

# Asynchronicity And Networking

Introduction

Created by Zain Afzal



@zainafzal08



@blockzain

# Overview

Client server Interactions

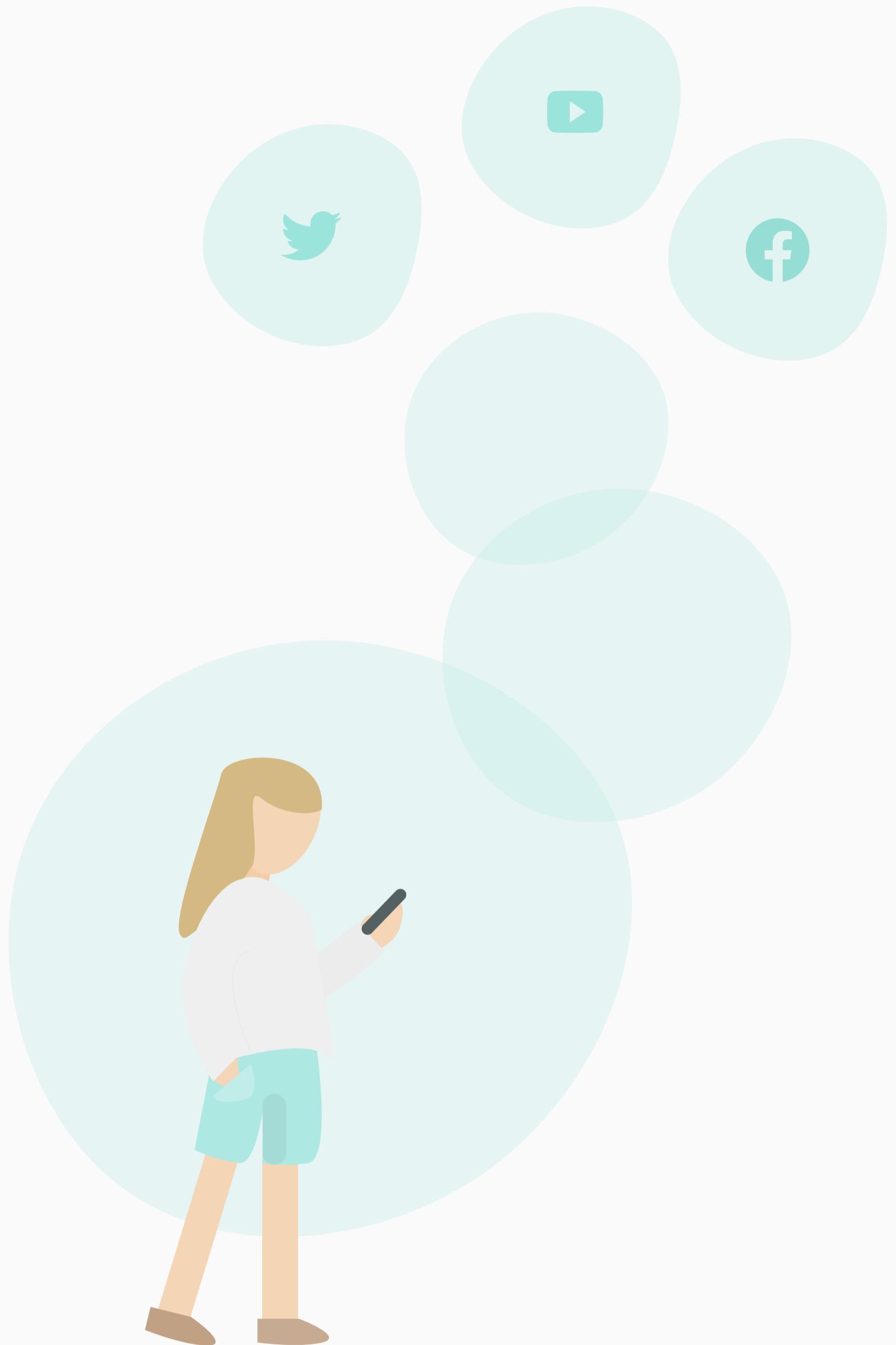
AJAX

Event Loop

Callbacks and XMLHTTP

