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.

```js
/**
 * Represents a person.
 * @class
 */
class Person {
  /**
   * Create a person.
   * @constructor
   * @param {string} name - The name of the person.
   * @param {number} age - The age of the person.
   */
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }

  /**
   * Get the name of the person.
   * @returns {string} The name of the person.
   */
  getName() {
    return this.name;
  }

  /**
   * Set the name of the person.
   * @param {string} name - The new name of the person.
   */
  setName(name) {
    this.name = name;
  }

  /**
   * Get the age of the person.
   * @returns {number} The age of the person.
   */
  getAge() {
    return this.age;
```

_____

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

```javascript
// @ts-check

/**
 * Represents a person.
 * @class
 */
class Person {
  /**
   * Create a person.
   * @constructor
   * @param {string} name - The name of the person.
   * @param {number} age - The age of the person.
   */
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }

  /**
   * Get the name of the person.
   * @returns {string} The name of the person.
   */
  getName() {
    return this.name;
  }

  /**
   * Set the name of the person.
   * @param {string} name - The new name of the person.
   */
  setName(name) {
    this.name = name;
  }

  /**
   * Get the age of the person.
   * @returns {number} The age of the person.
   */
```

_____

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

```javascript
/**
 * Represents a person.
 * @class
 */
class Person {
  /**
   * Create a person.
   * @constructor
   * @param {string} name - The name of the person.
   * @param {number} age - The age of the person.
   */
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }

  /**
   * Get the name of the person.
   * @returns {string} The name of the person.
   */
  getName() {
    return this.name;
  }

  /**
   * Set the name of the person.
   * @param {string} name - The new name of the person
   */
  setName(name) {
    // Update the name of the person
    this.name = name;
  }

  /**
   * Get the age of the person.
   * @returns {number} The age of the person.
```

_____