Google | طويق

Introduction to Full Stack Web Development

أكاديمية طويق
TUWAIQ ACADEMY

وزارة الاتصالات وتقنية المعلومات
MINISTRY OF COMMUNICATIONS AND INFORMATION TECHNOLOGY

الاتحاد السعودي للأمن السيبراني والبرمجة والدرونز
SAUDI FEDERATION FOR CYBERSECURITY, PROGRAMMING & DRONES

مهارات المستقبل

stc

Objective

- Full Stack Web Development Concept
- Web Developer Responsibilities
- Crucial Knowledge for Back-End Development
- Crucial Knowledge for Front-End Development

Full Stack Web Development

- A web developer should create custom code for a client's unique needs
- Web development has 3 parts:
  1. Client-side scripting: executes in a web browser and determines what your customers or clients will see on your website
  2. Server-side scripting: executes on a web server and powers the behind-the-scenes mechanics of how a website works
  3. Database technology: stores and manages all data needed for your web application (this will come in level 3)

Web Developer Responsibilities

Front-End Developer:

- Prioritising user experience
- Bringing a designer's concept to life
- Production and maintenance of websites and web app user interfaces
- Creating tools that enhance user interaction in any browser
- Implementing responsive design for mobile sites
- Using a project management tool like GitHub to maintain software workflow management
- Looking at search engine optimisation (SEO) best practices
- Testing the site for usability and fixing bugs during development

Web Developer Responsibilities

Back-End Developer:

- Database creation, integration and management
- Using back-end frameworks (e.g.Express.js) to build server-side software
- Web server technologies
- Cloud computing integration
- Server-side programming languages
- API integration
- Content management system development, deployment and maintenance
- Security settings and hack prevents
- Reporting, generating analytics and statistics
- Backup and restore technologies for a website’s

Web Developer Responsibilities

Full Stack Developer:

- Server, Network and Hosting Environments
- Data modelling
- Business Logic
- User Interface
- User experience
- Customer and business needs

Crucial Knowledge for Back-End Development

Web apps are usually backed up by 2 types of servers:

- Web servers
- Application servers

Web servers: computer with an internet connection and software to make web pages available to clients.

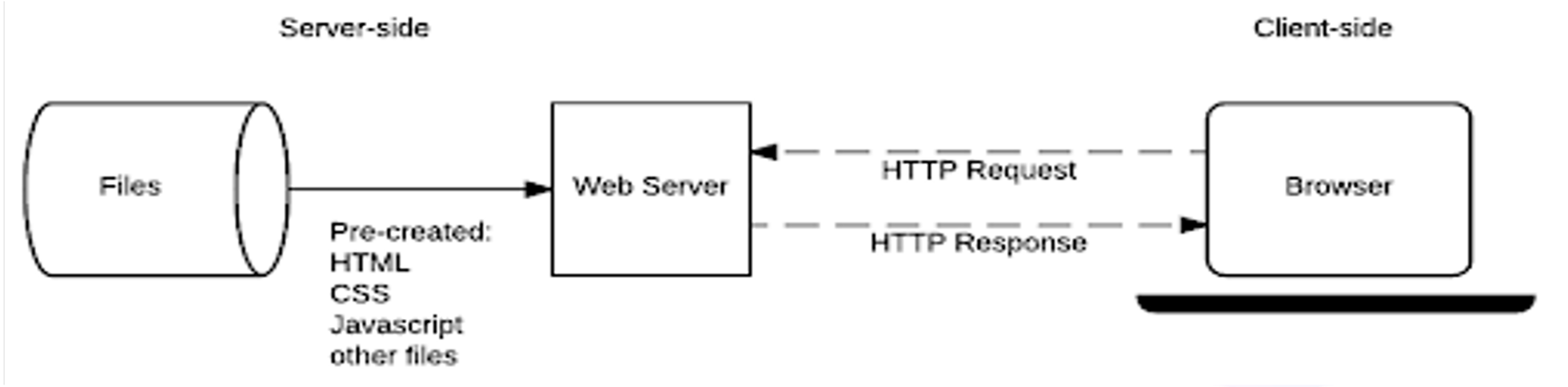
- Stores resources for web apps, e.g. images, HTML, CSS and JS files
- Must have software that allows it to act as an HTTP server

Crucial Knowledge for Back-End Development

Application servers: contain the business logic needed to build resources to be passed back to the browser.