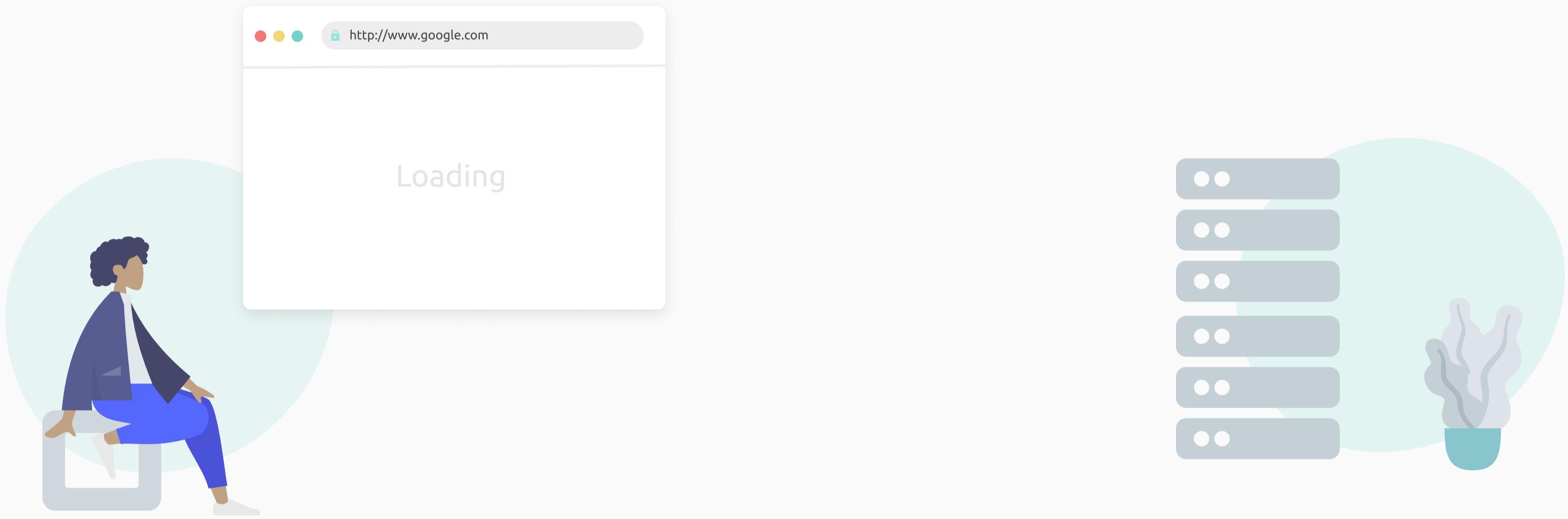
Promises and Fetch

Async Await



# Client server Interactions

Slide 1 of 8



## Client

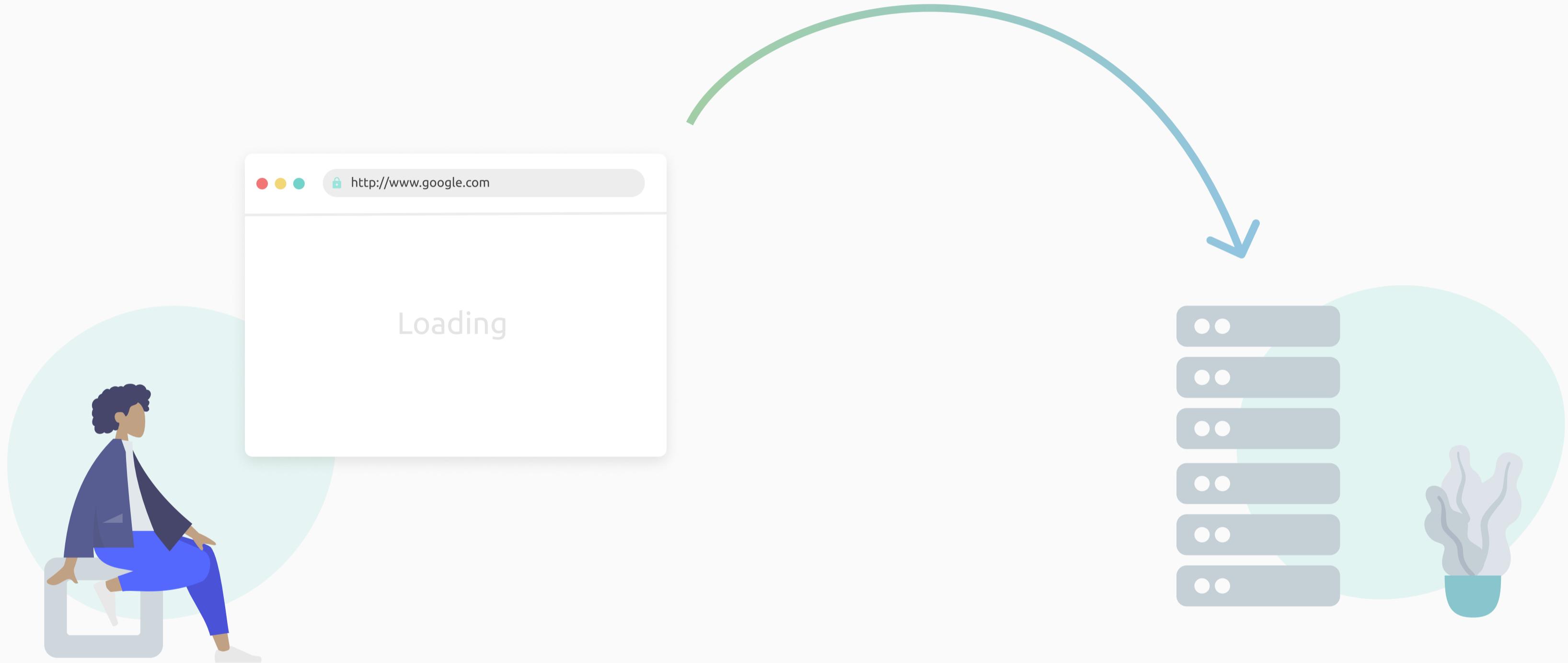
Your home PC, laptop, phone, washing machine etc.

