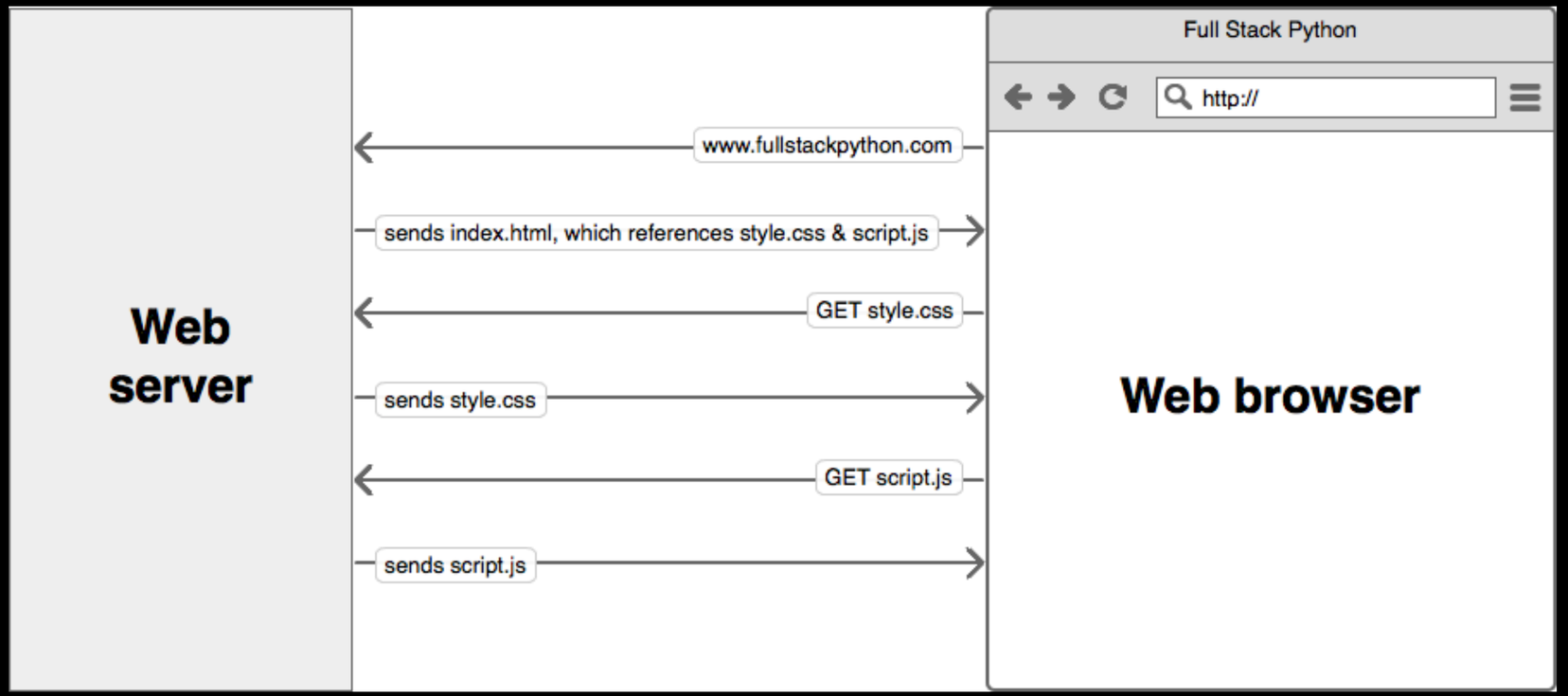
Requests

- A **web request** is a communicative message that is transmitted from **the client, or web browsers, to the servers**. A web request is also known as **HTTP request.**
- Mainly of four types: **GET, POST, DELETE and PUT**.



Responses

- An **HTTP response** is what is **sent by a server to a client** in response to an **HTTP request**.
- These responses contain a **status code** and if the request was successful, the client gets **the requested resource.**

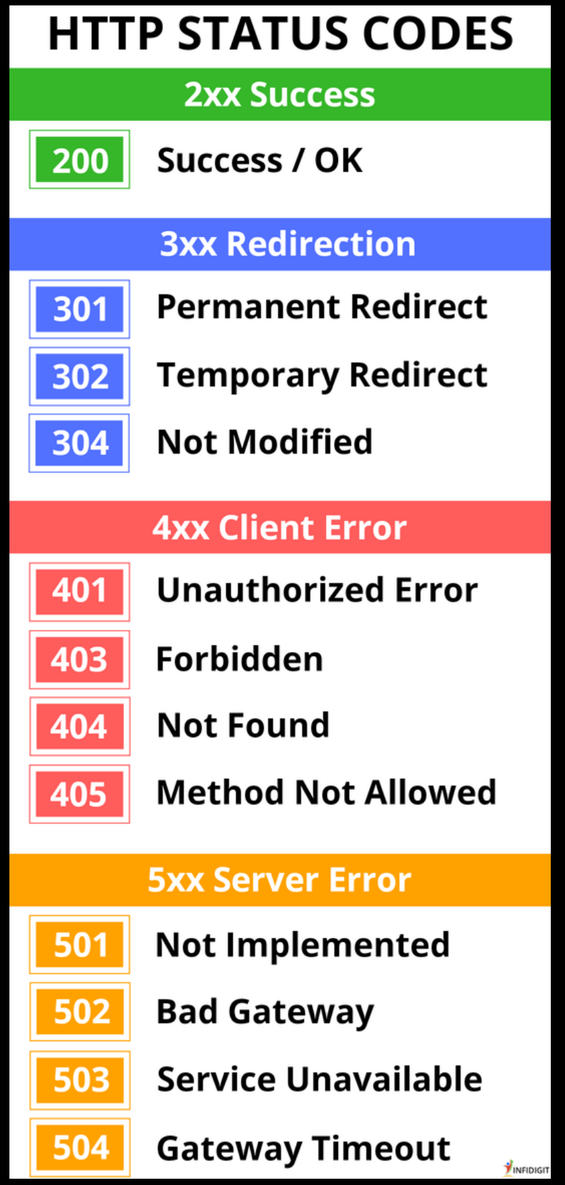


HTTP Status Code

- **1xxs – Informational responses**: The server is thinking through the request.
- **2xxs – Success**: The request was successfully completed and the server gave the browser the expected response.
- **3xxs – Redirection**: You got redirected somewhere else. The request was received, but there’s a redirect of some kind.
- **4xxs – Client errors**: Page not found. The site or page couldn’t be reached. The request was made, but the page isn’t valid – this is an error on the website’s side of the conversation and often appears when a page doesn’t exist on the site.
- **5xxs – Server errors**: Failure. A valid request was made by the client but the server failed to complete the request

Examples of Status Code

- **200**: The request succeeded. The result meaning of "success" depends on the HTTP method, such as "GET", "POST" etc.
- **301**: The URL of the requested resource has been changed permanently. The new URL is given in the response.
- **304**: This is used for caching purposes. It tells the client that the response has not been modified.
- **401**: The client must authenticate itself to get the requested response.
- **403**: The client does not have access rights to the content.
- **404**: The server can not find the requested resource. In the browser, this means the URL is not recognized.
- **502**: The server, while working as a gateway to get a response needed to handle the request, got an invalid response.
- **504**: This error response is given when the server is acting as a gateway and cannot get a response in time.



THANK YOU

May the source be with you !