Guide to Website/ Twine Based Games

With the Bits and Bots study group

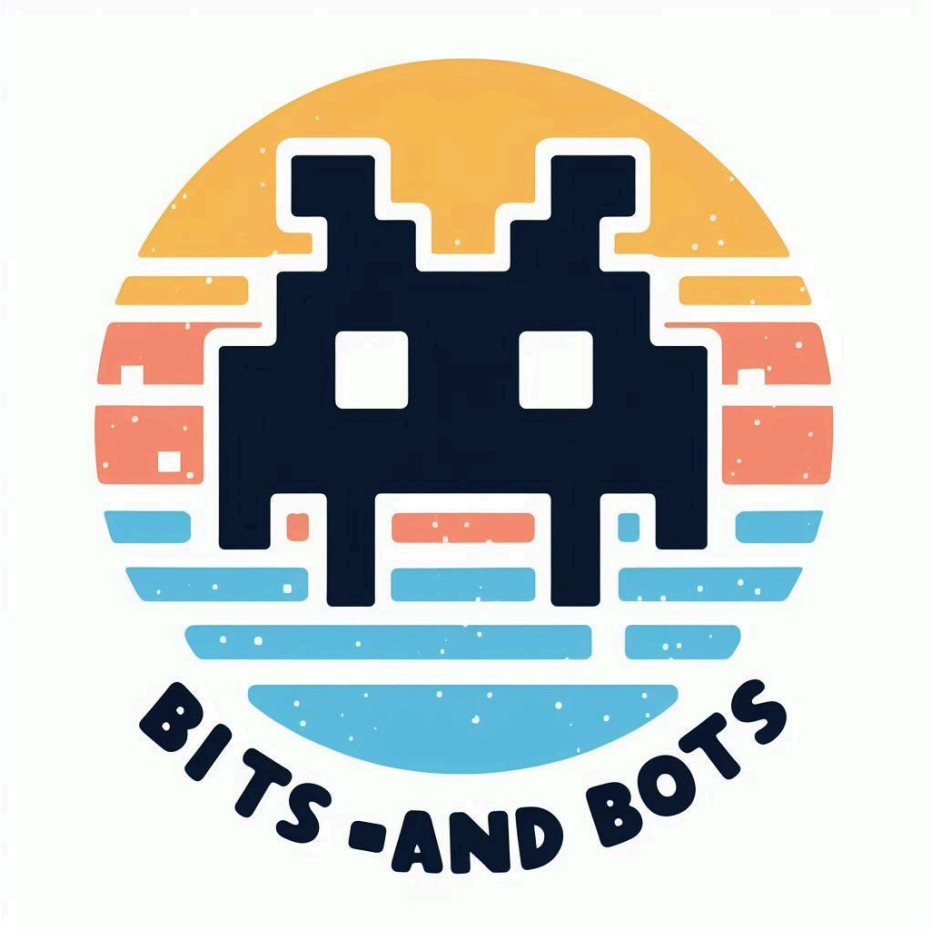


Table of Contents

Introduction

Learning Objectives

This course includes familiarisation with tools such as Twine, GitHub and Google Pages as well as knowledge of the coding languages MarkDown, HTML, CSS and JavaScript that are used for creating web pages. The main goal is to gain some understanding of lots of different tools, to give you flexibility and allow you to pick the best options, plans for your specific final project. Or give you an indicator of what you would wish to focus on.

Module Overview

Module	Dates
1	15 Feb - 15 March
2	15 March - 15 April
3	15 April - 15 May
4	15 May - 15 June
5	15 June - 15 July
Summer recess	
6	15 September - 15 October
7	15 October - 15 November
8	15 November - 15 December

Note: for dates of monthly meetings, have a look at the General Guide.

Module 1: Websites 101 and Twine Basics

In this module we will:

1. Learn about the basic components that make up a website
2. Create a GitHub account and learn about why GitHub is so great!
3. Understand the basics of Twine and build our first simple game

End Goal: to have a better understanding of the context of what we will be learning and to have created your first text based game!

Module 2: HTML, Markdown and Your First Website

In this module we will:

1. Learn some of the basics about writing HTML
2. Learn some of the basics about writing MarkDown
3. Create your own website using GitHub and/or GooglePages
4. Put the Twine game you created on it's own website

End Goal: to have created your first website, or multiple first websites. One of which contains your Twine game.

Module 3: HTML and CSS

In this module we will:

1. Learn how to format websites
2. Learn how CSS and HTML can be applied to Twine

End Goal: Use CSS and HTML to improve something you've already created OR create a new Twine Game/ Website which includes the skills you've learnt.

Module 4: Embedding and Tools

In this module we will:

1. Explore different tools we can embed into websites

End Goal: To create some exciting and unique frankensteinian games!!!

Module 5: JavaScript

In this module we will:

1. Explore the basics of JavaScript

2. Create interactive websites
3. Start formulating ideas for a final project

End goal: Create an interactive website e.g a quiz, or a button changing colour

Module 6: JavaScript Continued/ Advanced Twine

At this stage we are hoping that you will have formed groups, or decided to solo create a final project. The contents of this module will equip you to work collaboratively and you can choose a focus in improving your skills in Twine/ JavaScript or both!

In this module we will:

1. Form groups/ go solo and come up with ideas for a final project
2. Learn more about GitHub as a collaborative tool
3. Learn more about JavaScript (optional)
4. Look at the capabilities of Twine (optional)

Module 7: Prototyping, User Testing and Design

In this module we will:

1. Focus on more in-depth learning on areas covered in the course
2. Work towards a prototype of your game
3. Conduct play testing and look at principles of UX

End goal: Create a prototype of your final game and test it on colleagues for iterative improvements.

Module 8: Final Projects

In this module we will:

1. Refine and finish off the final games

End goal: A GAME

Module 1: Websites 101 and Twine Basics

Websites 101

We thought that it might be useful to do some exploring and reading before starting out with HTML in the next module. A good introduction to website development that includes clear definitions on the different coding languages and a bit more of an overview of how website development works can be found here. Based on this resource as a starting point please feel free to search and explore these concepts further as suits you. It would be great to share any other resources you find helpful as a starting point. A rather less serious, fun and only slightly relevant video about country internet codes can be found here for general enjoyment.