## Server

A dedicated machine run by a website to receive and process requests

# Client server Interactions

Slide 2 of 8



## Client

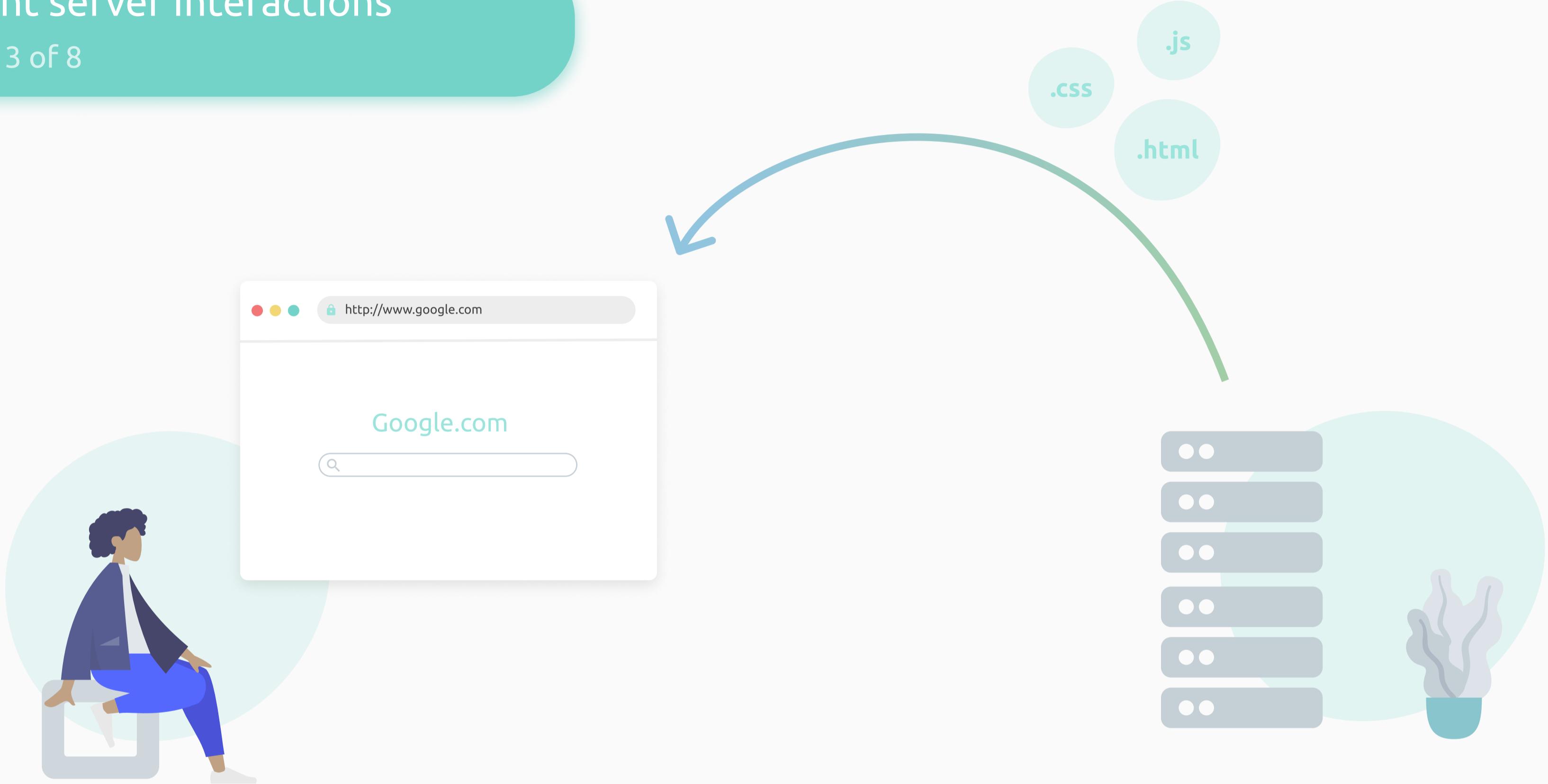
Sends a request for <http://www.google.com>

