Task 1: Basic Syntax and DOM Manipulation

Objective:

Create a simple JavaScript application that processes user input for their name, age, and whether they are a student. The application should display a personalized message based on the input. Additionally, implement a function to toggle a Boolean value for a separate task.

Pre-requisites:

- Understanding of basic JavaScript (variables, data types, functions)
- Familiarity with conditional statements
- Knowledge of loops

Concepts Covered:

- Variables and data types
- Conditional statements
- Loops

Setup:

Install Node.js:

- Ensure Node.js is installed on your machine. You can download it from nodejs.org.

Tasks:

1. **Create Personalized Message Function:**
    - **Task:**
        - Define variables to store the user's name, age, and student status.
        - Create a function to generate a personalized message based on the input values.
        - Use conditional statements to check if the user is a student and create a personalized message.
        - Use a loop to repeat the personalized message based on the user's age (if age is less than or equal to 10, repeat the message that many times).
    - **Outcome:**
        - Ensure the function generates the correct personalized message and repeats it as required.

    Example:

```
// index.js
const userName = process.argv[2];
const userAge = parseInt(process.argv[3]);
const isStudent = process.argv[4] === 'true';

function createMessage(name, age, student) {
  let message = `Hello, my name is ${name}`;
  if (student) {
    message += ' and I am a student.';
  }
  if (age <= 10) {
    // Insert code logic here
  } else {
    console.log(message);
  }
}

createMessage(userName, userAge, isStudent);
```

2. **Implement Toggle Function:**
   - **Task:**
     - Implement a function to toggle a boolean value.
   - **Outcome:**
     - Ensure the function toggles the boolean value correctly.

   Example:

```
// index.js
let toggleValue = false;

function toggleBoolean() {
  toggleValue = !toggleValue;
  console.log(`Toggled value: ${toggleValue}`);
}

if (process.argv[2] === 'toggle') {
  toggleBoolean();
}
```

Instructions:

- **Perform the following tasks:**
  - Write the required code in `index.js`.
  - Run the file using Node.js to ensure the code executes without errors and demonstrates the use of basic JavaScript concepts.

Example Input:

1. Set 1:

   - Name: "Alice"
   - Age: 5
   - Is Student: true

   Expected Output:

   ```
   "Hello, my name is Alice and I am a student."
   "Hello, my name is Alice and I am a student."
   "Hello, my name is Alice and I am a student."
   "Hello, my name is Alice and I am a student."
   "Hello, my name is Alice and I am a student."
   ```

2. Set 2:
   - Name: "Bob"
   - Age: 15
   - Is Student: false

   Expected Output:

   ```
   "Hello, my name is Bob."
   ```

Resources:

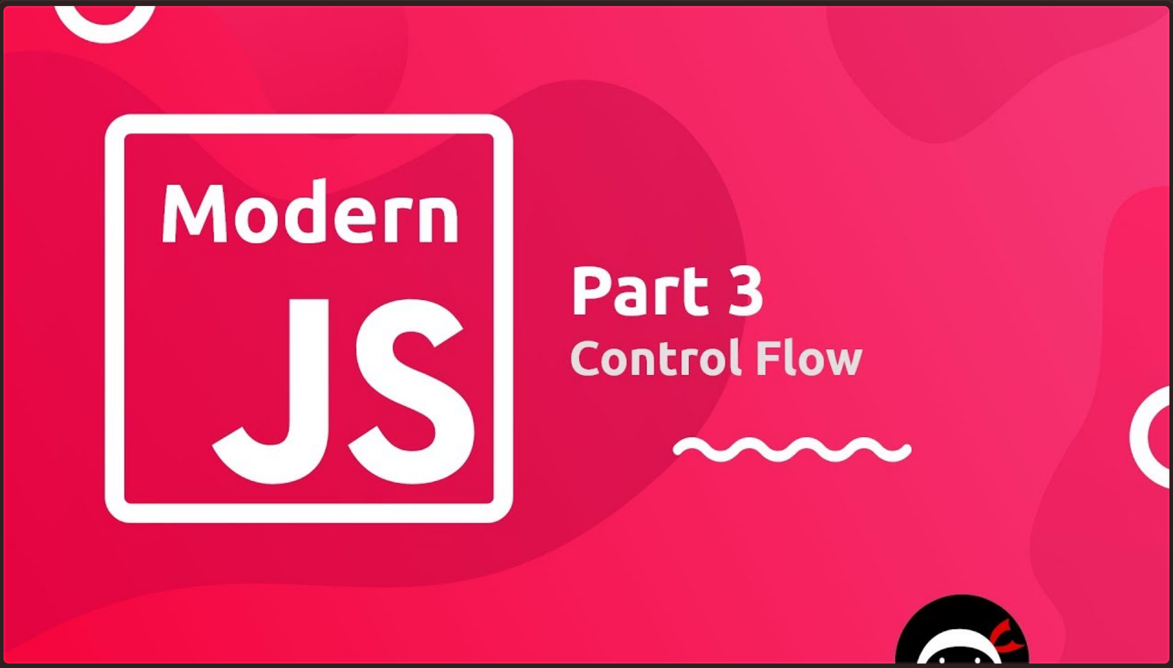
- Grammar and types - JavaScript | MDN
  This chapter discusses JavaScript's basic grammar, variable declarations, data types and li...
  https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Grammar_and_types#declarations
- Control flow and error handling - JavaScript | MDN
  JavaScript supports a compact set of statements, specifically control flow statements, th...
  https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control_flow_and_error_handling
- Loops and iteration - JavaScript | MDN
  Loops offer a quick and easy way to do something repeatedly. This chapter of the JavaScr...
  https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Loops_and_iteration