Creating Your First GitHub Account

GitHub is the ultimate collaboration tool for coding projects but can be used as a platform for any collaborative work. It's also really useful to store your work and share it. GitHub also allows you to create free webpages which we will look at later! In this part of the module we will understand more about why GitHub is so useful, create your account and try out some of the features. To do this we will mostly be following some of the tutorials provided by GitHub itself for learning.

To start with, read more about GitHub here. One thing to bear in mind is that GitHub, while an amazing tool does come with a lot of jargon. A great explanation for some of the terminology around GitHub can be found here.

To create your first GitHub account you will first need to register at www.github.com and follow the instructions provided here.

Finally it's time to try out a few features with your new account. This will be creating a repository, creating a pull request, creating branches and merging a pull request. If these terms seem strange then that is because they are slightly. A link to the tutorial you will be following for the first stages of GitHub can be accessed here.

If you are enjoying learning and wish to practice further then you can follow the tutorial on the GitHub skills page 'Introduction to GitHub'

Building Your First Twine Game

Background

Twine is an open-source tool for telling interactive, nonlinear stories.

Why would you want to use Twine? This introduction gives you good examples of why this tool can be useful.

Play Some Games

To start this module lets try playing some games that have been created by Twine to get a feel for them:

Examples of notable games that were created with Twine:

- Rat Chaos
  This game is about a spaceship captain choosing between going about their daily routine or "unleashing rat chaos". Fun fact: While learning the software Twine, the author created this game in a couple hours on a single day (July 18th, 2012).
  Link: https://debacle.us/ratchaos/
- You Are Jeff Bezos
  You start the game by waking up as Jeff Bezos. To get back to your own body, you need to spend all of Jeff Bezos' money. Then the game provides you with various choices that you can spend the money on. These choices evoke reactions.
  Link: https://direkris.itch.io/you-are-jeff-bezos

Try Out Twine

Twine is all about creativity and writing. Now that you've played some games the next stage of the module is to try out the software yourself. To introduce yourself to Twine we will go to https://twinery.org/ and press click on browser. For this part of the module we will be following along with the first video in Adam Hammond's https://www.adamhammond.com/twineguide/. Bear in mind what the final games you played looked like, and try and see how the techniques that you are learning may be shown in the final product of a game. If you prefer not to follow the video the Video Game Museum has another guide to follow or there is the Twine reference guide as well (though this can be a bit overwhelming).

Developing Your Own Story

Now that you are more familiar with how Twine works it is time to take a step back and look at how you may apply it to something you would like to create. Taking into account

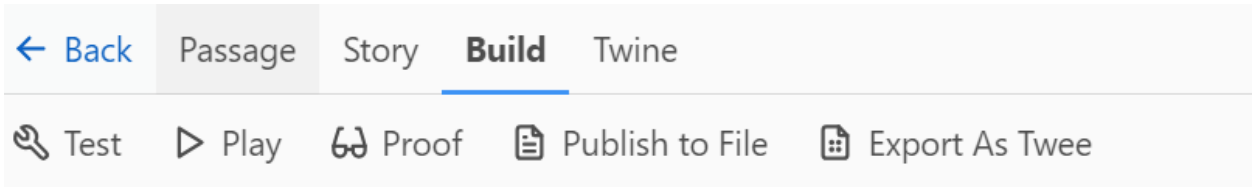
what you have learnt, try and create a story or a game within those parameters. It may not be digital preservation related if you don't wish it to be so. Try looking up other Twine games to inspire you online.

Create Your First Story

The next stage is to create your first story using the techniques from the tutorial provided above and if you prefer reminders in written text or to explore more functionality then the Twine reference guide is also available here.

Share Your Story

For this module we will be saving the story as an html file. To do this go to Build at the top of the screen and then 'Publish to File'



This will download a copy of the game. If you click on this you will be able to view and play the game in your browser! Or send it to others to play as well!

Browser vs. Desktop App

For this module we have been using the browser to create our first game. However it is also possible to install the **desktop app**. Both versions are available on the Twine Homepage.

There are pros and cons of each version:

The **browser** version release uses your selected web-browser's Local Storage to store all the passages of your story, and by default the web-browser's Local Storage area is limited in size to between 5-10MBs (depends on the brand & version) which means you are limited if you want to embed media files within the passages of your story. The Local Storage are can also be affected by the "Empty Cache" (or similar concepts) feature of the web-browser (this depends on brand & settings) so it is recommend to regularly use the Twine 2 Archive option to backup all your projects in one go, or to use the Publish to File option to create locally stored HTML files for each of your projects.

The **desktop app** release stores each Story Project as a local file in a automatically predetermined location/folder on your local machine, this location should not be used by you as a destination for the Story HTML file you create using the Publish to File option or

you could end up overwriting (and losing) your projects. Because this release uses local files for storage then the limits on embedding a media file in a Passage is removed.

Module 2: HTML, Markdown, and Your First Website

Welcome to Module 2! Here you will begin to understand the basics of web development. By exploring the essentials of HTML and Markdown, you will soon be equipped with the skills needed to create and publish your very first website. You'll also discover how to showcase your Twine game on your own personal webpage.

In this module the goal is to get familiar with writing content on websites. We will look at two different types of markdown language. The first, HTML, is common across web pages. We will also look at using Markdown on GitHub.

By the end of this module, your achievements will include:

- Publishing your first website, whether it's a single or multiple pages, utilizing HTML or Markdown.
- Integrating your Twine game into a website, thus showcasing the practical application of your new coding skills.

HTML Basics

What is HTML?

HTML, or HyperText Markup Language, is the foundational language for crafting web pages. It provides a means for structuring content that browsers can interpret and display.

