**Lesson 3 - Rock, Paper, Scissors:**

| **Title of Unit** | Foundations | **Grade Level** | 11-12 |
| --- | --- | --- | --- |
| **Subject** | Mobile App Development | **Time Frame** |  |
| | **Description** | Here we finally dig into JavaScript and learn how to make the web dynamic. | | --- | --- | | | | |
| **Stage 1 - Identify Desired Results** | | | |
| **Learning Outcomes**  What relevant goals will this unit address? | | | |
| Computer and Information Sciences, General.  **CIP#**: 11.0101  Pathway Competencies   * **Algorithms & Programming**: | | | |
| **Enduring Understandings** | | **Essential Questions** | |
| *Students will understand…*   * *What variables are.* * *What the comparison operators are.* * *What conditionals are.* * *What nesting is.* * *What truthy and falsy values are.* * *What can you do to help get a clearer understanding of the problem?* * *What are some of the things you should do in the planning stage of the problem solving process?* * *What is an algorithm?* * *What is pseudo code?* * *What are the advantages of breaking a problem down and solving the smaller problems?* | | *Content specific….*   * *Name the three ways to declare a variable?* * *Which of the three variable declarations should you avoid and why?* * *What rules should you follow when naming variables?* * *What should you look out for when using the + operator with numbers and strings?* * *How does the % operator work?* * *Explain the difference between == and ===.* * *When would you receive a NaN result?* * *How do you increment and decrement a number?* * *Explain the difference between prefixing and post-fixing increment/decrement operators.* * *What is operator precedence and how is it handled in JS?* * *How do you access developer tools and the console?* * *How do you log information to the console?* * *What are the eight data types in JavaScript?* * *Which data type is NOT primitive?* * *What is the relationship between null and undefined?* * *What is the difference between single, double, and backtick quotes for strings?* * *What is the term for embedding variables/expressions in a string?* * *Which type of quote lets you embed variables/expressions in a string?* * *How do you embed variables/expressions in a string?* * *How do you escape characters in a string?* * *What are methods?* * *What is the difference between slice/substring/substr?* * *What are the three logical operators and what do they stand for?* * *What are the comparison operators?* * *What are truthy and falsy values?* * *What are the falsy values in JavaScript?* * *What are conditionals?* * *What is the syntax for an if/else conditional?* * *What is the syntax for a switch statement?* * *What is the syntax for a ternary operator?* * *What is nesting?* * *What are the three stages in the problem solving process?* * *Why is it important to clearly understand the problem first?* * *What are three reasons why you may see a TypeError?* * *What is the key difference between an error and a warning?* * *What is one method you can use to resolve an error?* | |
| **Knowledge:** | | **Skills:** | |
| *Students will know how to...*   * *How do you declare a variable?* * *What are three different ways to declare a variable?* * *Which one should you use when?* * *What are the rules for naming variables?* * *What are operators, operands, and operations?* * *What is concatenation and what happens when you add numbers and strings together?* * *What are the different types of operators in JavaScript?* * *What is the difference between == and ===?* * *What are operator precedence values?* * *What are the increment/decrement operators?* * *What is the difference between prefixing and post-fixing them?* * *What are assignment operators?* * *What is the “Unary +” Operator?* * *Explain the three steps in the problem solving process.* * *Explain what pseudocode is and be able to use it to solve problems.* | | *Students will be able to…*   * *Name the eight data types in JavaScript.* * *Understand the difference between single, double, and backtick quotes.* * *Embed a variable/expression in a string.* * *Define what a method is.* * *Name the three logical operators.* * *Be able to break a problem down into subproblems.* * *Name at least three kinds of Javascript errors* * *Identify two parts of an error message that help you find where the error originates* * *Be able to understand how to research and resolve errors* | |

| **Stage 2 – Assessment Evidence** | | |
| --- | --- | --- |
| **Performance Task** | | |
| **PROJECT: ROCK PAPER SCISSORS****Introduction** We’re going to make a simple implementation of grade-school classic “rock paper scissors”. If you don’t know what that is, check the [Wikipedia article](https://en.wikipedia.org/wiki/Rock%E2%80%93paper%E2%80%93scissors) or [this](https://www.wikihow.com/Play-Rock,-Paper,-Scissors) ridiculous step-by-step. For the moment we’re just going to play the game from the browser console, but we will revisit this project in a later lesson and add a Graphical User Interface with buttons and text, so don’t forget to keep the code on GitHub! You might notice some ‘Live Preview’ links in the student solutions that have a GUI - this is coming in a later lesson. When you get there don’t forget to come back and add your link! | | |
| **Other Evidence** | | **Student Self-Assessment** |
| * Practice * Assignment/exercise * Knowledge Checks | | * Reflection |

| **Stage 3 – Learning Plan** | | | | |
| --- | --- | --- | --- | --- |
