

WORKING WITH NODE

CREATE A NODE SERVER:

- **require('http'):** Imports the built-in `http` module.
`const http = require('http');`
- **hostname and port:** Defines the hostname (localhost) and port number (3000) for the server.
`const hostname = '127.0.0.1';`
`const port = 3000;`
- **http.createServer:** Creates an HTTP server. The callback function handles incoming requests.
 - `req` object: Represents the incoming request.
 - `res` object: Represents the outgoing response.
 - `res.statusCode = 200;` Sets the HTTP status code to 200 (OK).
 - `res.setHeader('Content-Type', 'text/plain');` Sets the content type of the response to plain text.
 - `res.end('Hello, World!\n');` Sends the "Hello, World!" message as the response body and ends the response.

```
const server = http.createServer((req, res) => {  
  res.statusCode = 200;  
  res.setHeader('Content-Type', 'text/plain');  
  res.end('Hello, World!\n');  
});
```

- **server.listen:** Starts the server on the specified port and hostname. The callback function is executed once the server is listening.

```
server.listen(port, hostname, () => {  
  console.log(`Server running at http://${hostname}:${port}/`);  
});
```

Testing:

1. Run `node file.js` and leave it running.
2. Open a web browser
3. Enter the URL: `http://127.0.0.1:3000`
4. You should see "Hello, World!" displayed on the page.

MAKE A SINGLE PAGE APPLICATION WITH NODE HTTP ROUTES:

- Import the `http` module, define the hostname and port, and create an HTTP server using `http.createServer`.

- The server will use an `if...else if...else` structure to handle different URLs (routes) based on the `req.url` property of the request object.
- If the URL is `/`, it should display a "Welcome to the homepage!" message.
- Check for `/about` and `/contact` URLs and send specific messages for each route.
- If the URL doesn't match any defined route, return a 404 Not Found status code and a "Not found" message.

Create a Node server as above. Use this to replace the previous contents of `createServer`:

```
const url = req.url;

if (url === '/') {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Welcome to the homepage!');
} else if (url === '/about') {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('This is the about page.');
```

```
} else if (url === '/contact') {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Contact us at support@example.com');
} else {
  res.statusCode = 404;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Not found');
```

```
}
```

Run with `node file.js`

Leave that running and test by going to these URLs. You should see different results for each one

`localhost:3000`

`localhost:3000/about`

`localhost:3000/contact`

`localhost:3000/abc`

READ THE URL'S PARAMETERS (QUERY STRINGS):

Start with the node server code again. We will do the following.

Read query parameters:

- Inside the server callback function we will use:
 - `req.url`: This property contains the complete URL requested by the client.

- `url.parse(req.url, true).query`:
 - This line parses the URL using the `url` module.
 - `true` as the second argument indicates parsing the query string into an object.
 - The resulting object (`queryObject`) will contain key-value pairs for each parameter in the URL after the `?` symbol.

Display a conditional message:

- `let message = 'Hello!';` Initializes a variable to hold the message to be sent.
- `if (queryObject.name)`: Checks if there's a key named `"name"` in the `queryObject`.
 - If the name parameter exists:
 - `message =Hello, ${queryObject.name}!;` Uses string interpolation to personalize the message by including the value of the `name` parameter.

Respond to the client:

- The server sets the status code to 200 (OK), sets the content type to plain text, and sends the personalized message or the default "Hello!" message based on the presence of the `name` parameter.

The contents of `createServer` should be:

```
const queryObject = url.parse(req.url, true).query;
let message = 'Hello!';

if (queryObject.name) {
  message = `Hello, ${queryObject.name}!`;
}

res.statusCode = 200;
res.setHeader('Content-Type', 'text/plain');
res.end(message);
```

Run with `node file.js`

Leave the node server running and test with:

`http://127.0.0.1:3000?name=YourName`