Key Concepts

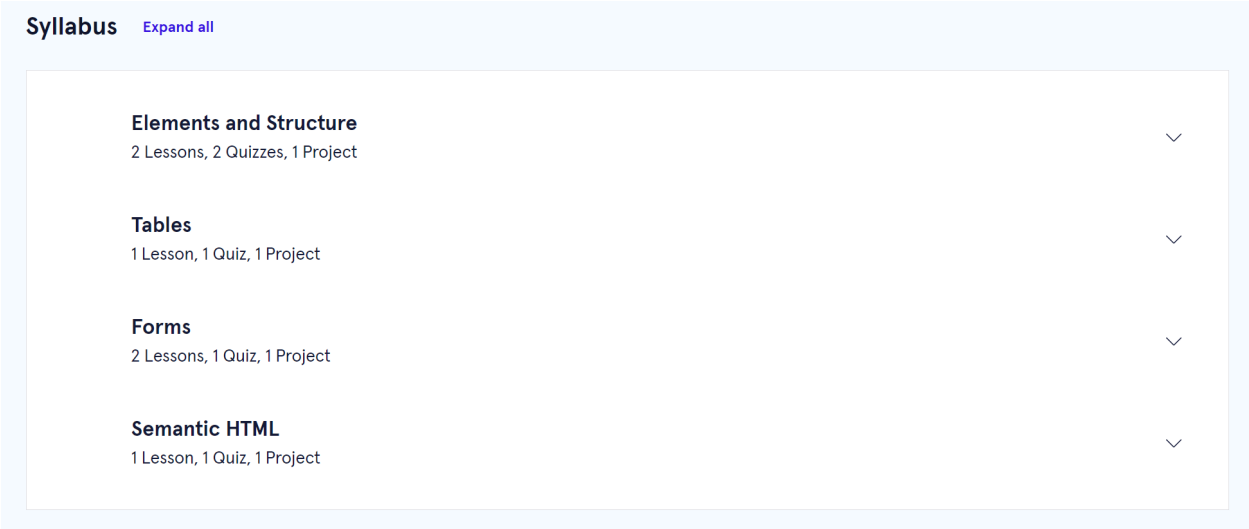
By studying HTML, you will learn how to:

- Define the structure of web pages using elements like headings, paragraphs, and lists.
- Add images and links to create interactive experiences.
- Layer your content for visual appeal and organizational clarity.

HTML can be used by itself to make a webpage but often is used combined with CSS and JavaScript which we will look at later in the course. To understand how the languages interact, read the following blog: Coding for Web Design 101: How HTML, CSS, and JavaScript Work.

Learning Resources

We recommend starting with the Code Academy 'Learn HTML' course, which is divided into four modules, as you can see in the screenshot below. Everyone should complete at least the introductory module on 'Elements and Structure', but you are very welcome to complete the remaining modules at your discretion, time, and enthusiasm. Note: You do need to log in before you can access this course.



Creating Your Own Website Using GitHub

Creating Your First Website

Now that you've learnt HTML you can try writing some yourself using an editor. Remember to start the file with the html tag. W3 schools is very useful as a guide to reminding yourself of HTML rules. Once saved as a .html file you should be able to click on the file from your computer and it should open in your browser. It is recommended that you save the main page of the website as index.html. From there you can see how it will appear in the final webpage and make edits. You can also put other files in the same folder such as pictures. Provided everything is embedded correctly it should appear. If you are feeling adventurous you can also create other HTML files for other pages of your website and link them.

Once you are happy then upload your file or files to GitHub and take a look at the following instructions to create a website.

Publishing Your Twine Game

You downloaded your Twine game and it consisted of a .html file, as well as maybe some other media. The next steps are simple! Rename the main HTML file of the Twine game downloaded to index.html. Now repeat the steps above and publish it as a website. This will make it easier to share with friends. Do make sure you are comfortable with everything you are publishing to the web however, as these websites will not be private.

Make sure you have fun! This is only a guide and doesn't cover everything. If you want to experiment with different themes or ideas, learn more and play around then please do and share what you find out on our Discord server. There's so many exciting things you can make with a bit of creativity and a good search engine!

Markdown Basics

What is Markdown?

Markdown is a lightweight markup language with a plain text formatting syntax. It is a tool used to convert text to HTML for easy content management. You can use it to write ReadMe pages about your repositories in GitHub as well as format GitHub pages. It is simpler and quicker to write than HTML, if you aren't looking to do anything too complicated.

Learning Resources

- We will also be following this interactive tutorial here.
- To understand more context of Markdown try this introductory video tutorial on YouTube.
- Use the blog Markdown Crash Course to remind yourself of the basics.

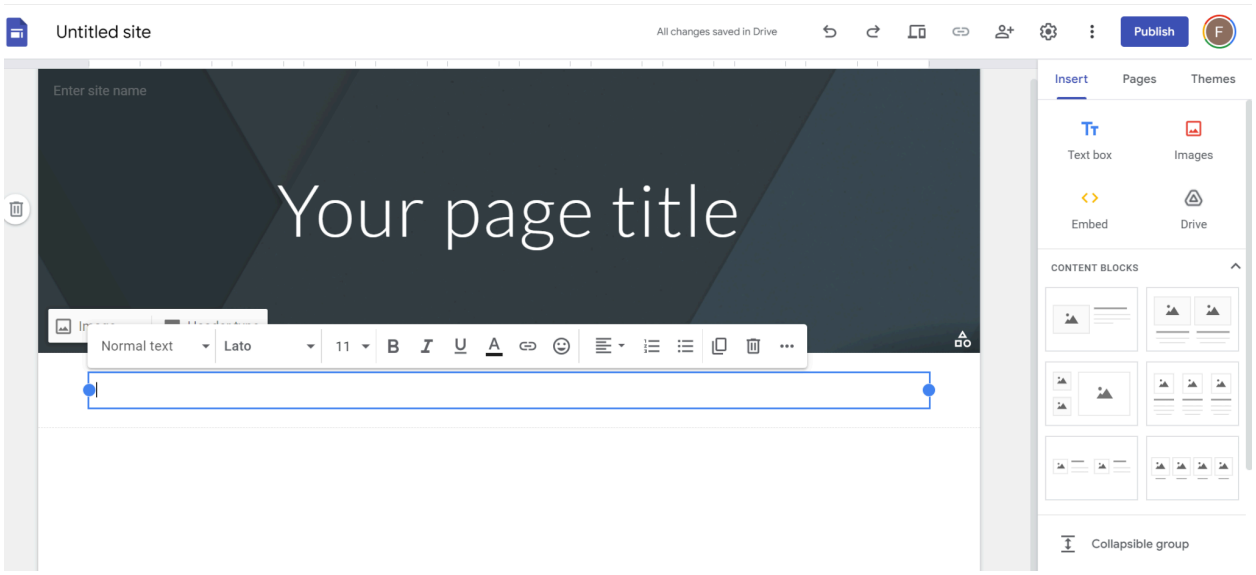
Next Steps

Once you've got the hang of Markdown, create some ReadMe.md files for your repositories to let people know what your repository is about. Learn more about readme files and why they are important here.