## Server

Process the request and answers some key questions such as, "can this user see this website" and "what exact html needs to be send back"

# Client server Interactions

Slide 3 of 8



## Client

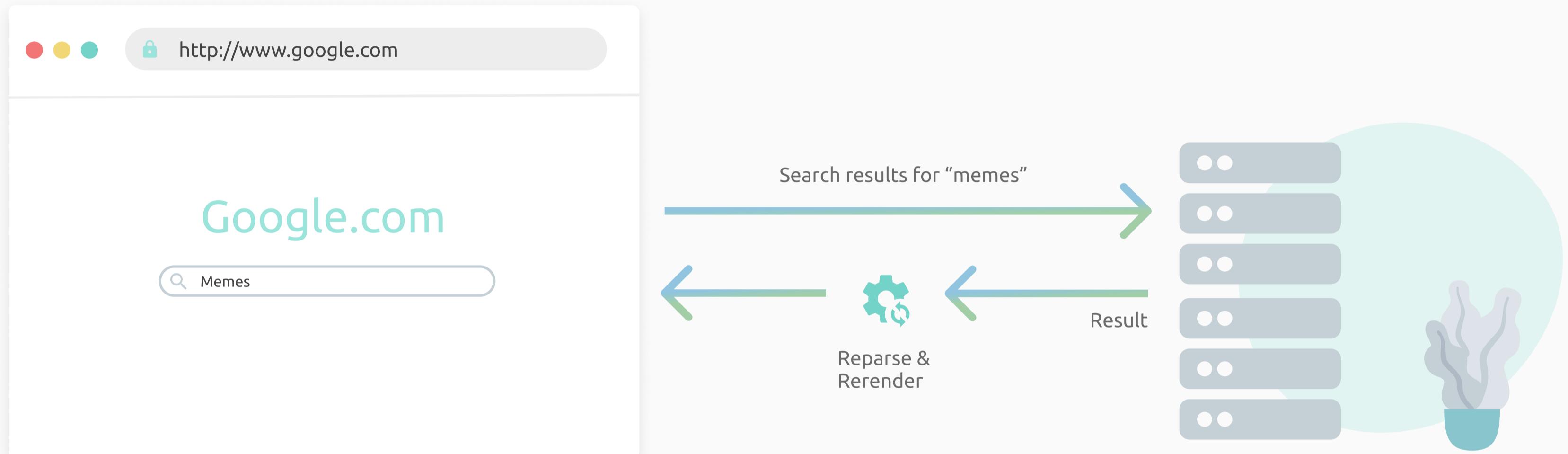
Receives the payload of data, parses it, compiles it and runs it producing the web page.

## Server

Decides what to send back, creates a response of HTML, CSS and JS and shoots it off. This is called **Server Side Rendering**

