Front-end development

Module 5

JavaScript

Bits and Bots
study group

Content

This Module

Module 5 will focus on further exploring the basics of JavaScript and using that to further develop what you already have by making an interactive website. Additionally, you can use this module to start formulating an idea for your final project. This final project is something you can do by yourself, but you can also do it with someone else.

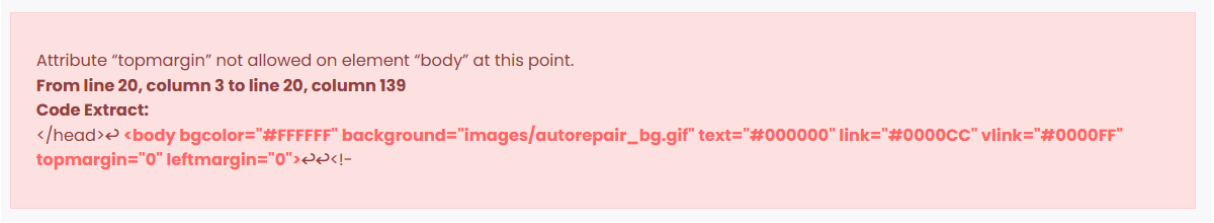
Answers Exercise Module 4

Exercise

In the expert session, we dove a bit deeper into questions digital preservationists can ask themselves when it comes to web archiving. These questions can relate to technology but also to archival principles.

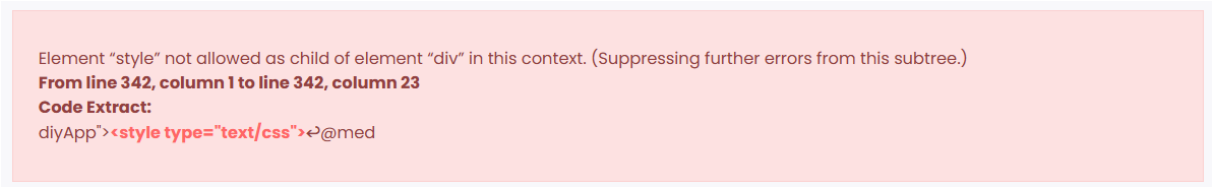
Solution

Because websites change, it is highly probable that any solutions about the website HTML we could give here would differ from the solutions you'd find at the moment you are doing this exercise. We can however, give you some examples of what you might find during your search. For these examples, the websites https://arngren.net/ and https://www.discountbedsbelfast.co.uk/ were used.



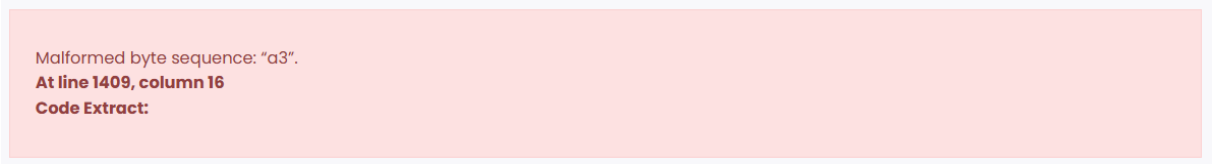
The attribute given to the element is invalid. This can mean that in HTML, this specific attribute cannot be a part of this element according to HTML-rules. It can also mean that the attribute is deprecated.

This is a case of deprecated coding. HTML has been used to style this website, where nowadays CSS would be used.



The HTML structure violates the rules of how HTML elements can be nested. Specifically, it indicates that the element used as a child (inside) is not permitted within the context of its parent element. This often relates to the content model of the parent element, which dictates what types of elements can be its direct children.

Lightbox is a script for overlaying images. The plugin string for the use of Lightbox doesn’t consist of values that are defined in HTML.