You can also try creating a new repository on your GitHub account and using the techniques you've learnt with GitHub pages try making a website using Markdown instead. You will need to upload to your repository either a readme.md file or an index.md file rather than the index.html you created earlier. Remind yourself with the previous guide.

Creating Your Own Website Using Google Sites

Here is where you can get creative! Google Sites is similar to Google Docs or Google Sheets except you get to publish a website instead. Google Sites provides a user-friendly, drag-and-drop interface for website building, allowing you to create a clean and responsive site without coding knowledge.



You can get as creative as possible without needing any coding languages. However, you may want to use it to host your new Twine game. Clicking the Embed button on the right hand side will allow you to embed your Twine game on the webpage by copying your HTML in. But you could also link between multiple Google pages, embed videos from YouTube (all YouTube videos have an embed button). Here are some instructions on embedding. Here is an example of an escape room style game created with Google pages. Play around or go further and create something to share! Don't forget to keep us posted in Discord.

Module 3: HTML with CSS

Module 3 will build on some of the fundamentals that you have already learnt such as HTML, GitHub and the basics of Twine. We will learn about the purpose of CSS and how it can be incorporated into Twine. We will also look at styling GitHub pages and templates. It will also take a quick look at accessibility.

Introduction to CSS

What is CSS?

CSS, which stands for Cascading Style Sheets, is the language we use to style a webpage. There is a great introduction to CSS provided here on the W3 Schools website. It is also useful to refer back to when remembering syntax, as a reference guide or reminder of the scope that CSS has. CSS is not just about changing colour but is also crucial when making websites accessible (for example readable on mobile and laptop).

Learning Resources

We recommend starting with the Code Academy 'Learn CSS' course. After completing this course it would be useful to also read this blog and maybe some of the learning resources at the end of it on accessibility in websites. If you still have time and enthusiasm after this, you will notice that there are other CSS courses available, either on different specialist topics or 'Learn Intermediate CSS' course. The Intermediate course on Code Academy will give you a more advanced overview but if you are interested in a specialist topic and it catches your eye then feel free to go in a different direction.

If you have time, see if you can improve or create something that you have learnt. There are lots of fun resources online that can make your website stand out, so you may want to explore these too. For example the Google Font corpus.

Note: You do need to log in to access Code Academy courses.

Incorporating CSS with Twine

To start incorporating CSS into your Twine Game we will be following the second and third parts of Adam Hammond's tutorial 'A Total Beginners Guide to Twine 2.1'. These are the sections titled 'Making Your Game Look Awesome with CSS' and 'Adding Videos and Music'.

GitHub and Jekyll

If you have been enjoying using Markdown and GitHub pages then you may be wondering a bit more about how CSS and styling may apply to these. To build on your work around creating pages in GitHub then this section of the GitHub documentation will help. Specifically around creating your own themes and incorporating CSS.

Final Goal for this Module

After working through the resources mentioned in this module, try and incorporate some of these in your game/website. Also try to document what you have used and added somewhere. Do not forget to share your updated games. It is fun to see the games we build grow over time as we learn more.

Module 4: Embedding and Introduction to JavaScript

Module 4 will take the websites that you have so far built and allow you to explore different pages and tools you can use online to make them more exciting. Such as adding video content, interactive exhibition software, tweets and games. We will also start looking at JavaScript, the coding language that is designed to make websites interactive.

Embedding and Tools

Not everything needs to be made from scratch when coding and embedding is a great way to bring other external resources from across the internet into your website. Read more about embedding and the embed tag here and how to add them here. These can also be used within Twine. The tools element in this module is simply a way of describing the range of different features and resources that can be used in embedding. This module will be about you discovering these and being creative about what you can now make. Here are a few ideas to get you started!

- Starting with something simple. You may wish to put posts from twitter elsewhere which is possible following these instructions.
- Or you may want to add a youtube video.
- Add a map to your website
- Or a personal favourite, an interactive exhibition or timeline
- There are also lists of games that are embeddable that can be found with a quick search

This is by no means an exhaustive list, there are lots of tools out there that you can use to enhance your websites and create something really unique. See what you can make with all these ideas!

Introduction to JavaScript

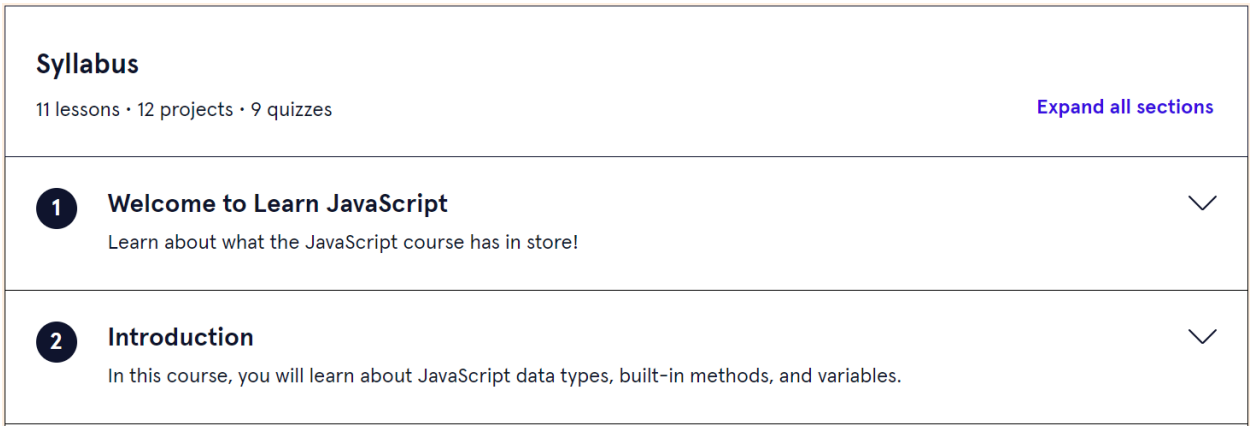
What is JavaScript?

