

# Appendix 1

## Comments & Documentation

Shadi Lahham - Programmazione web - Frontend - Javascript

# Comments

Comments are human-readable text ignored by the computer

Why comments are important

- Comments help you understand what you are coding
- They help other programmers understand your code
- They help you to remember what code that you wrote 3 months ago does

# Documentation & comments in HTML

# Required HTML elements

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>The mysterious orient</title>
  <meta name="description" content="An account of my travels in the east">
  <meta name="author" content="marco polo">
</head>

</html>
```

An HTML page must at least include the following:  
Doctype, language, charset, title, description and author.

# Required HTML elements

```
<body>  
  <!-- page content -->  
  <div>meaningful content here</div>  
  
  <!-- end of the body -->  
  <script src="js/scripts.js"></script>  
</body>  
  
</html>
```

A comment must be included to indicate the end of the body before any scripts.  
If there are many included scripts, add a comment for each to explain what they do.

# HTML comments

```
<!-- This is a single line comment -->
```

```
<!--  
This is a comment  
that spans multiple lines  
-->
```

```
<!-- This is an example of a Closing Tag Comment -->
```

```
<div id="main-wrapper">  
  ...  
</div><!-- #main-wrapper -->
```

# Comments in CSS

# CSS comments example

```
/**
 * icons.css
 *
 * Icons should convey in a simple way the concept of they represent.
 * When designing new icons remove complexities and follow the linear appearance of the set.
 */

.icon {
  font-family: 'family name';
  font-style: normal;
  text-transform: none !important; /* forces transform */
}

/* individual icons */
.icon-microchip:before {
  content: "\f2db";
}
```



# Comments in Javascript

# JS Comments

```
// You can write single-line comments
```

```
let x = 4; // Or you can comment after a statement
```

```
/*
```

```
Or you can write multi-line comments, if you have something very long  
to say like this gratuitously long description.
```

```
*/
```

# JS Comments and documentation

Code, especially functions, should be well commented:

```
/*  
 * Adds two numbers and returns the sum  
 */  
function addNumbers(num1, num2) {  
    return num1 + num2;  
}
```

# JS Comments and documentation

Using JSDoc documentation standard is even better:

```
/**
 * Returns the sum of num1 and num2
 * @param {number} num1 - the first number
 * @param {number} num2 - the second number
 * @returns {number} Sum of num1 and num2
 */
function addNumbers(num1, num2) {
    return num1 + num2;
}
```

[Use JSDoc: Documentation](#)

[JSDoc on github](#)

# JS File documentation

Each Javascript file header should be well commented:

```
/**
 * @file: filename.js
 * @author: author's name
 * Purpose of file
 *
 * Detailed explanation of what the file does
 * on multiple lines
 */
```

# JS File documentation

## Example:

```
/**
 * @file: math-demo.js
 * @author: james jones
 * A demonstration of Math functions
 *
 * This exercise file demonstrates some uses of the JavaScript Math Object
 * More examples at: https://www.w3schools.com/js/js\_math.asp
 */

Math.round(4.7);    // returns 5
Math.round(4.4);    // returns 4
Math.pow(8, 2);     // returns 64
Math.abs(-4.7);     // returns 4.7
Math.ceil(4.4);     // returns 5
Math.floor(4.7);    // returns 4
```

# Comments

- All the code that you write for this course must be properly commented
- Code without proper comments will be considered incomplete

# Markdown



# What is markdown

- A plain text formatting syntax aimed at making writing for the internet easier
- Plain text documents should be readable without tags
- There should still be ways to add text modifiers like lists, bold, italics, etc

## Common usages

- In rich input fields on the web
  - e.g. forums, comments, etc
- In project documentation
  - most commonly as **readme.md**

# Markdown features

- Easy to learn
- Fast to use
- Converts to HTML easily
- Flexible use
  - Project documentation
  - Rich user input in blogs, forums, comments
- Quite readable even as plain text

# Markdown in action

An h1 header

=====

Paragraphs are separated by a blank line.

2nd paragraph. *\*Italic\**, **\*\*bold\*\***, and ``monospace``. Itemized lists look like:

- \* this one
- \* the other one

> Block quotes are written like so.

# Markdown in action

An h2 header

-----

Here's a numbered list:

1. first item
2. second item
3. third item

```
```javascript
// this is javascript code
function addNumbers(num1, num2) {
  let result = num1 + num2;
}
```
```

# Documentation

- All exercises and projects that you submit for this course must have proper documentation
- Include a **readme.md** file in the root folder
- Projects without proper documentation will be considered incomplete

# Further reading

[HTML style guide with coding standards and best practices](#)

[Commenting in HTML » Best Practices Web Design](#)

# Further reading

[Commenting JavaScript code with JSDoc](#)

[An introduction to JSDoc](#)

[JSDoc](#)

[JSDoc returns](#)

[Source Code Comment Styling: Tips and Best Practices](#)

# Further reading

[Markdown Basics](#)

[Markdown - Getting Started](#)

[Learn X markdown in Y minutes](#)

[Online Markdown Editor - Dillinger](#)