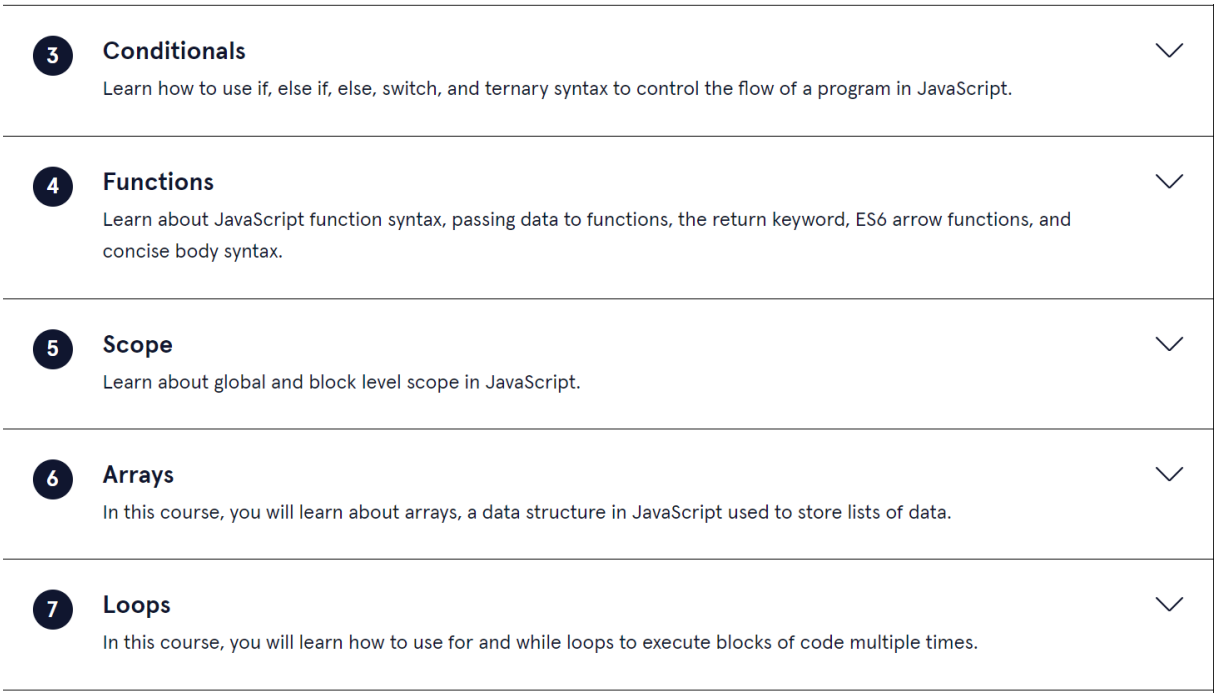


Malformed UTF-8 characters occur when byte sequences do not conform to the encoding rules of UTF-8. These errors can happen due to various reasons, such as data corruption, improper encoding/decoding, or incorrect handling of character sets. Specifying UTF-8 at the top of your HTML-script can help.

Conditionals, Functionals, Arrays, and Loops

JavaScript has some similarities to Python. It also uses many of the same mechanisms such as loops.

Follow the rest of the Introduction to JavaScript course on Codecademy (see screenshot below)



Let's Make Some Games!

There are now a few different games that you can adapt or try making on your webpage using the interactivity provided by JavaScript. Try making one of the following or come up with your own ideas. You can adapt the games below to be more archivally themed, for example the word guessing game could be used alongside archival definitions.

- Magic Eight Ball
- Rock Paper Scissors
- Joke Generator
- Whack-a-Mole
- Word Guessing Game

The nice thing about JavaScript is it can be combined with HTML and CSS so you can add as many images, bright colours and inventiveness to your game as you like.

Final Goal

Try putting together the new knowledge and previous knowledge within the game to create a new website that displays your JavaScript Game. Share it with the group!

Start thinking of the final project you wish to develop. The final modules will be much more self-directed and focus on areas such as collaborative working, UX, and playtesting. You may wish to put more effort into learning some skills over others in the next few modules and will have the flexibility to do so. Ideally you will form groups and work on a larger project together to share at the end of Module 7[1]. Though you are also welcome to work alone. Whilst working on this module consider everything you have learnt so far. Are there any projects that now may seem achievable that you wish to make? Do you have any ideas for a final project? What areas do you wish to focus on?