'JavaScript is the programming language of the web' and also one of the most popular coding languages in the world. Many of the concepts in JavaScript follow the same principles as python and the same concepts of variables, loops, if and else statements etc.

can be applied in JavaScript coding language as well, it just has slightly different syntax and ways of achieving the same results.

Learning Resources

We recommend starting with the Code Academy 'Introduction to JavaScript' course. The W3 Schools introduction is also complimentary or can be used instead if you prefer that learning style. The next two modules will focus heavily on JavaScript from the same courses so you can take your time or jump ahead depending on how you wish to manage your workload. For now we recommend just the first couple of introductory modules.



If you have time, see if you can incorporate JavaScript into something that you have already learnt.

Note: You need to log in to access Code Academy courses.

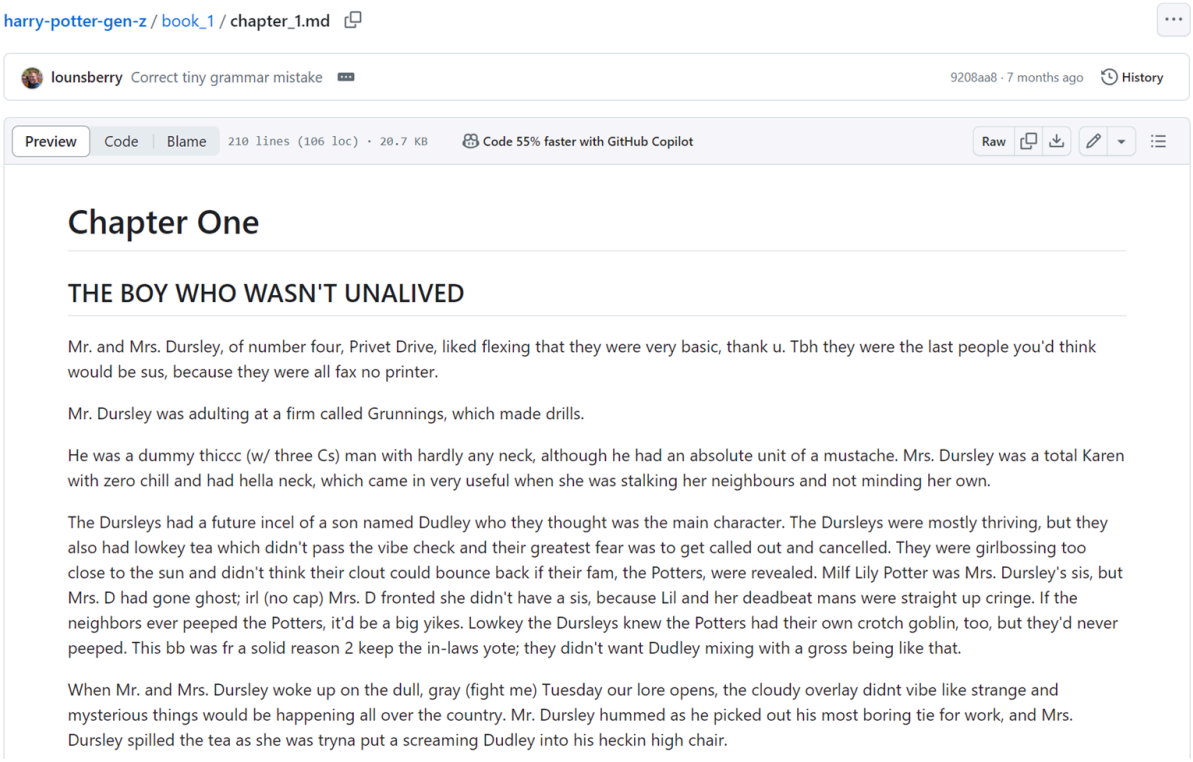
Final Goal Embedding and Tools

See what amazing frankenstenian websites you can now make. Either create a completely new website, this could be an exhibition, a game or any website you have in mind. Or you can embed and add some JavaScript functionality to something that you have already built!!! We would love you to share any ideas you have on what you can embed into websites, these are only just a few!