# Client server Interactions

Slide 4 of 8



Client

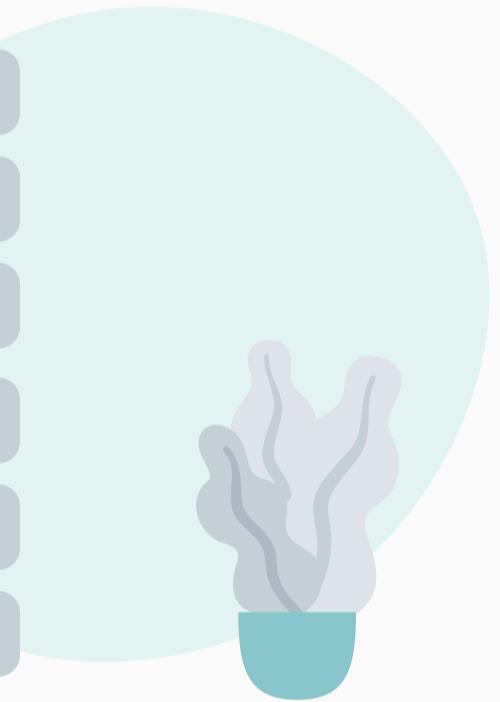
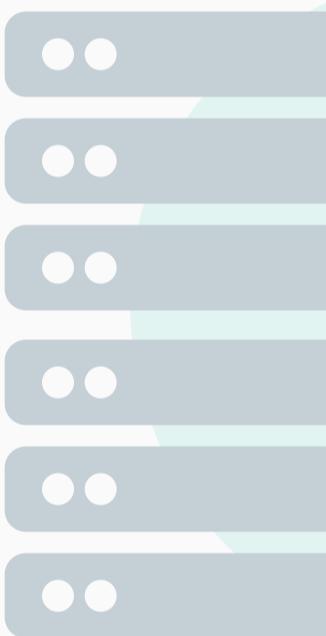
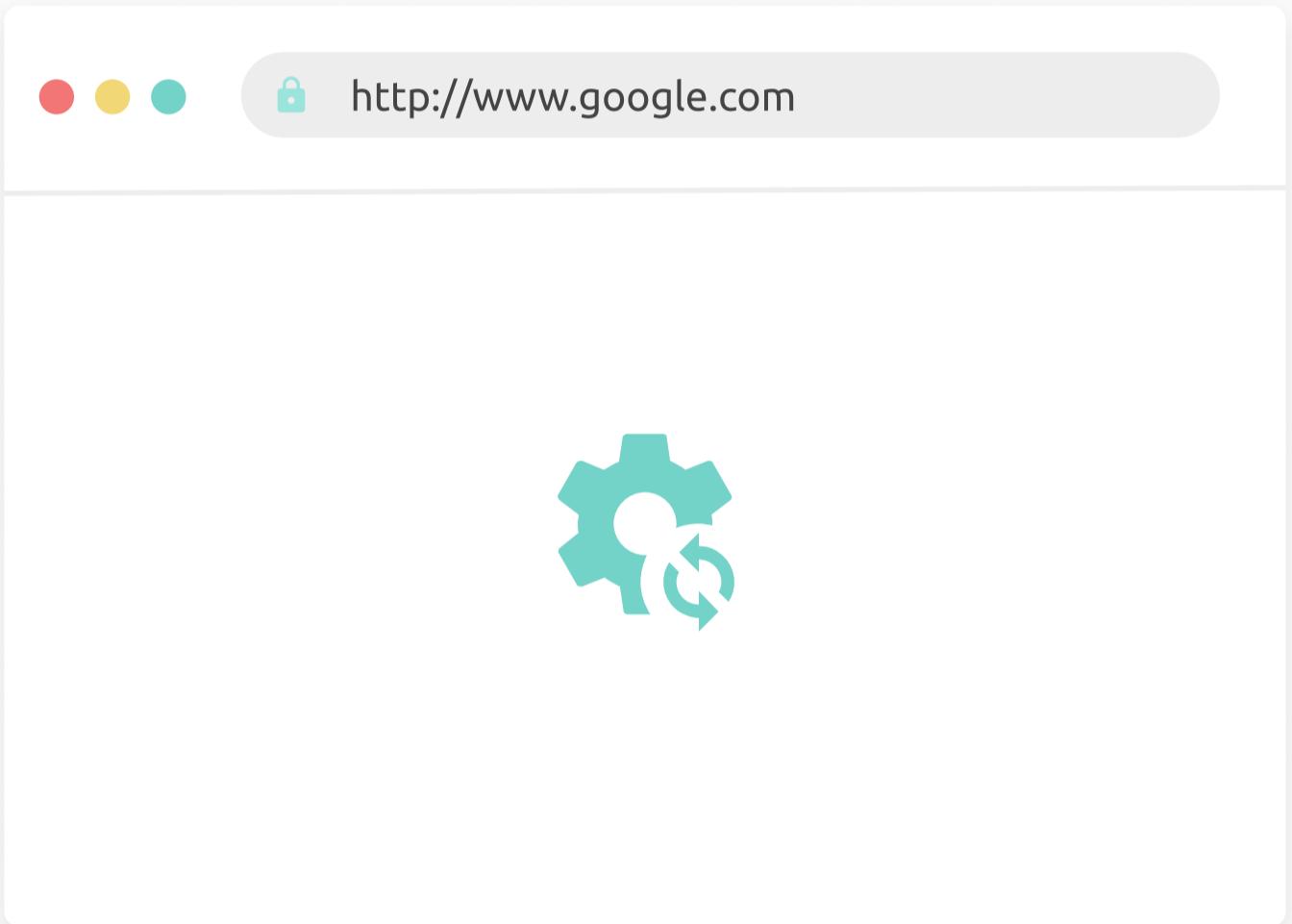
Forms and interactivity were built on top of this, a request with some information is sent and the response triggers a reparse and rerender

Server

Needs to spend time and resources forming a full page response and sending it over