[1] You can use GitHub as a handy tool when working in a group. This allows you to work on a project (repository) together while changes are being tracked and where multiple branches are possible.

Preservation tools on the Command-Line Interface

For the expert session in October, we are joined by Lode Scheers and Nastasia Vanderperren. They will show us how to use preservation tools on the Command-Line Interface (CLI), specifically ExifTool. If you want to follow along live during the expert session, make sure to follow the steps below ('To install before starting'), beforehand.

Exercise Preservation Tools in the Command-Line Interface

If you are still new at using the command-line interface (CLI), please check the Command-Line Interface Guide.

Please note that these practice based exercises are focused on trying out tools via the CLI. We do not provide answers for the exercise in the next module. Let us or the experts know if you are stuck or need help.

To install before starting[2]

Windows users need to install Scoop

1. open Windows PowerShell
2. paste Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser Invoke-RestMethod -Uri https://get.scoop.sh | Invoke-Expression in the terminal and press <enter>
3. ready!

macOS users need to install Homebrew

1. open your terminal
2. paste /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)" in your terminal and press <enter>
3. in case step 2 ends with an error, paste xcode-select --install in your terminal and press <enter> and repeat step 2
4. ready!

Editing Image Metadata with ExifTool

In this exercise, you will:

1. Download an image (e.g. from Wikimedia Commons).
2. Install command line programs by using a package manager
3. Inspect its embedded metadata (information about the image, like creator, description, etc.).
4. Modify the metadata to include your own values for Creator and Description.
5. Save the modified metadata back into the image.

[2] Struggling with installation? You can ask your question in the Discord server or you can contact Lode (lode.scheers@meemoo.be) or Nastasia (nastasia.vanderperren@meemoo.be) directly.

This is useful if you want to add descriptive information - or remove privacy information - to images you post online, or if you want to practice automating workflows that involve images.

What do you need?

- A terminal (Command Prompt, PowerShell, or Mac/Linux terminal).
- A package manager for installing/updating/removing command line programs (optional, but recommended)
  - Scoop for Windows
  - Homebrew for macOS
- At least one downloaded image (e.g., .jpg file).

Optional (extra challenge): Python users can try to do this with Python as well.

This assignment has four parts: installing ExifTool, inspecting your metadata, exporting your metadata to a CSV file, and re-importing your modified metadata.

Part 1. Installing ExifTool

ExifTool is a command-line tool for reading, writing and editing metadata in images.

1. Open your command-line interface (command prompt or terminal)
2. Install ExifTool by using your package manager (scoop or brew)

   Windows
   ```
   scoop install exiftool
   ```

   macOS
   ```
   brew install exiftool
   ```

If you don't want to use or install a package manager, you can download ExifTool (it works on Windows, Mac, and Linux).

Part 2. Inspecting your metadata

1. Make sure you have downloaded one or more images and have installed ExifTool
2. Make sure your command-line interface (command prompt, powershell or terminal) is opened
3. Navigate to the folder where your images are stored using the *cd* command
4. Run ExifTool on an image. This will list all the metadata currently inside the image.

   ```
   exiftool [yourimagename.jpg]
   ```

Part 3. Exporting your metadata to a CSV file

1. Use ExifTool to export metadata from all images in a folder:

   ```
   exiftool -csv *.jpg > metadata.csv
   ```

2. Open metadata.csv in a spreadsheet program (e.g, Excel or Libreoffice)
3. Check if the Creator and Description columns exist

a) If they exist → try to change some values
b) If they don't exist → add those columns and fill in values

Part 4. Re-importing your modified metadata

1. Save your edited metadata.csv
2. Import it back into your images:

```
exiftool -csv=metadata.csv -overwrite_original_in_place *.jpg
```

This updates the metadata in each image according to the CSV file.