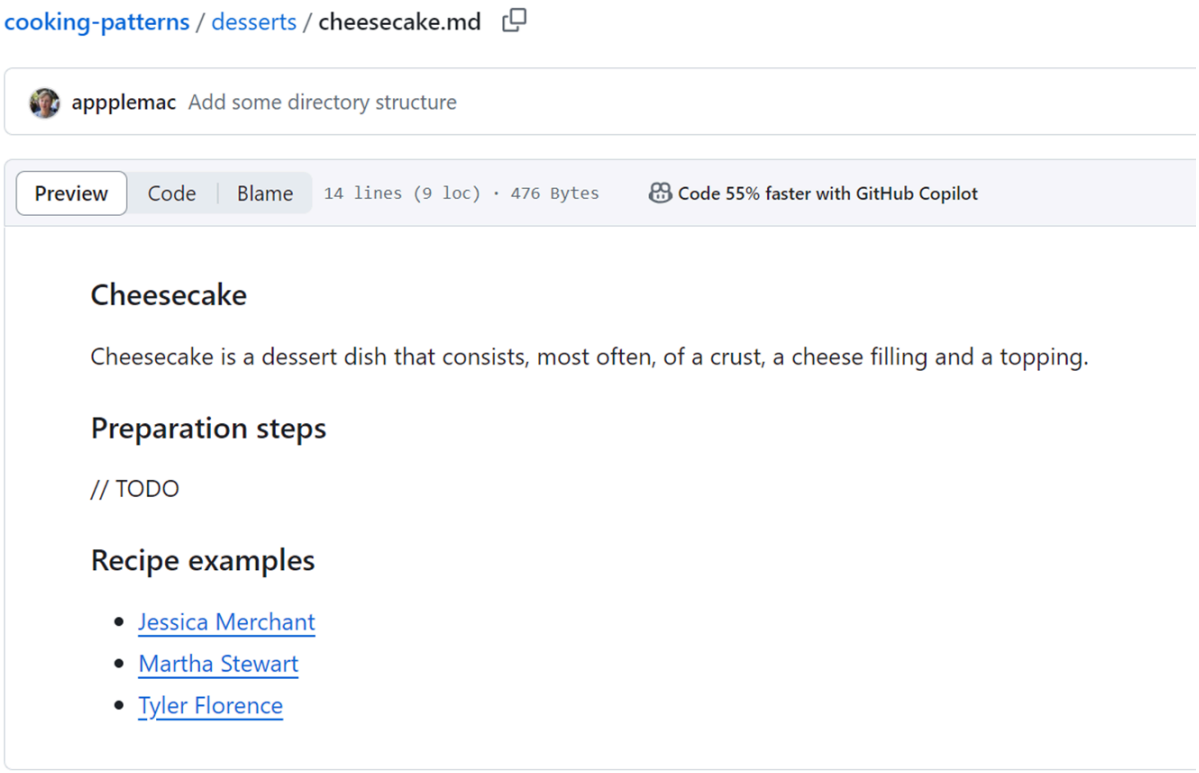
GitHub

GitHub is a platform that allows anyone to create, store, manage, and share content. Oftentimes this is code. But that is not the only thing present in GitHub. Here are some of the use cases for GitHub:

- Software development
- Fan fiction
- Zines
- Community/crowdsourcing projects
- Finding file samples
- Educational resources
- Reading lists
- Datasets
- Blog posts
- Documentation



Here you can see some fanfiction that is being written with a community. But you can also have recipes on there:



As digital archivists, we often use open-source tools to aid us with e.g. identifying and validating digital objects or harvesting websites. Here are some useful examples of GitHub repositories used in the field that you can find:

- https://github.com/openpreserve/jhove
- https://github.com/digital-preservation/droid
- https://github.com/webrecorder/browsertrix-crawler
- https://github.com/noord-hollandsarchief (see the repositories in this account)

So why is GitHub great to use? GitHub allows people to work collaboratively on any project, with tracking and control settings favouring non-linear workflows and multiple contributions at a time. By tracking everything, there is also a lot of data integrity. Additionally, you can use GitHub to let the developers know you have an issue/something is not working.

There are a lot of people working with digital archives that use GitHub. This is the reason that we found it very important for everyone to learn how to use it. Additionally, to show you that it is not just a scary programmer platform.

For this module, we want you to:

1. Create a GitHub account if you do not have one.
2. Follow this tutorial: https://github.com/skills/introduction-to-github
3. Try one of the following things with the Bits and Bots repository[1]:
    - Add your own game. You can place it in its own repository and link to it in the text.
    - Add something to the command prompt guide. You will notice it is a .md file. This means that it is written using MarkDown. You can use this website to see how it works and to practise.
    - Add a batch script to the command prompt folder.
    - Help us improve the repository. You can do this by raising an issue.
        - Issues are suggested improvements, tasks or questions related to the repository. Issues can be created by anyone (for public repositories), and are moderated by repository collaborators. Each issue contains its own discussion thread. You can also categorise an issue with labels and assign it to someone.
        - What makes a good issue is being very clear, adding detail to it and if possible, to provide a solution for it.

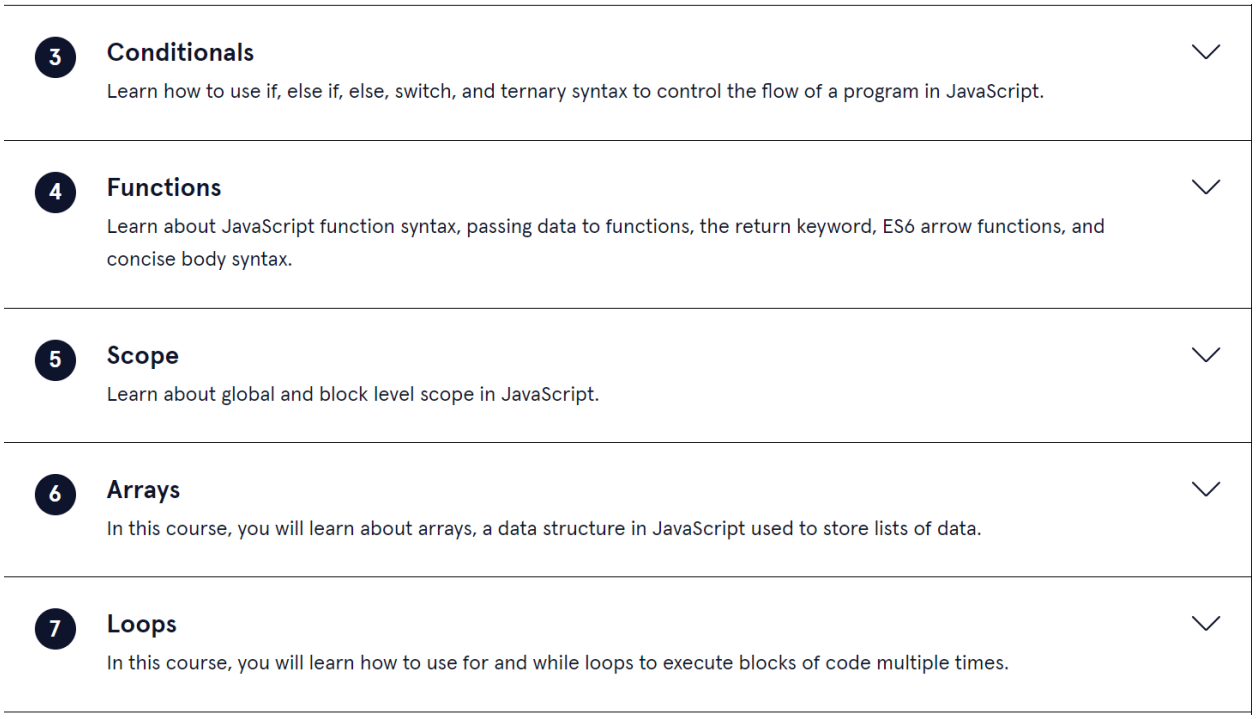
[1] We will also be using this repository to add some guides from the in-depth sessions here so you can add to these to make them more extensive.