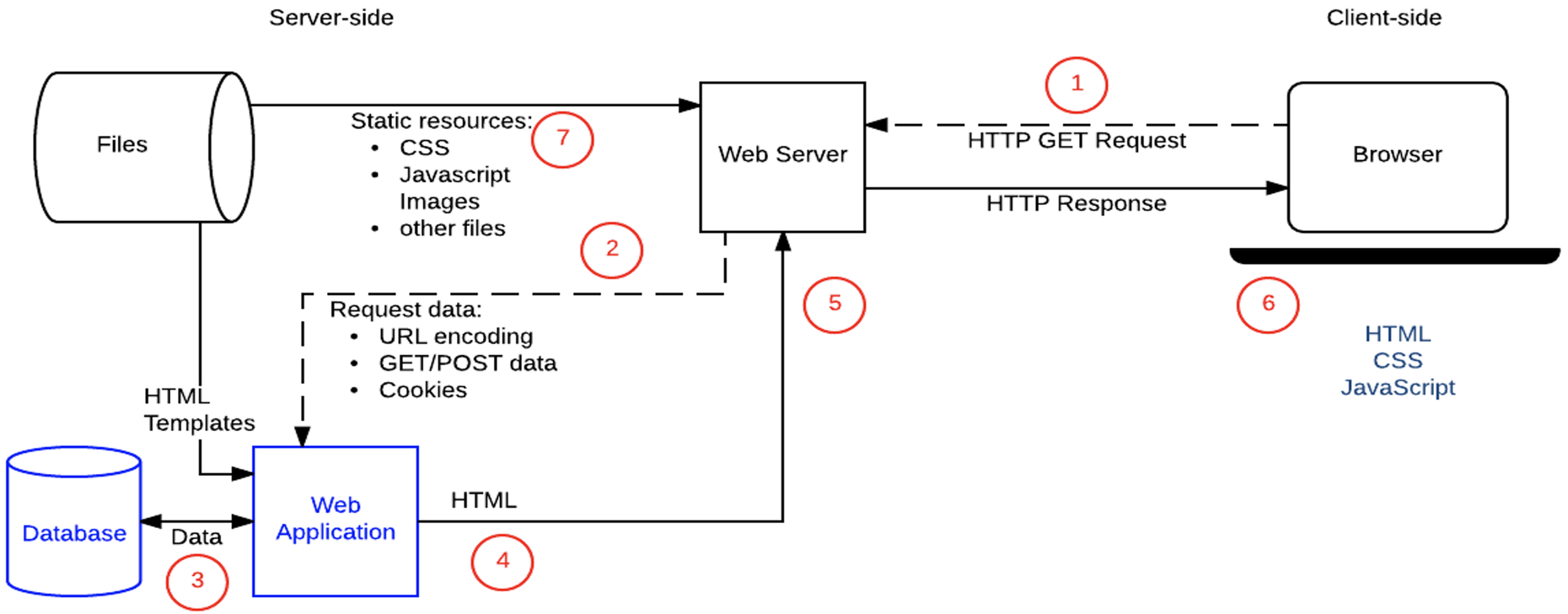
- Usually have a web framework running on it
- Framework is able to match requests and dynamically generate responses.

Crucial Knowledge for Back-End Development



Web Server Architecture for a Static Site (developer.mozilla.org)