# Client server Interactions

Slide 5 of 8



## Client

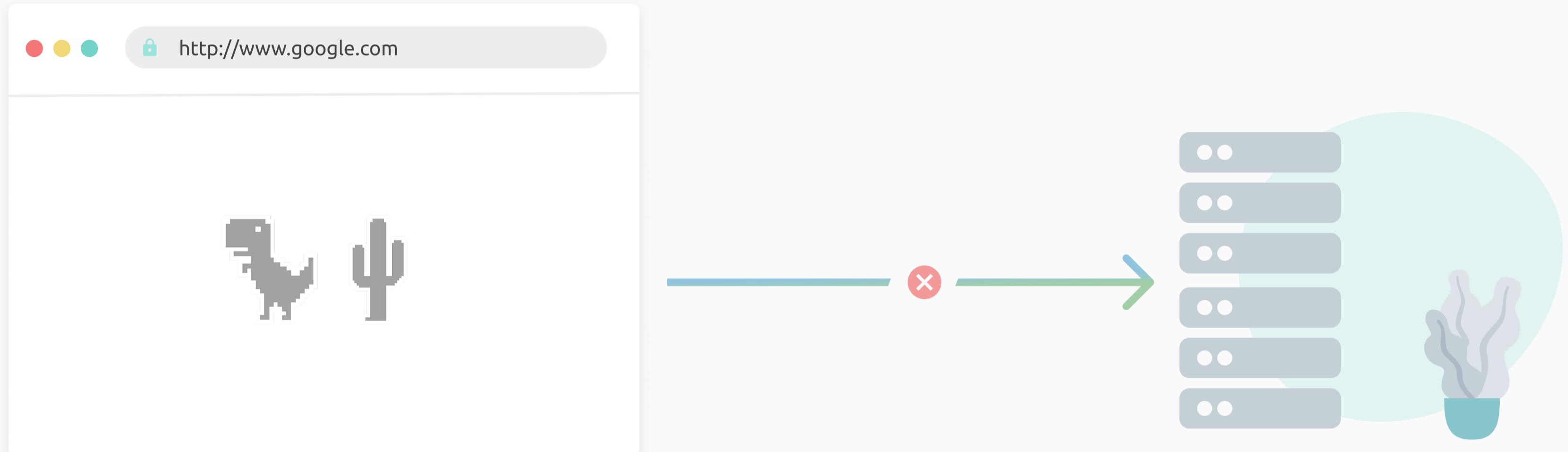
It takes time to do all the steps to render a page to screen, initial load time can easily be in the seconds.

## Server

Needs to spend time and resources forming a full page response and sending it over

# Client server Interactions

Slide 6 of 8



**Client**

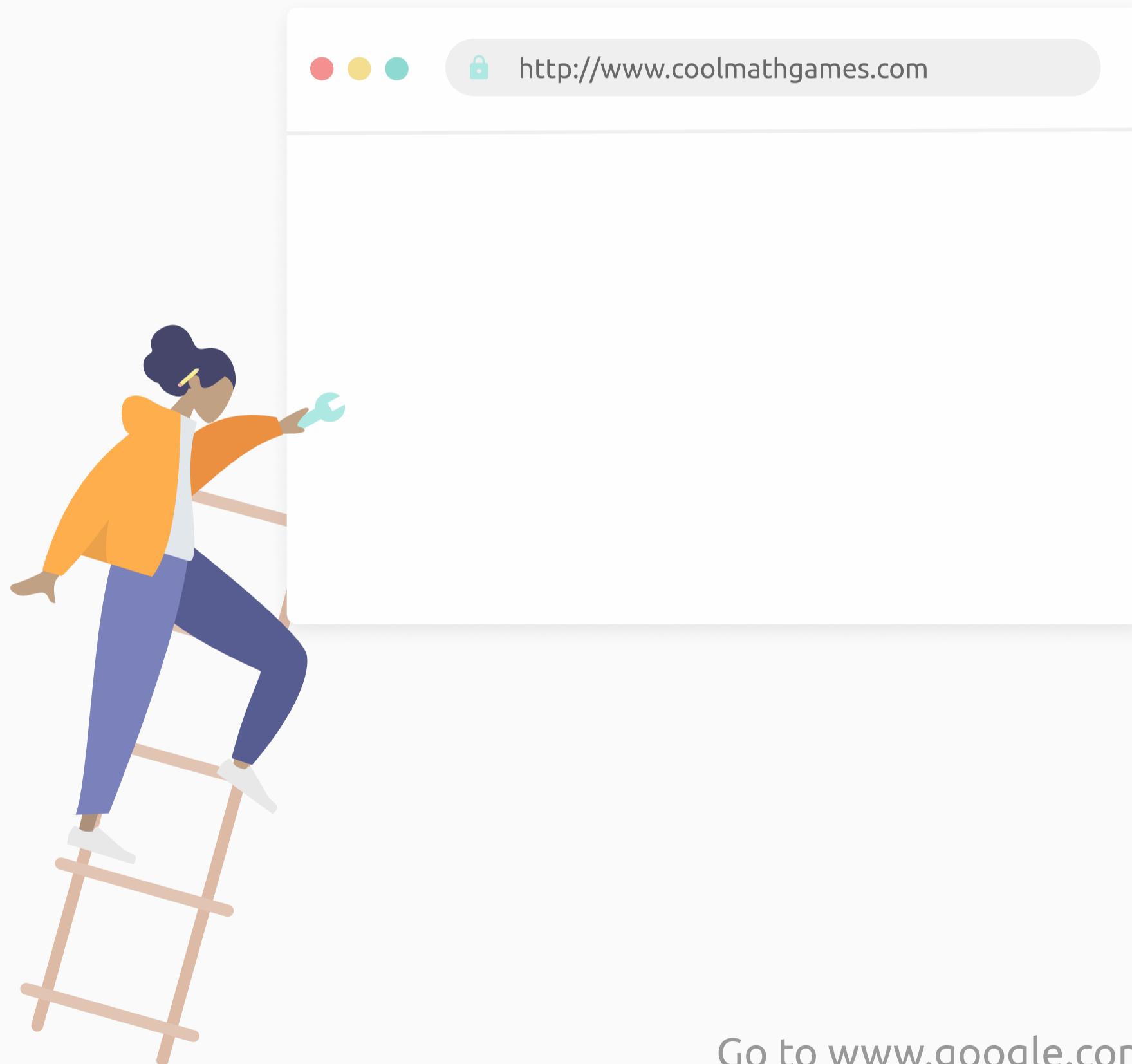
If the request fails the client can't do anything but show a built in error page.