| * Activity 1 - Javascript Fundamentals Part 1 * Activity 2 - Javascript Fundamentals Part 2 * Activity 3 - Developer Tools 2 * Activity 4 - Javascript Fundamentals Part 3 * Activity 5 - Problem Solving * Activity 6 - Understanding Errors * Project 3: Rock Paper Scissors | | | | |
| **How will you engage students at the beginning of the unit? (motivational set)** | | | | |
| **Variables** You can think of variables as simply “storage containers” for data in your code. Until recently there was only one way to create a variable in JavaScript — the var statement. But in the newest JavaScript versions we have two more ways — let and const.   1. [This variable tutorial](http://javascript.info/variables) will explain everything you need to know! Be sure to do the Tasks at the end. Information won’t stick without practice!   The above tutorial mentioned this, but it’s important enough to note again: let and const are both relatively new ways to declare variables in JavaScript. In *many* tutorials (and code) across the internet you’re likely to encounter var statements. Don’t let it bother you! There’s nothing inherently wrong with var, and in most cases var and let behave the same way. But sometimes the behavior of var is *not* what you would expect. Just stick to let (and const) for now. The precise differences between var and let will be explained later. **Numbers** Numbers are the building blocks of programming logic! In fact, it’s hard to think of any useful programming task that doesn’t involve at least a little basic math… so knowing how numbers work is obviously quite important. Luckily, it’s also fairly straightforward.   1. [This W3Schools lesson](https://www.w3schools.com/js/js_numbers.asp) followed by [this one](https://www.w3schools.com/js/js_arithmetic.asp), are good introductions to what you can accomplish with numbers in JS. 2. [This MDN article](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Math) covers the same info from a slightly different point of view. There’s much more that you can do with numbers, but this is all you need at the moment. 3. Read through (and code along with!) [this article](http://javascript.info/operators) about operators in Javascript. Don’t forget to do the “Tasks” at the bottom of the page! It will give you a pretty good idea of what you can accomplish with numbers (among other things!) in JavaScript. | | | | |
| **#** | **Lesson Title** | **Lesson Activities** | **CCCs** | **Resources** |
| 1 | Javascript Fundamentals Part 1 | 1. [This W3Schools lesson](https://www.w3schools.com/js/js_numbers.asp) followed by [this one](https://www.w3schools.com/js/js_arithmetic.asp), are good introductions to what you can accomplish with numbers in JS. 2. [This MDN article](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Math) covers the same info from a slightly different point of view. There’s much more that you can do with numbers, but this is all you need at the moment. 3. Read through (and code along with!) [this article](http://javascript.info/operators) about operators in Javascript. Don’t forget to do the “Tasks” at the bottom of the page! It will give you a pretty good idea of what you can accomplish with numbers (among other things!) in JavaScript.  **Practice!** Try the following exercises:   1. Add 2 numbers together! (just type console.log(23 + 97) into your html file) 2. Add a sequence of 6 different numbers together. 3. Print the solution to the following equation: (4 + 6 + 9) / 77    1. Answer should be approximately 0.24675 4. Let’s use variables!    1. Type the following at the top of the script tag: let a = 10    2. console.log(a) should print 10    3. Try the following: 9 \* a    4. and this: let b = 7 \* a (returns undefined) and then console.log(b) 5. You should be getting the hang of this by now… try this sequence:    1. Declare a constant variable max with the value 57    2. Set another variable actual to max - 13    3. Set another variable percentage to actual / max    4. If you type percentage in the console and press enter you should see a value like 0.7719 6. Take a few minutes to keep playing around with various things in your script tag. Eventually, we will learn how to actually make those numbers and things show up on the webpage, but all of this logic will remain the same, so make sure you’re comfortable with it before moving on. |  |  |
| 2 | Javascript Fundamentals Part 2 | **Practice** To give you a good bit of practice, we have created some repl.it exercises for you to play with. We believe that it’s best to practice programming on your *own* computer, rather than in an online environment, but we’ll get to that soon enough.  Be sure to do the lessons in order presented here. Pressing “run” at the top will run the code. Read all directions, watch the terminal, and read all the errors. Don’t forget to use ‘console.log’ extensively.  To get started, create a free repl account and click “Fork '' in order to have access to the exercises. Note: Feel free to browse the files on the left column in order to gain familiarity with it.   * [Exercise 1](https://repl.it/@I3uckwheat/troubleshooting#troubleshooting.js)   + In this exercise, you will be working out of the file called troubleshooting.js * [Exercise 2](https://repl.it/@I3uckwheat/enter-a-number#script.js)   + You will be working out of script.js * [Exercise 3](https://repl.it/@I3uckwheat/lets-do-some-math#math.js)   + You will be working out of math.js * [Exercise 4](https://repl.it/@I3uckwheat/direction-follow#follow.js)   + You will be working out of follow.js |  |  |