Crucial Knowledge for Back-End Development

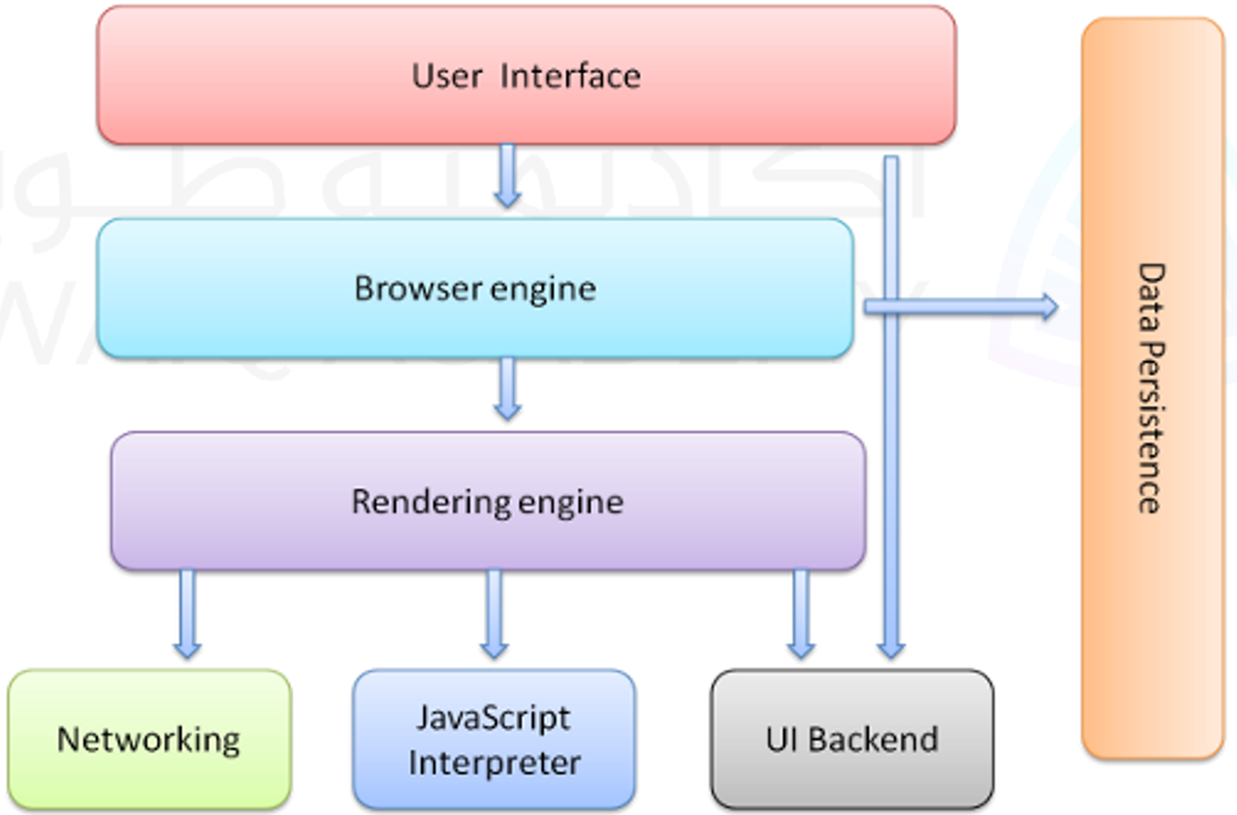


Web Server Architecture for a Dynamic Site (developer.mozilla.org)

Google | طويق

Crucial Knowledge for Back-End Development

How does a browser actually work?