Module 5: JavaScript

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.

Conditionals, Functionals, Arrays, and Loops

JavaScript has some similarities to Python. It also uses many of the same mechanisms such as loops. For this module we will be working our way through the rest of the Introduction to JavaScript course on Code Academy.



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 coming 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!

As this module is a little longer than the others it would be good to start considering what 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 the year.[2] 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?

Summertime Tips

Since this module has a somewhat longer timespan, here are some extra tips for the summer if you want more practice:

- Practising with GitHub

Practice with GitHub by placing all your games in either your own repository (and linking to it in ours) or placing them directly into the Bits and Bots repository. It is recommended that if you are hoping to make lots of changes to your game that you keep them within your own repository for better access and control.

- Adapt and improve

Try and adapt some games to make them directed to the field and add some extra functionality where you can.

- Look backwards

[2] 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.

Try looking at the work that you created in the earlier modules, do you want to change the functionality, add more colour, more interactivity, place it on it's own website?

- Adapt existing code

In the expert session by Remco van Veenendaal he demonstrated his box which was made by starting with code someone else made (the 3D box) and adapted it to his needs. Even adapting code can really help you get to grips with learning coding. In the future it can happen that you want to tweak something a bit more towards your own purpose instead of starting something from scratch.

Module 6: JavaScript / Advanced Twine

Objects, Modules and Classes

In this module we will look a bit closer at how data is stored in JavaScript by doing some mini courses in CodeAcademy on Objects as well as another on Modules and Classes. Objects are ways of storing multiple types of data, whilst classes are templates for objects. This means that you can store more complex data and extract it in your code. After completing the modules follow a simple tutorial on how to make a quiz in JavaScript using objects, as well as some of the previous concepts in the last module.

JavaScript in Twine

JavaScript can also be applied to Twine in various ways. You'll see lots of concepts such as arrays, variables, loops and functions come up in more complex Twine games. For instance you can now introduce variables into your game. There are two types of variables in Twine. A story variable that can be information stored throughout the entire game. For example the name a player inputs at the start of the game and is called throughout. Secondly there can be a temporary variable which is a variable that is introduced just for one passage of your twine game. You can use these concepts to decide the fate of a character depending on the choices they have made earlier in the story, or allow them to input data, or create little quizzes within your game.

Another thing to be aware of is that Twine like JavaScript has different types of data you can use in your code such as strings, null, boolean.

We will be following the fourth video in Adam Hammond's series on Twine to learn about incorporating variables and other concepts into your twine games. You can also follow Dan Cox's tutorials on introducing strings, objects and arrays. If you have further time or want to explore further he has created a range of video tutorials with examples of how to apply these concepts. I would recommend sticking to SugarCube unless you have a good reason not to. These videos are good as they provide specific examples of game dynamics that may be more relevant to what you are looking to create.

Final Goal

Add some of the concepts you have learnt in JavaScript to a twine game. Incorporate a short quiz, add input variables and see what else you can manage.

In JavaScript create your own quiz or another game that uses objects.

Module 7: Prototyping, User Testing and Design

This module is all about practical ideas for a final project. You may have thought of something that you wish to create by the end of the year or you may wish to keep going on lots of little projects. There are a few different areas for you to focus on in this module and at this stage it should be considered more guidance than official learning to be used as is most useful for your project or your work.

In last month's expert session we covered a brief introduction to UX and started defining what problems the products we are making solve and how they will solve them. A guide for which will be later made available. In this module we will look a bit more at UX but through the lens of user testing and prototyping. However if you want more of an overview of this subject area this is a good starting point for links and other resources.

Prototyping and User Testing

Coding can take a really long time and there isn't much point doing a lot of work only to find out later that that isn't actually what you wished to make in the first place! You also don't want to get to the end of the project and realise that it doesn't solve the problem that you had set out to solve. User testing and prototyping allows you to catch these issues early and change the plan before you've done lots of work on it.

Prototypes

Prototypes are quick unfinished versions of what you wish to finally create. They can contain key elements of the website to test on users to see if a concept works or be a full overview of the final product. You don't have to start with a screen to prototype. You can start by sketching your design out on paper, using post it notes and moving things around till you are happy with what you want to make. Prototyping using paper can also be useful for user testing or play testing anything you create before you start coding. When you want to try out your design on screen then it could be time to explore the concept of rapid prototyping.

User Testing

Assuming you have chosen to make a game as a final project. Taking advice from this excellent blog post by Matteo Menapace you could be asking:

- Is the game functional?
- Is the game fun?
- Is the game learnable?

Most of the creation of a game, or website, or almost any project is user testing. Here Matteo has chosen three things that are required for the games created. Fun, functional and learnable. But you could pick more things, less things and have an entirely separate list. By posing questions about the key things your project needs to do you can then begin to answer these during user testing.

> **Task:** ***write out the key goals of your project, and then a set of questions like the ones above that you wish to be answered when testing.***

The key is to not wait until it is perfect for others to try it out, the earlier that it can be tested the more time you have to improve your project and make it better.

Take a look at Matteo's blog post (which is also great for tips on creating speedy game concepts), this one on play testing of board games and also this practical guide to user testing. Some of what is being suggested is more complex than others and also for quite large scale projects, the principles, however remain the same. Try writing down your own user testing methodology and questions that match the time you have available and the scale of your project. Then give it a go and have fun with it!