| 3 | Developer Tools 2 | **Assignment**  1. Head to the [Chrome DevTools Documentation](https://developers.google.com/web/tools/chrome-devtools/) by Google. The following subsections cover what you’ll be using the Developer Tools for 95% of the time. Feel free to skip the elements you are already familiar with:    * Open DevTools    * CSS      + View and change CSS      + CSS features reference    * Console      + Console overview    * Mobile Simulation      + Simulate mobile devices with Device Mode    * Get Started With Viewing And Changing The DOM    * JavaScript      + Debug JavaScript      + Pause your code with breakpoints 2. Then, watch [this video](https://www.youtube.com/watch?v=JzZFccCEgGA) by The Net Ninja for more detail on using the JavaScript Console. |  |  |
| 4 | Javascript Fundamentals Part 3 | **Practice** Let’s write some functions! Write these in the script tag of a skeleton HTML file. If you’ve forgotten how to set it up, review the instructions from fundamentals 1.  For now, just write each function and test the output with console.log.   1. Write a function called add7 that takes one number and returns that number + 7. 2. Write a function called multiply that takes 2 numbers and returns their product. 3. Write a function called capitalize that takes a string and returns that string with *only* the first letter capitalized. Make sure that it can take strings that are lowercase, UPPERCASE or BoTh. 4. Write a function called lastLetter that takes a string and returns the very last letter of that string:    1. lastLetter("abcd") should return "d" |  |  |
| 5 | Problem Solving | **Assignment**  1. Read [How to Think Like a Programmer - Lessons in Problem Solving](https://www.freecodecamp.org/news/how-to-think-like-a-programmer-lessons-in-problem-solving-d1d8bf1de7d2/) by Richard Reis. 2. Watch [How to Begin Thinking Like a Programmer](https://www.youtube.com/watch?v=azcrPFhaY9k) by Coding Tech. It’s an hour long but packed full of information and definitely worth your time watching. 3. Read this [What is Pseudo Coding](https://www.vikingcodeschool.com/software-engineering-basics/what-is-pseudo-coding) article from The Viking Code School. |  |  |
| 6 | Understanding Errors | **Exercise:** Work through [this lesson on MDN](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_went_wrong). Be sure to download their starter code that has intentional errors. |  |  |
| P3 | Rock Paper Scissors | **Assignment**  1. Start a new Git repo for your project. 2. Create a blank HTML document with a script tag (Hint: it is best practice to link an external .js file). This game is going to be played completely from the console, so don’t worry about putting anything else in there. 3. Your game is going to play against the computer, so begin with a function called computerPlay that will randomly return either ‘Rock’, ‘Paper’ or ‘Scissors’. We’ll use this function in the game to make the computer’s play. *Tip: use the console to make sure this is returning the expected output before moving to the next step!* 4. Write a function that plays a single round of Rock Paper Scissors. The function should take two parameters - the playerSelection and computerSelection - and then return a string that declares the winner of the round like so: "You Lose! Paper beats Rock"    1. Make your function’s playerSelection parameter case-insensitive (so users can input rock, ROCK, RocK or any other variation). 5. Important note: you want to return the results of this function call, *not* console.log() them. To test this function console.log the results: function playRound(playerSelection, computerSelection) { // your code here! } const playerSelection = "rock"; const computerSelection = computerPlay(); console.log(playRound(playerSelection, computerSelection)); 6. Write a NEW function called game(). Use the previous function *inside* of this one to play a 5 round game that keeps score and reports a winner or loser at the end.    1. You have not officially learned how to “loop” over code to repeat function calls… if you already know about loops from somewhere else (or if you feel like doing some more learning) feel free to use them. If not, don’t worry! Just call your playRound function 5 times in a row. Loops are covered in the next lesson.    2. At this point you should be using console.log() to display the results of each round and the winner at the end.    3. Use prompt() to get input from the user. [Read the docs here if you need to.](https://developer.mozilla.org/en-US/docs/Web/API/Window/prompt)    4. Feel free to rework your previous functions if you need to. Specifically, you might want to change the return value to something more useful.    5. Feel free to create more “helper” functions if you think it would be useful. |  |  |

| **Stage 4 - Assess and Reflect** | |
| --- | --- |
| **Considerations** | **Comments** |
| **Is there alignment between outcomes, performance assessment and learning experiences?** |  |
| **Have I made purposeful adjustments to the curriculum content (not outcomes), instructional practices, and/or the learning environment to meet the learning needs and diversities of all my students?** | For struggling students:                    For students who need a challenge: |
| **Do I use a variety of teacher directed and student centered instructional approaches?** |  |
| **Do the students have access to various resources on an ongoing basis?** |  |
| **Have I nurtured and promoted diversity while honoring each child’s identity?** |  |