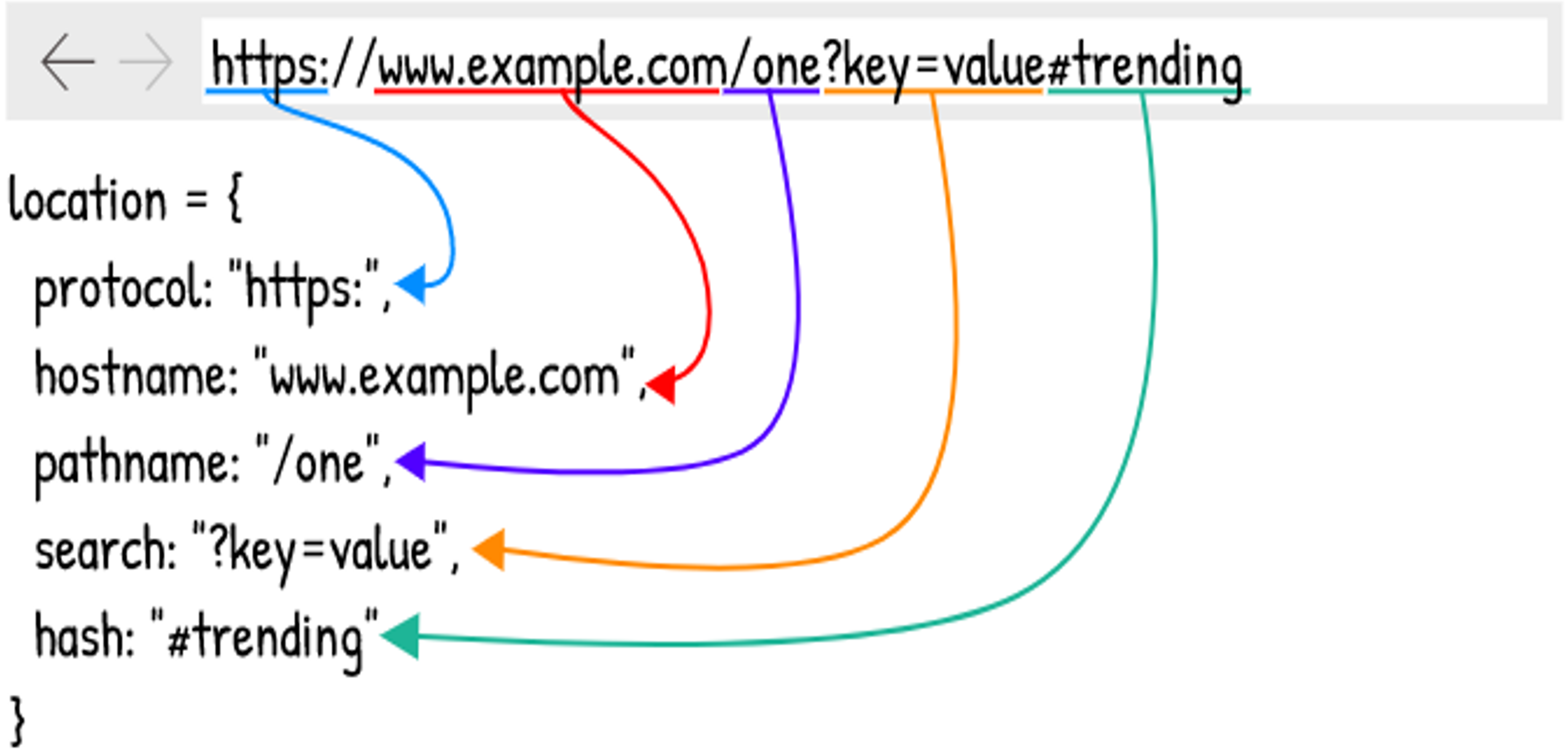


Crucial Knowledge for Back-End Development

Rendering engine does the following to render the page:

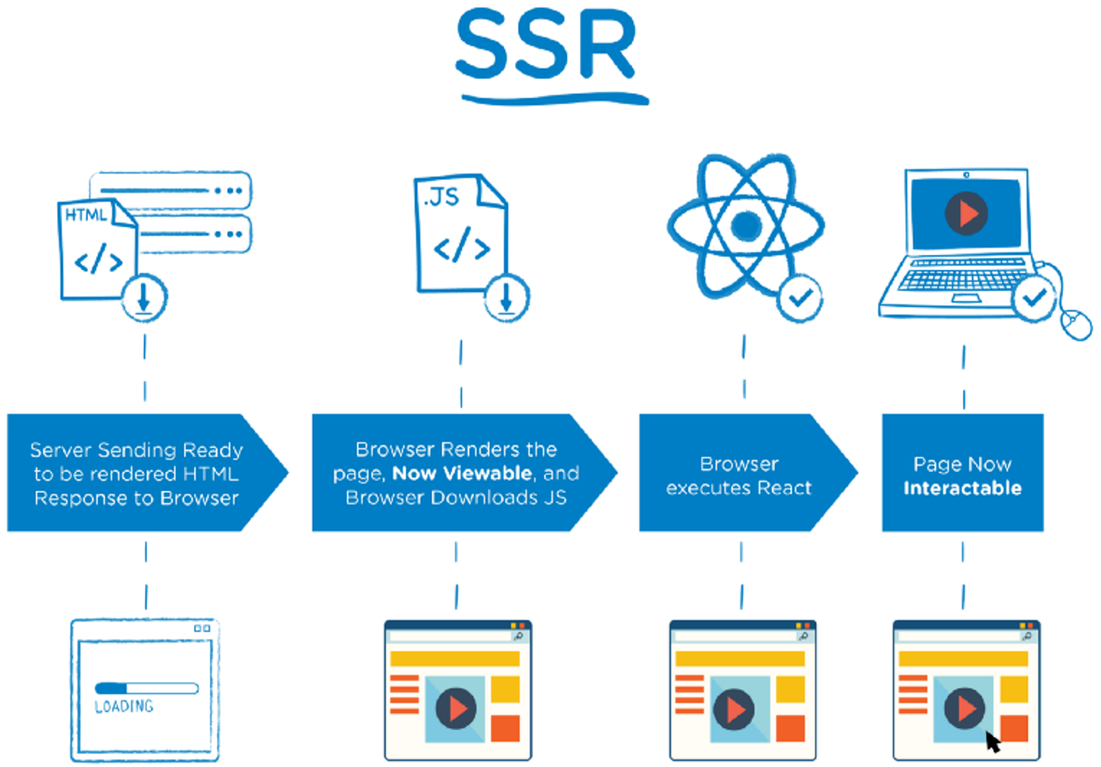
Crucial Knowledge for Back-End Development

