Javascript tutorial for beginners pdf

If you like this come check out my page www.brian-best.com for more blogs! The following blog is apart of my growing list of guides for teachers and students of online tutorials in web technology. To help assess what a tutorial expects of the reader to know before taking. This time we're talking about JavaScript, but check out my other guides for HTML and CSS. JS is a big one, growth in it has only exploded in the last ten years. What used to be a system to add a splash of logic to a page can now run entire websites, hardware and 3D games. For this guide, I only cover Vanilla JS as it is available on the front in a browser. Later I'll do other blogs to include more advanced features including various frameworks like React, Vue, as well as the backend side of JavaScript with Node.js.For this scale I will use Beginner, Basic, Intermediate, and Advanced to specify what prior knowledge a student should know before taking on a tutorial. For each level, the student should know or have:Beginneringn's knowledge of JavaScriptBasic JavaScriptintermediate knowledge of both HTMLand CSSvariables, and how to declare them operators, such as +, -, etc.functionsWhat an API isHow to access and query domIntermediate JavaScriptLoopsCreate and reproduce new elements on the DOM. How and when JS is rendered on a web pageEvents. How to listen for events and create your own. Function and variable scopes, you should know what this isAnonymous features and Immediately Invoked Function Expression (IIFE)Browser support for new JS features and how to use polyfills to fill in support when neededAdvanced JavaScriptHow to make an AJAX requestHow to use promises and know when to get data. The differences in where, allow and const to declare variables. What closures are and how to utilize themWhat Object Oriented Programming, or OOP, is and how it can happen in JavascriptOptimization of logic for speedThis blog is just a guideline, of course, there are plenty of cases where some overlap of knowledge will be required. Examples are a basic tutorial that requires some intermediate knowledge. For students, just remember googling for terms that you don't fully understand are encouraged. For teachers, it's not a bad thing to go into more detail on what you think might be necessary knowledge. Join Hacker Noon Create your free account to unlock your personalized reading experience. Through Jackie Lohrey Quicken financial management software, having all its choices, instructions and options can be overwhelming, especially for a new user. Everything seems just as important, and everything seems like you need to master it right now. Before you dive in and start the process of using accounts, creating reports and filling in budget amounts, it's important to learn some basic concepts that can help you optimize Quicken for your individual needs. One of the most important concepts to understand is how to categories and tags to track your account transactions. A Quicken category is a label, label, as Daily Consumer Goods or Households, it defines a broad group of both income and expense transactions. You use a single category per transaction to track how you earn and spend money. For example, all purchases of groceries, no matter where you buy them, go under the category of Groceries and all income, regardless of source, goes into the category Income. Use a subcategory to provide larger organization and detail within a category and to help create a budget. This is especially useful in a category like Auto, where you incur multiple types of expenses. Creating fuel, maintenance, and registration subcategories makes it much easier to see where your money is going. Tags are like keywords. They allow you to track transactions within and between category groups. For example, to track the money you spend on incidentals for a child in college, create a tag with the name of the child, and then apply it to the appropriate category or subcategory, such as Groceries, Auto: Fuel, or Phone: Mobile Phone. Quicken comes with many built-in categories that cover many standard items. Not all built-in categories are visible from the Category List window, because much of what you see depends on information you provide to Quicken during booting. Before you create a new category, access Quicken's built-in categories and see if anything appropriate is already there. To access built-in categories, select Tools from the main menu, and then select Category List. Click the Add Categories button, select the categories you want to add from the menu on the left side of the screen, and click the Add button. Click OK to exit and return to the List categories. To add a new category or subcategory, click the New button in the Categories List window, create a name for the category or subcategory, provide an optional description, and specify category type, such as Income, Expense, or Subcategory. If you select Subcategory, use the drop-down list next to the selection to enter the main category name. To create a new tag, select Tools from the main menu, and then select Tag List. Click the New button at the top of the window and type in a name and description of the tag. Ignore the Copy number box that applies to track rental items only. Click OK to exit. While it's possible to create categories, subcategories, and