Accessibility

Accessibility is a huge area when it comes to website design and there is a lot of in depth information online. It is also possible that your organisation will have their own set of criteria to be measured against when they develop their websites, and often these will be assessed externally as well. For government websites in the UK, for example, these have to meet the AA of Web Content Accessibility Guidelines. Which may be worth a read if you wish to learn more in depth about this. However, for your own personal website this is a list of simple things to consider when designing and building your website, or to change in an existing one. I would suggest using this list as a checker for any of your web projects. Though this guide will talk about three of these a little more in depth.

Responsive Websites

One tip you may want to look at for a final project is to create a page that looks good both on the laptop you are developing it on, a mobile device or on a big screen. There are a number of small ways you can alter your code to do this, and chances are that if you are modifying other code then it already has been designed to do this. Common tips include setting the widths to be responsive to the size of the screen, for example not a set number of px but as a percentage. This page from W3 schools lists many of the ways that you can modify CSS to make your webpage responsive.

Alternative Text

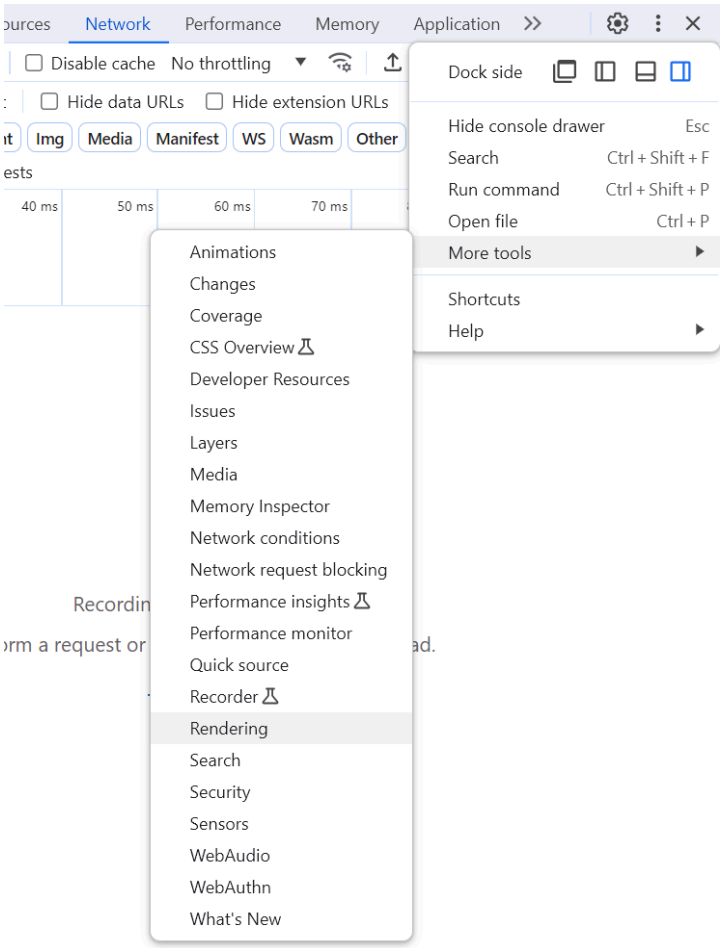
Adding alternative text to images on your website means if your images cannot be displayed or your user is using a screen reader, that the intention behind the design can still be understood. ALT text can be added to your HTML easily as an attribute, as described here.

Using the Inspect Tool

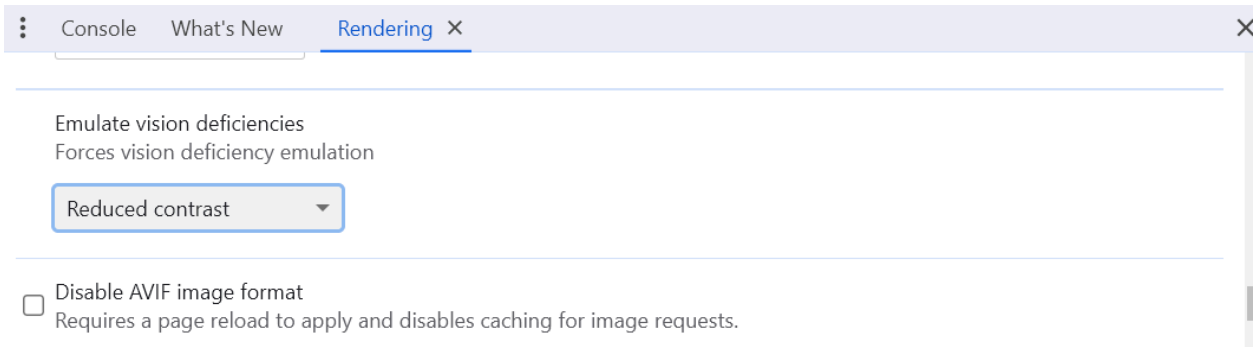
If you haven't used the F12 button yet to investigate websites then you should! You can inspect a webpage either by right clicking and pressing inspect or pressing the F12 button. This allows you to view the source code of a website, make edits to see immediate results and quite a few other hacks and tricks.

One use for the inspect tool is to render your webpage to emulate vision deficiencies or highlight areas that may not be suitable for people prone to photosensitive epilepsy.

On your website enter the inspect area and click on the three dots. Then navigate to 'More Tools', 'Rendering'



Once you have navigated here then at the bottom a selection of different rendering options become available.



This is a really helpful tool if you want to check that your colour choices, possibly chosen to stand out against each other, still do. Or that certain font choices can easily be seen.

There are lots of different features available with the inspect element. Look around and see what is most useful for you.

Delving Deeper

At this stage in the course you have tried a lot of different elements of website design! The three different languages, lots of tricks in Twine, saw how you can embed different parts of other websites and tested out different platforms to host your websites. Maybe you really enjoyed learning JavaScript and you would like to spend more time building games using it, some inspiration could be found here. Maybe you would like to use JavaScript to draw information using APIs into your website. Maybe you would like to create more complex Twine games. Or maybe you loved using CSS and would like to work further on that. This course has brushed the surface of many different subject areas to give a wide understanding of a few different things.

If you have a specific project in mind for the completion of the course or at work then at this point also you could start thinking 'Which of these skills would help me most with that?'