Single Page Application (SPA): "an application that loads a single HTML page and all the necessary assets (such as JavaScript and CSS) required for the application to run. Any interactions with the page or subsequent pages do not require a round trip to the server which means the page is not reloaded." (by React).



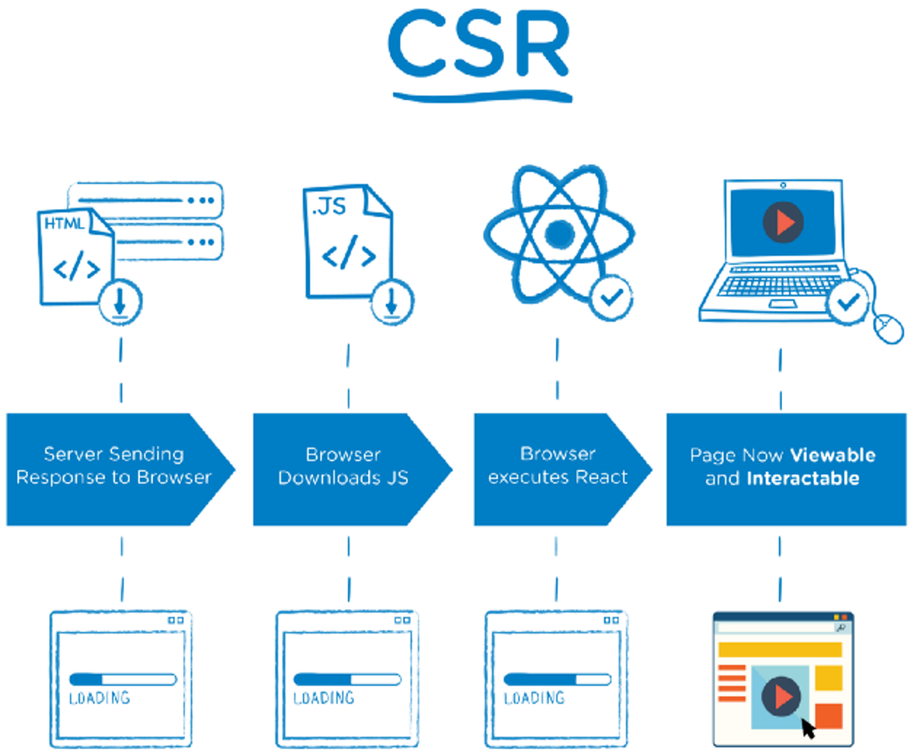
Crucial Knowledge for Back-End Development

Server Side Rendering (SSR): taking a JavaScript framework (like React) and rendering it to static HTML and CSS on the server.



Crucial Knowledge for Back-End Development

Client Side Rendering (CSR): your browser will start rendering the HTML from your server without having to wait for all the JavaScript to be downloaded and executed.



When to use server-side rendering

- An application has very simple UI with fewer pages/features
- An application has less dynamic data
- Read preference of the site is more than write
- The focus is not on rich site and has few users

When to use client-side rendering

- An application has very complex UI with many pages/features
- An application has large and dynamic data
- Write preference of the site is more than reading
- The focus is on rich site and a huge number of users

Resources

- How Browsers Work: Behind the scenes of modern web browsers
- How Browsers Work: Behind the scenes of modern web browsers

- *How Single-Page Applications Work*
- *The Benefits of Server Side Rendering Over Client Side Rendering*

Summary

- Full Stack Web Development Concept
- Web Developer Responsibilities
- Crucial Knowledge for Back-End Development
- Crucial Knowledge for Front-End Development