**Server**

The website can't interact with the user at all

# Client server Interactions

Slide 7 of 8

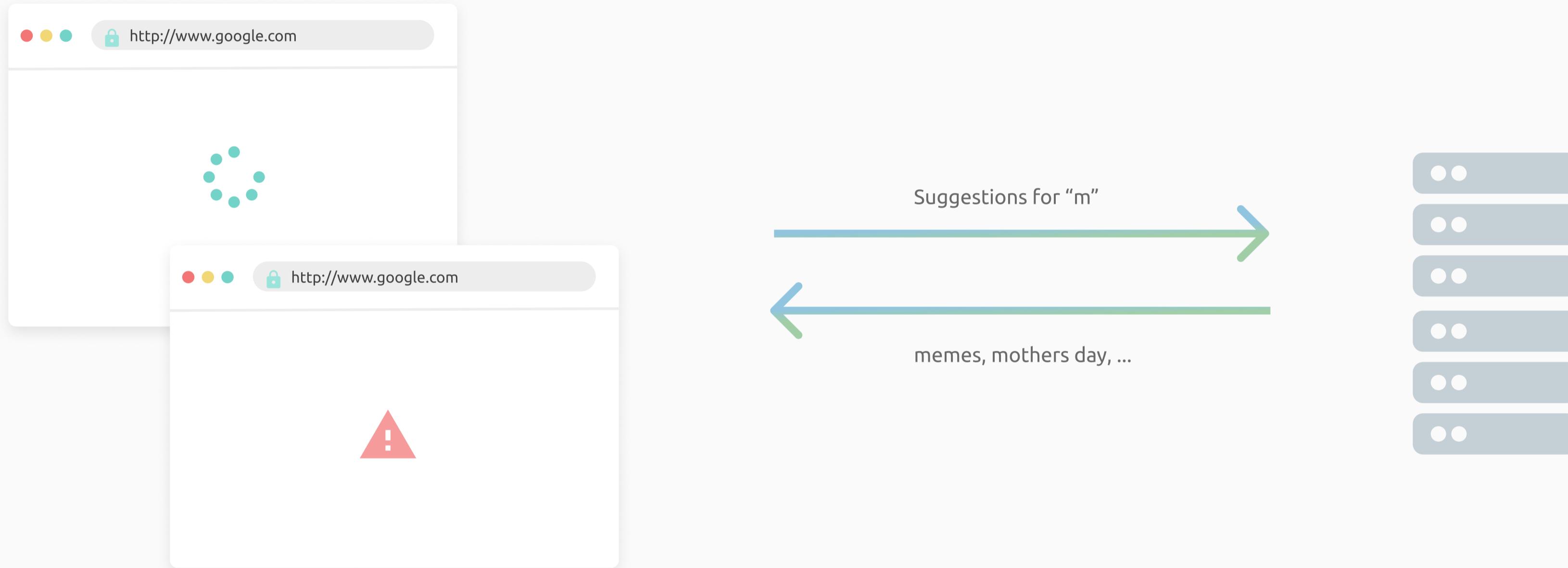


 Demo

Go to [www.google.com](http://www.google.com) and type something in, notice how the suggestions pop up without a full page reload

# Client server Interactions

Slide 8 of 8



## Client

Chooses what data to request and can gracefully handle both the loading and fail cases. When the response comes can update the screen without a full reload.

