**W3D1 - Asynchronous JS Lesson Learning Objectives**

Below is a complete list of the terminal learning objectives for this lesson. When you complete this lesson, you should be able to perform each of the following objectives. These objectives capture how you may be evaluated on the assessment for this lesson.

* Identify JavaScript as a language that utilizes an event loop model
* Identify JavaScript as a single threaded language
* Describe the difference between asynchronous and synchronous code
* Execute the asynchronous function setTimeout with a callback.
* Given the function function asyncy(cb) { setTimeout(cb, 1000); console.log("async") } and the function function callback() { console.log("callback"); }, predict the output of asyncy(callback);
* Use setInterval to have a function execute 10 times with a 1 second period. After the 10th cycle, clear the interval.
* Write a program that accepts user input using Node’s readline module

**W3D1 - Quizes**

* [Callbacks Recall](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/callbacks-quiz-recall)
* [Async](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/async-quiz)

**W3D2 - Node.js Learning Objectives**

Below is a complete list of the terminal learning objectives for this lesson. When you complete this lesson, you should be able to perform each of the following objectives. These objectives capture how you may be evaluated on the assessment for this lesson.

* Define NodeJS as distinct from browser based JavaScript runtimes.
* Write a program that reads in a dictionary file using node's FS API and reads a line of text from the terminal input. The program should 'spell check' by putting asterisks around every word that is NOT found in the dictionary.

**W3D2 - Git Lesson Learning Objectives**

Below is a complete list of the terminal learning objectives for this lesson. When you complete this lesson, you should be able to perform each of the following objectives. These objectives capture how you may be evaluated on the assessment for this lesson.

* Use Git to initialize a repo
* Explain the difference between Git and GitHub
* Given 'adding to staging', 'committing', and 'pushing to remote', match attributes that apply to each.
* Use Git to clone an existing repo from GitHub
* Use Git to push a local commit to a remote branch
* Use git to make a branch, push it to github, and make a pull request on GitHub to merge it to master
* Given a git merge conflict, resolve it
* Match the three types of git reset with appropriate descriptions of the operation.
* Use Git reset to rollback local-only commits.
* Identify what the git rebase command does
* Use git diff to compare a local 'staging' branch and 'master' branch.
* Use git checkout to check out a specific commit by commit id

**W3D2 - Quizes**

* [Async Recall](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/async-quiz-recall)
* [Git Rebase](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/git-rebase-quiz)
* [Git Actions](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/git-actions-quiz)

**W3D3 - Command Line Interface Basics Lesson Learning Objectives**

Below is a complete list of the terminal learning objectives for this lesson. When you complete this lesson, you should be able to perform each of the following objectives. These objectives capture how you may be evaluated on the assessment for this lesson.

* Given a folder structure diagram, a list of 'cd (path)' commands and target files, match the paths to the target files.
* Create, rename, and move folders using unix command line tools.
* Use grep and | to count matches of a pattern in a sample text file and save result to another file.
* Find what -c, -r, and -b flags do in grep by reading the manual.
* Identify the difference in two different files using diff.
* Open and close nano with and without saving a file.
* Use ‘curl’ to download a file.
* Read the variables of $PATH.
* Explain the difference between .bash\_profile and .bashrc.
* Create a new alias by editing the .bash\_profile.
* Given a list of common scenarios, identify when it is appropriate and safe to use sudo, and when it is a dangerous mistake.
* Write a shell script that greets a user by their $USER name using echo.
* Use chmod to make a shell script executable.

**W3D3 - Quizes**

* [Navigation With CD](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/navigation-with-cd-quiz)
* [Dotfiles](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/dotfiles-quiz)
* [Safety With Sudo](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/safety-with-sudo-quiz)

**W3D4 - JS Trivia Lesson Learning Objectives**

Below is a complete list of the terminal learning objectives for this lesson. When you complete this lesson, you should be able to perform each of the following objectives. These objectives capture how you may be evaluated on the assessment for this lesson.

* Given a code snippet of a unassigned variable, predict its value.
* Explain why functions are “First Class Objects” in JavaScript
* Define what IIFEs are and explain their use case
* (Whiteboarding) Implement a closure
* Identify JavaScript’s falsey values
* Interpolate a string using back-ticks
* Identify that object keys are strings or symbols
* A primitive type is data that is not an object and therefore cannot have methods(functions that belong to them).
* Given a code snippet where variable and function hoisting occurs, identify the return value of a function.

**W3D4 - Quizes**

* [Primitive Data Types](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/primitive-data-types-quiz)
* [Scope Recall](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/scope-quiz-recall)
* [Object Key](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/object-key-quiz)
* [IIFE](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/iife-quiz)
* [Function Hoisting](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/function-hoisting-in-javascript-quiz)
* [Context Recall](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/context-quiz-recall)
* [Predicting Variable Evaluations](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/predicting-variable-evaluations-quiz)
* [Falsey Values](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/falsey-values-in-javascript-quiz)
* [Variable Recall](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/variable-quiz-recall)
* [First Class Objects](https://open.appacademy.io/learn/js-py---jun-2020-online/week-3-jun-2020-online/first-class-objects-in-javascript-free-response)