|  |  |  |  |
| --- | --- | --- | --- |
| Sub-Section # | Assignment Title | The task | Assignment outline |
| 1.8 | "Create an Object using Object Initializers" | * Create an object, * give it properties, & * change a property * ?Use ‘if’ statement on naughty/nice? | Santa needs to digitize his user files, and assign elves to them. There may be some re-assignments based on workload, changes in “naughty/nice” status etc. Create a template file using Object Initializers   * Assigned Elf : variable (with name stored) * First Name, Last Name : array / string * Naughty/Nice: boolean * TopThreeWishes: array of strings * NoCPPY (Number of cookies provided previous year): number |
| 1.9 | "Create an Object with a Constructor Function" | * Create an object with properties * Create four instances with values - at least one of whom has been bad * Add an address property, and store its value as an array of one or more strings * Use an ‘if’ statement on naughty/nice | Santa wants to know if you’re ready to move forward on the project. Create a few user files for potential stops in late December, and add an address property. Then, compile a list based on who has been nice (and will be getting a visit). Note: at least one child 1. Must be designated as “bad” (using a boolean value) and also have provided 0 cookies last year - though representatives apparently can neither confirm nor deny any connection |
| 2.4 | "Accessing Nested Object Parameters" | * Convert the address property, and the top wishes property to nested objects (Line1, Line 2 etc & Top, Second etc) * For one child, it is discovered that they have separated parents, so convert their address into an array of two address objects * Find info using conditional chaining | El capitan Kringle is hours from take-off, but the head admin elf has received last-minute updates (apparently, their surveillance network is 24/7) and you need to do emergency sort and filing. |
| 2.5 | "Create a Nested Object and Iterate" | * Create nested object * Iterate object to find all keys * Create log-to-console instructions for all keys | Either use a text file provided by us (see previous assignment-comment idea) or  You’ve been hired by a new tech startup, and want you to introduce yourself to your new teammates by creating a personal object. You should detail three things you’re passionate about - say, the parts of your ideal PC, the players on your favorite team, and the characters of your favorite movie. List these as sub-objects, as nested objects, and then provide for loops for each in your code file so that teammates can easily read them!  Note: In theory, your key:values should be things like [cpu:”Intel”, gpu:”EVGA Geforce 3090”] of nested-object idealPC; [forward:”Cristiano Ronaldo” ] of object Manchester United F.C) and [title:”Matrix”, protagonist:”Neo” of favMovie) etc  Also, feel free to do other lists that reflect you better! |
| 3.8 | "Declare & Call a Function" |  |  |
| 3.9 | "Functions v Methods" |  |  |
| 4.7 | "Nested Functions" |  |  |
| 4.8 | "More Complex nested functions" |  |  |
| 4.9 | "Using Scope Variables" |  |  |
| 5.6 | "Function w/ a Timeout" |  |  |
| 5.7 | "Factorials" |  |  |
| 6.6 | "Create a JS Pattern" |  |  |
| 7.5 | "Create a Pattern - Rewrite a Given Code" |  |  |
| 8.5 | “JS in Strict Mode” |  |  |
| 8.6 | “Find & Fix Errors in Code” |  |  |
| 9.2 | "Exit Project" |  |  |