



Coding a comic

Introduction

In this project children create a sequence of juxtaposed images and texts, also known as a *comic*, with HTML and CSS.

Do you **like comics**?

Do you have a **story to tell**?

Can you turn a bunch of images into **your own Web-comic**?

Resources

Files included in this project's **Resources** folder

- *waiter.png*

Software required

- A code editor. We recommend [Sublime Text 2](#)
- A browser with Developer Tools enabled. We recommend [Google Chrome](#)

Children are asked to use their own images in this project, but they may want to source images online. They should therefore be able to access resources like [Flickr Creative Commons](#).

Learning Objectives

1. Use a **code editor** to write code, and the **browser** to debug it
2. Structure a Web page with the **appropriate markup**
3. Identify and use **openly-licensed images**
4. Create something new on the Web using existing resources
5. Use CSS to change the **style and layout** of a Web page
6. Understand the **syntactic differences** between HTML and CSS
7. **HTML**
 1. Elements introduced: `section`, `img`, `p`
 2. Nest HTML elements inside each other to create a structured hierarchy
 3. Use `class` names to create groups of elements that share some properties
 4. Use `id` names to reference unique elements
 5. Understand the difference between `class` and `id`
8. **CSS**
 1. Use *attribute selectors* to target HTML elements appropriately
 2. `width` and `height`
 3. `margin` and `padding`
 4. `border`
 5. `position` and offsets
 6. `float`
 7. `overflow`
 8. typographical properties such as `font-size`, `text-align`, `font-weight`, `line-height` and `font-family`
 9. `after` pseudo-class

Challenges

1. Use the browser's Inspector to experiment with CSS properties
2. Copy-paste existing elements instead of writing copies
3. Use `img` to add image files to an HTML document

4. Use `top` and `left` offsets to position elements
5. Use `id` and `classes` to make both unique and shared CSS rules
6. Use `float` to control the document flow
7. Use `overflow` to control the display of nexted elements
8. Use CSS typographical properties in conjunction with web-fonts to personalize a Web page layout
9. Use `:after` and borders to create triangular shapes

Frequently asked questions

1. *Why has the `img` no closing thing like the others?*
2. *I copied the X file into here but it doesn't come up in my page.*
3. *My page looks weird.*

Check for *syntax errors* in the HTML: unnecessary spaces, closing tags are missing, brackets are missing, quotation marks are missing, etc

4. *My browser is not picking up the CSS I just wrote.*

Check for *typos* and *syntax errors* in the `style` : unnecessary spaces, brackets are missing, quotation marks are missing, etc

5. *I changed something in the browser and now it's gone.*

You may have to explain that changes in the Inspector are temporary. Useful while making a page, but wiped on refresh. Otherwise anyone could hack (or improve) any website :)

6. *I want to have X panels on the same line but the browser is putting them on two lines.*
7. *My image is too big for this page.*
8. *I picked a font from Google Fonts but it's not showing up in my comic.*

License: CC BY-SA 4.0 **Author:** Matteo Menapace