

DWA_03.4 Knowledge Check_DWA3.1

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

```
1  ## Code Snippet
2
3  ```javascript
4  // Define a function to calculate the square of a number
5  function calculateSquare(num) {
6      return num * num;
7  }
8
9  // Call the calculateSquare function with an argument
10 const result = calculateSquare(5);
11
12 console.log(result);
```

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

```
1  /**
2   * Represents a person.
3   * @class
4   */
5  class Person {
6      /**
7       * Create a person.
8       * @constructor
9       * @param {string} name - The name of the person.
10      * @param {number} age - The age of the person.
11      */
12     constructor(name, age) {
13         this.name = name;
14         this.age = age;
15     }
16
17     /**
18      * Get the name of the person.
19      * @returns {string} The name of the person.
20      */
21     getName() {
22         return this.name;
23     }
24
25     /**
26      * Set the name of the person.
27      * @param {string} name - The new name of the person.
28      */
29     setName(name) {
30         this.name = name;
31     }
32
33     /**
34      * Get the age of the person.
35      * @returns {number} The age of the person.
36      */
37     getAge() {
38         return this.age;
39     }
40
41     /**
42      * Set the age of the person.
43      * @param {number} age - The new age of the person.
44      */
45     setAge(age) {
46         this.age = age;
47     }
48 }
49
50 // Create a new person object
51 const person = new Person('Alice', 25);
52
53 // Get the name and age of the person
54 console.log(person.getName());
55 console.log(person.getAge());
```

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

```
1 // @ts-check
2
3 /**
4  * Represents a person.
5  * @class
6  */
7 class Person {
8   /**
9    * Create a person.
10   * @constructor
11   * @param {string} name - The name of the person.
12   * @param {number} age - The age of the person.
13   */
14   constructor(name, age) {
15     this.name = name;
16     this.age = age;
17   }
18
19   /**
20    * Get the name of the person.
21    * @returns {string} The name of the person.
22    */
23   getName() {
24     return this.name;
25   }
26
27   /**
28    * Set the name of the person.
29    * @param {string} name - The new name of the person.
30    */
31   setName(name) {
32     this.name = name;
33   }
34
35   /**
36    * Get the age of the person.
37    * @returns {number} The age of the person.
38    */
39   getAge() {
40     return this.age;
41   }
42
43   /**
44    * Set the age of the person.
45    * @param {number} age - The new age of the person.
46    */
47   setAge(age) {
48     this.age = age;
49   }
50 }
51
52 // Create a new person object
53 const person = new Person('Alice', 25);
54
55 // Get the name and age of the person
56 console.log(person.getName());
57 console.log(person.getAge());
```

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

```
1  /**
2   * Represents a person.
3   * @class
4   */
5  class Person {
6      /**
7       * Create a person.
8       * @constructor
9       * @param {string} name - The name of the person.
10      * @param {number} age - The age of the person.
11      */
12      constructor(name, age) {
13          this.name = name;
14          this.age = age;
15      }
16
17      /**
18       * Get the name of the person.
19       * @returns {string} The name of the person.
20       */
21      getName() {
22          return this.name;
23      }
24
25      /**
26       * Set the name of the person.
27       * @param {string} name - The new name of the person
28       */
29      setName(name) {
30          // Update the name of the person
31          this.name = name;
32      }
33
34      /**
35       * Get the age of the person.
36       * @returns {number} The age of the person.
37       */
38      getAge() {
39          return this.age;
40      }
41
42      /**
43       * Set the age of the person.
44       * @param {number} age - The new age of the person.
45       */
46      setAge(age) {
47          // Update the age of the person
48          this.age = age;
49      }
50  }
51
52  // Create a new person object
53  const person = new Person('Alice', 25);
54
55  // Get the name and age of the person
56  console.log(person.getName()); // Output: Alice
57  console.log(person.getAge()); // Output: 25
```