Videos:

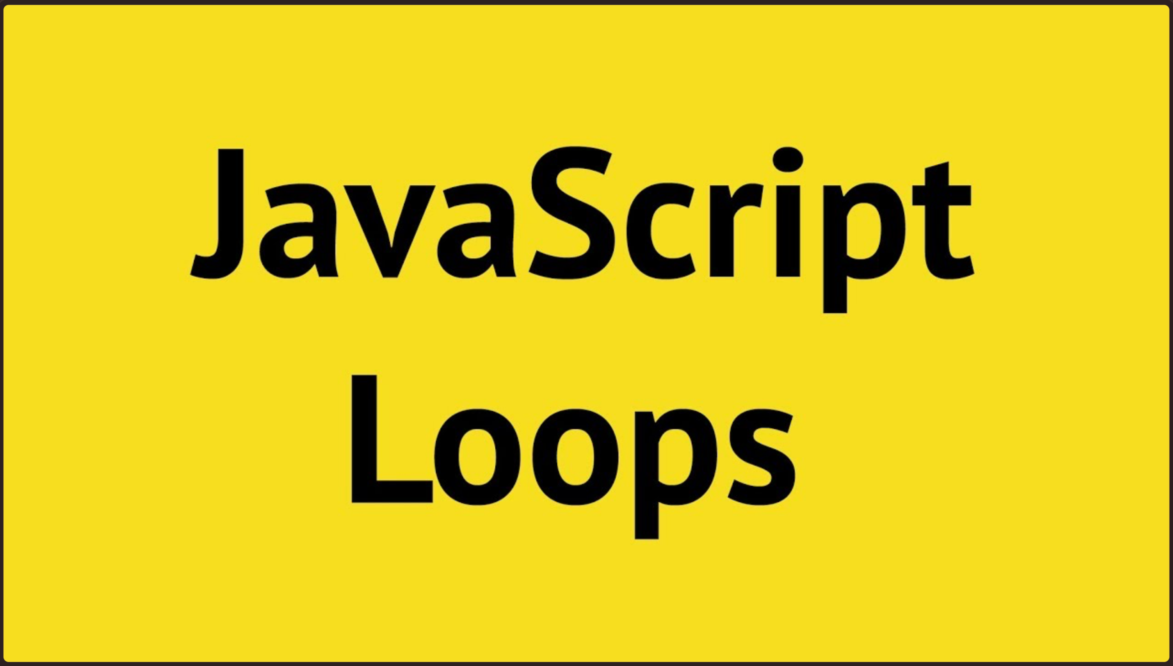
-

```
function greet(user, msg){
    console.log(msg, user)
}
greet("Navin", "Welcome Back")

let i = 6

do {
    console.log("Telusko")
    i++
} while(i<=5)
```

  Telusko 16:52 min 247,392 Views 5,151 Likes

- 

  Net Ninja 53:00 min 248,759 Views 3,452 Likes

- 

Programming with Mosh 6:49min 412,467 Views 7,764 Likes

GitHub Instructions:

1. **Open in Visual Studio Code:**
   - After clicking on the "Open in Visual Studio Code" button from the GitHub Classroom confirmation page, Visual Studio Code (VSCode) will open the repository directly.
   - If prompted, select "Open" or "Allow" to open the repository in VSCode.
2. **Open the Terminal in VSCode:**
   - In VSCode, open a terminal by selecting Terminal > New Terminal from the top menu.
3. **Complete the Task:**
   - In VSCode, write your solution in the `index.js` file.
4. **Run and Test Your Code:**
   - In the VSCode terminal, navigate to the directory containing `index.js`.
   - Run your code to ensure it works correctly. Use the following commands:

```
node index.js "Alice" 5 true
node index.js "Bob" 15 false
node index.js "Charlie" 8 true
node index.js toggle
```

5. **Commit Your Changes:**
   - In the VSCode terminal, add your changes to git:

```
git add index.js
```

- Commit your changes with a meaningful message:

  ```
  git commit -m "Completed task 1"
  ```

6. **Push Your Changes to Your Repository:**
   - Push your changes to your forked repository:

     ```
     git push origin main
     ```

7. **Create a Pull Request:**
   - Go to your repository on GitHub.
   - Click on the "Pull Requests" tab.
   - Click the "New Pull Request" button.
   - Ensure the base repository is the original template repository and the base branch is `main` .
   - Ensure the head repository is your forked repository and the compare branch is `main` .
   - Click "Create Pull Request".
   - Add a title and description for your pull request and submit it.

Summary of Commands:

```
# Open in Visual Studio Code

# Open terminal in VSCode

# Complete the task by editing index.js

# Navigate to the directory containing index.js
cd path/to/your/index.js

# Run your code
node index.js "Alice" 5 true
node index.js "Bob" 15 false
node index.js "Charlie" 8 true
node index.js toggle

# Add, commit, and push your changes
git add index.js
git commit -m "Completed task 1"
git push origin main

# Create a pull request on GitHub
```