## Server

Can produce the smallest and quickest response with exactly what the client needs.

# A synchronous Javascript And XML

Smoothens UI

Improves Speed

Allows for offline web apps

Separates frontend and backend responsibilities

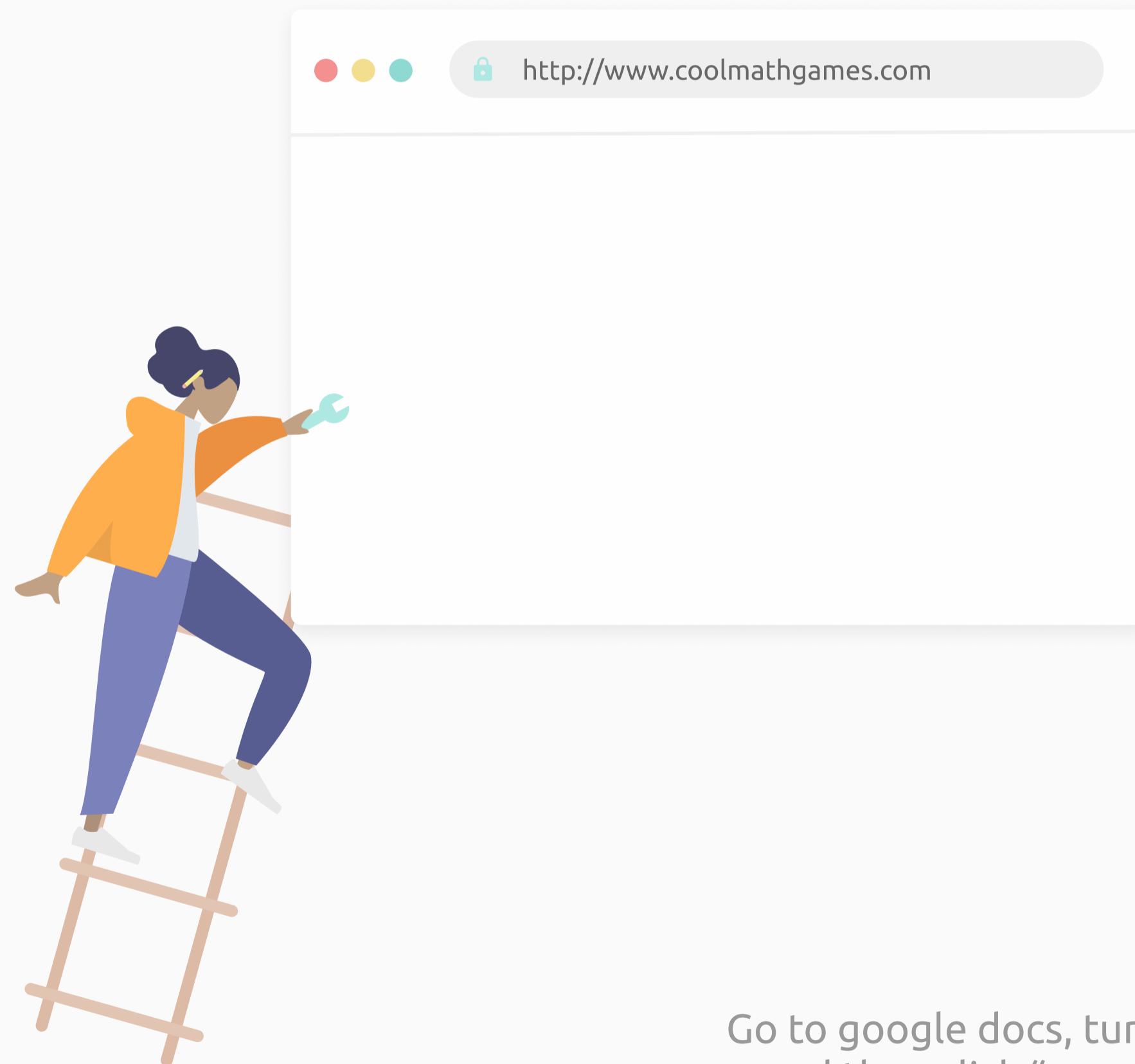
## Ajax

A development technique in which a frontend client sends and receives data to the backend asynchronously (in the background) without interfering with the display and behavior of the existing page.



## Client Side vs Server Side Rendering

Client side rendering is when the server doesn't provide the final html which is rendered but rather a set of javascript which generates the dom to render. This is most commonly used in conjunction with ajax so the client can fully control what is displayed. This is unlike server side rendering where the server has full control (think python flask).



## Demo

Go to google docs, turn off the internet (network tab) and then click “new file” notice how the whole page changes without a single request to the server