

Front-end Development

Module 6

JavaScript and Advanced Twine





Content

This module	2
Recap of Module 5	2
Using APIs with JavaScript.....	3
JavaScript in Twine	3
Final Goal	3
Introduction to APIs	4



This module

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.

Recap of Module 5

Like Python, JavaScript uses loops, conditionals, and arrays to control program flow and manage data. But unlike Python, JavaScript is designed to work seamlessly with HTML and CSS, making it ideal for building interactive web games. These games—like Magic Eight Ball, Rock Paper Scissors, or Whack-a-Mole—can be customized with bright colours and creative visuals. JavaScript allows you to combine logic with design, encouraging inventive gameplay. The final goal is to apply your skills to build and share a JavaScript-powered website. Collaboration, UX, and playtesting will become key focuses as you move toward a self-directed final project.



Using APIs with JavaScript

This month we will be focussing on APIs in our expert session. After you may wish to explore integrating these into your websites. You could for example, with a weather API, turn your website blue and animate it with rain in the event of rain. Or have it yellow in the event of sun. There are probably far better ideas than that as well!

After the expert session you may wish to take a look at some of the JavaScript code that can be used to call an API and how it can be used there are some good examples [here](#). If you think that you wish to use an API, are not sure which API and don't want it to be one about weather then a list of over 800 can be [found here](#) categorised.

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.



Introduction to APIs

This month's expert session will focus on building and using APIs with Python. You'll learn what an API is, how to request data from an API in Python, and how to create your own basic API. You'll also explore how to integrate APIs into your website using JavaScript. Exercises for both Python and front-end development will be shared and explained during the session, and later made available on our channels.

APIs—short for *Application Programming Interfaces*—are sets of rules that allow different software applications to interact. They act as messengers between systems, enabling one program to request data or services from another. For example, when a website displays weather updates, it likely uses an API to fetch real-time data from a weather service.

Here are some key reasons APIs are useful:

- **Automation:** APIs allow programs to perform tasks automatically, such as retrieving data, sending emails, or processing payments, without manual intervention.
- **Integration:** They enable different systems and platforms to work together. For instance, a website can use APIs to connect with social media platforms, payment gateways, or databases.
- **Scalability:** APIs help developers build modular systems that can grow and evolve. You can update or replace parts of a system without affecting the whole, making maintenance easier.
- **Efficiency:** Instead of building every feature from scratch, developers can use APIs to access existing services—saving time and resources.
- **Security:** APIs can control access to data and services, ensuring that only authorized users or systems can interact with sensitive information.

Understanding how APIs work and how to use them effectively is a vital skill for anyone building modern applications or websites.