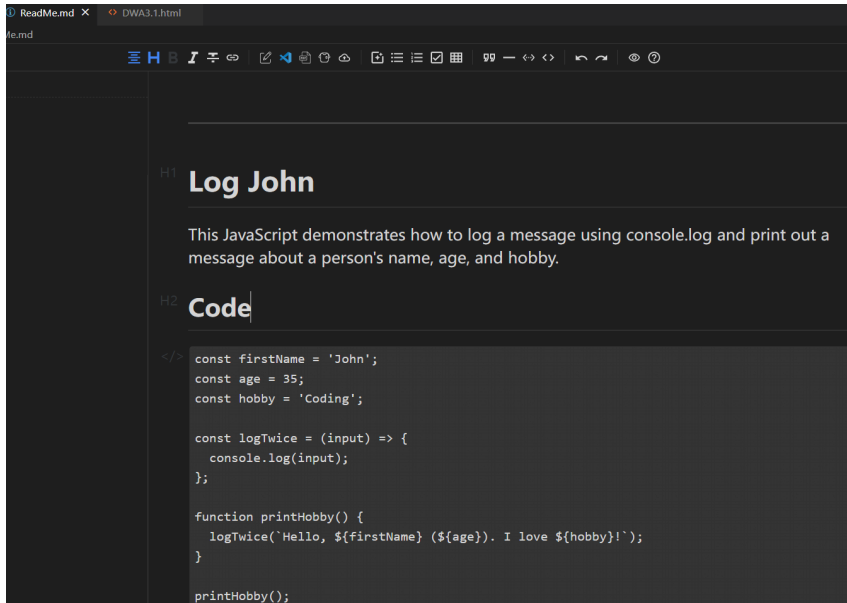


DWA_03.4 Knowledge Check_DWA3.1

1. Please show how you applied a Markdown File to a piece of your code.



The screenshot shows a code editor with a dark theme. The top bar shows two tabs: 'ReadMe.md' and 'DWA3.1.html'. The active tab is 'ReadMe.md'. The editor content is a Markdown file. It starts with an H1 heading 'Log John'. Below the heading is a paragraph: 'This JavaScript demonstrates how to log a message using console.log and print out a message about a person's name, age, and hobby.' Below the paragraph is an H2 heading 'Code'. Below the heading is a code block containing JavaScript code. The code defines three constants: 'firstName' (value 'John'), 'age' (value 35), and 'hobby' (value 'Coding'). It also defines a function 'logTwice' that takes an 'input' parameter and logs it twice. Finally, it defines a function 'printHobby' that calls 'logTwice' with a formatted string: 'Hello, \${firstName} (\${age}). I love \${hobby}!'. The code block is highlighted with a light blue background.

```
const firstName = 'John';
const age = 35;
const hobby = 'Coding';

const logTwice = (input) => {
  console.log(input);
};

function printHobby() {
  logTwice(`Hello, ${firstName} (${age}). I love ${hobby}!`);
}

printHobby();
```

2. Please show how you applied JSDoc Comments to a piece of your code.

```
/**
 * @const {string} firstName - The first name of the person.
 */
const firstName = 'John'

/**
 * @const {number} age - The age of the person.
 */
const age = 35

/**
 * @const {string} hobby - The hobby of the person.
 */
const hobby = 'Coding'

/**
 * Logs the given input twice.
 *
 * @param {string} input - The input to be logged.
 */
const logTwice = (input) => {
  console.log(input)
  console.log(input)
}

/**
 * Prints a message with the person's information and hobby.
 */
function printHobby() {
  /**
   * A formatted message including the person's name, age, and hobby.
   * @type {string}
   */
  const message = `Hello, ${firstName} (${age}). I love ${hobby}!`;

  logTwice(message);
}

// Call the function to print the hobby message.
printHobby();
```

3. Please show how you applied the @ts-check annotation to a piece of your code.

```
1 // @ts-check
2 /**
3  * @const {string} firstName - The first name of the person.
4  */
5 const firstName = 'John'
6
7 /**
8  * @const {number} age - The age of the person.
9  */
10 const age = 35
11
12 /**
13  * @const {string} hobby - The hobby of the person.
14  */
15 const hobby = 'Coding'
16
17 /**
18  * Logs the given input twice.
19  *
20  * @param {string} input - The input to be logged.
21  */
22 const logTwice = (input) => {
23   console.log(input)
24   console.log(input)
25 }
26
27 /**
28  * Prints a message with the person's information and hobby.
29  */
30 function printHobby() {
31   /**
32    * A formatted message including the person's name, age, and hobby.
33    * @type {string}
34    */
35   const message = `Hello, ${firstName} (${age}). I love ${hobby}!`;
36   logTwice(message);
37 }
38
39 // Call the function to print the hobby message.
40 printHobby();
41
42
```

4. As a BONUS, please show how you applied any other concept covered in the 'Documentation' module.

```
// @ts-check

/**
 * @typedef {Object} Person
 * @property {string} firstName - The first name of the person.
 * @property {number} age - The age of the person.
 * @property {string} hobby - The hobby of the person.
 */

/**
 * @type {Person}
 */
const person = {
  firstName: 'John',
  age: 35,
  hobby: 'Coding',
}

/**
 * Logs the given input twice.
 *
 * @param {string} input - The input to be logged.
 */
const logTwice = (input) => {
  console.log(input);
};

/**
 * Prints a message with the person's information and hobby.
 *
 * @param {Person} person - The person object.
 */
function printHobby(person) {
  /**
   * A formatted message including the person's name, age, and hobby.
   * @type {string}
   */
  const message = `Hello, ${person.firstName} (${person.age}). I love ${person.hobby}!`;

  logTwice(message)
}

// Call the function to print the hobby message.
printHobby(person)
```
