

## Why is accessibility important?

By making your website accessible, you are **ensuring that all of your potential users, including people with disabilities, have a decent user experience and are able to easily access your information.** By implementing accessibility best practices, you are also improving the usability of the site for all users.

## How do we test for accessibility?

1. Check alt text for images and other non-text content
2. Check for closed captions and transcripts on videos
3. Check color contrast
4. Make sure your site is keyboard-friendly
5. Make sure your site can be zoomed without loss of content or functionality

## What is semantic HTML?

**HTML that introduces meaning to the web page rather than just presentation.** That describes its meaning to both the browser and the developer.

## What is DOM Manipulation?

**Interacting with the DOM API to change/modify the HTML document that is to be rendered on the web browser.**

## Why should we always use eventListeners in JavaScript and not inline onClick handlers in the HTML?

OnClick is limited to one inline event listener, if more were added it will be overwritten. AddEventListener can have multiple event handlers applied to the same element. It doesn't overwrite other present event handlers.

## Why do we use github/git?

it is a version control system that lets you manage and keep track of your source code history. GitHub is a cloud-based hosting service that **lets you manage Git repositories**. And it helps developers and programmers to collaboratively work on code

## git clone?

if you want to get a copy of an existing Git repository — for example, a project you'd like to contribute to — the command you need is `git clone`. Git receives a full copy of nearly all data that the server has. Every version of every file for the history of the project is pulled down by default when you run `git clone`

## fork?

Creating a "fork" is producing a personal copy of someone else's project. Forks **act as a sort of bridge between the original repository and your personal copy**. Which allows **you** to freely experiment with changes without affecting the original project

## Git pull origin master ?

**fetch and update only a specific branch called master and origin in the remote repository**

## How do we create a new branch on our local machine?

`Git checkout -b <branch-name>`

## When we make a change to a file, how do we tell git to track it?

`Git add <filename>`

## **What does git push origin [branch-name] do?**

It's **used for pushing local content to GitHub**

## **Why do we make pull requests instead of just changing master directly?**

Pull requests **provide for checks and balances**, even if anyone can push to master. The biggest advantage is that they provide an opportunity for code review

## **Why is it important to run our team member's branches when they make a pull request?**

to **help manage the risk in the services and make sure that the application runs as it should be.**

## **When we have finished working on a branch, how do we make sure that our changes do not cause a conflict with master? (this can all be done locally)?**

By making sure that we have the latest version of the project. And by pulling the latest updates on the branch that we are working on before submitting the PR.

## **What does the Status Code of an HTTP response tell us?**

HTTP response status codes indicate whether a specific HTTP request has been successfully completed.

## **What are some common Status codes?**

**HTTP Status Code 200 – OK**

**HTTP Status Code 301 - Permanent Redirect**

**HTTP Status Code 302 - Temporary Redirect**

**HTTP Status Code 400 – Bad Request**

**HTTP Status Code 404 - Not Found**

**HTTP Status Code 500 - Internal Server Error**

**HTTP Status Code 503 - Service Unavailable**

## **What are HTTP methods and what are the different methods intended for?**

The primary or most-commonly-used HTTP verbs (or methods, as they are properly called) are **POST, GET, PUT, PATCH, and DELETE**. These correspond to create, read, update, and delete (or CRUD) operations, respectively.

## **Difference between toBe and toEqual?**

toBe is used to compare primitive values or to check referential identity of object instances. On the other hand, toEqual checks for deep equality.

## **How do you initialise an npm or node project?**

Run npm init in terminal

## **What is the only script that an npm project starts with, how do you add one?**

“test”, you can add your own custom scripts by adding key value under scripts in package.json.

## **What is the difference between dependencies and devDependencies?**

DevDependencies are modules which are only required during development, while dependencies are modules which are also required at runtime.

## What is the difference between `querystring.parse` and `url.parse`?

`querystring.stringify({search: 'javascript'})`, returns key value pairs of the search query, in this case `search=javascript`, while `url.parse('https://www.codewars.com/users/m4v15/completed')` returns an object with the url's components

## What file path should I use to require the `db_connect` file if I am in `app.js`?

```
`./src/database/db_connect`
```

## What is the file path to `index.html` from `app.js`

```
`./public/index.html`
```

## Why is linting important? Why should all of your code have the same style?

To save yourself trouble by avoiding common problems and to enforce a common code style across a team

## Why is it important to use asynchronous functions in Node?

The `async` function helps to write promise-based code asynchronously via the event-loop

## What are error-first callbacks, and why is it important to follow that pattern in your own code?

**Error-First Callback** in Node.js is a function which either returns an error object or any successful data returned by the function. Because if any error has occurred during the execution of the function, it will be returned by the first argument.

## What is database schema ?

Schema **defines how data is organized within a relational database**; this is inclusive of logical constraints such as, table names, fields, data types, and the relationships between these entities

## Why we use different database urls for production, development and testing?

Each database has it's own purposes, and to not risk to remove the data in the database

## How would you use the .query method to query the database?

```
Connection.query(`SQL COMMAND`)
```

## What are parameterized queries?

**a query in which placeholders are used for parameters and the parameter values are supplied at execution time.** The most important reason to use parameterized queries is to avoid SQL injection attacks

## What is express middleware?

Middleware functions have access to the request object and the response object and also the next function in the application request-response lifecycle.

## How to hash and compare using bcrypt?

```
Bcrypt.hash(password, 10)
```

```
Bcrypt.compare(password, incomingPassword)
```

## How to save and read a cookie on the server?

```
Res.cookie('key', 'value', options)
```

## **How to write a JWT on the server?**

`JWT.sign(payload, secretkey)`

## **why is a JWT secure?**

The signing algorithm takes the header, the payload, and the secret to create a unique signature. So only this data plus the secret can create this signature, all right? Then together with the header and the payload, these signature forms the JWT

## **How to secure a route and allow people with a valid jwt through?**

Adding a middleware that verify through jwt token using `JWT.verify`