tags on the fly, this isn't always a good option, especially if you're frustrated or simply trying to make it fit. Putting thought into the categories, subcategories and tags you use can make Quicken work better and more efficiently. Use categories, subcategories, and tags within trade repositories to identify and organize revenue and expenses. For example, in a checking account table, after entering the payee and the amount of the check, click the drop-down list under the payee's name to select either the category or subcategory of the transaction. Tab over or click in the next box to select the appropriate Tag from from Listbox. At the end of the month, run a report that will show an income/expensebreakdown by category, subcategory, and tag. For example, to run a regular Quicken expense report, click Reports from the main menu. Select Expenses, and then select Expenses by Category to view the report and the included pie chart. The expense report is the default to a note so far this year, so adjust date parameters to your needs. Introduction to JSON: A Complete JSON Tutorial series for beginnersJavaScript ObjectEmployment that is commonly known as JSON is one of the most popular data transition formats. It is a text-based and easy format for data transactions. JSON format was first calculated by Douglas Crockford.This is a text-based format is easier to read or write by the user and at the same time, its lightweight property makes it a stress-free option for machines to deconstruct or generate. It's basically a subset of JavaScript but JSON, as a text format is completely independent of any of the programming languages used as almost any language, can easily analyze the text. Its unique features like text-based, light, language independent etc. make it an ideal candidate for the data-interchange operations.**********LIST of JSON Tutorials in this series:Tutorial #1: Introduction to JSON (This tutorial) Tutorial #2: Create JSON items using C # Tutorial #3 Example: Create J SON Structure Using C# Tutorial #4: Using JSON for Interface Testing Tutorial #5: JSON Interview Questions **********properties, use and arrays with some examples for your simple and better understanding. The use of JSONJSON is mostly used to transfer data from one system to another. It can transfer data between two computers, database, applications, etc. It is primarily used to transfer serialized data over the network connection. It can be used with all the major programming languages. Useful in data transition from web application to server. Most web services use JSON based format for data transfer. Properties of JSONLet's summarize the properties:It is a text-based lightweight data interchange format. It has been extended from the JavaScript language. Its extension is .json. Being a text-based format, it is easy to read and write off both the user/programmer and the machines. This is independent of programming languages but it also uses the conventions that are pretty well known within the C family of languages like C, C++, C#, JavaScript, Java, Python, Perl etc. So far we have discussed JSON characteristics and use. From now on we will discuss the structure of JSON or JavaScript Object Notion.JSON grew out of a need for a real-time server to browser communication that can work without using any extra plugins like java applets or flash. So, after realizing the need for a communication protocol that can be used in real time, Douglas Crockford Crockford JSON in the early 2000.Earlier JSON was seen as the subcategory of JavaScript and was used vividly with the same. But the code for serialization and parsing JSON is available in almost all the major languages. Syntax of JSONBy now you must have gained some basic knowledge of JSON. Let's take a look at the basic syntax used to form a JSON. JSON can in principle be classified by being built on two structural units. They are a collection of name-value pairs and the ordered list of values. JSON is a universal data structure that most of the programming language available today supports them. This makes the work of a programmer much easier to have a interchangeable data type that can work across different languages. Let us know more about these data types:The name value par collection is realized as an object, strut, post, dictionary etc. The ordered value list is realized as an array, list, etc. We have seen almost all the basic theories up to now. Let's move on and take a look at the basic JSON structure. In this Example, we are considering a JSON representing the details of a Car.Let's assume that we have a car object with the following basic characteristics and their attributes:Make and Mode = Maruti Suzuki Swift Make Year = 2017 Color = Red Type = HatchbackSo, if we want to transfer this data using a JSON file, then the serialization of this data will create a JSON. That JSON will look something like this:We've seen about the use of JSON, its basic structure, and how data is presented in the JSON format. Now, let's take a closer look at how different elements are built up in JSON. What is a JSON Object? The JSON object is a set of Keys along with its values without any specific order. The key and their values are grouped using braces, both opening and closing { }. So, in the previous example when we were creating a JSON with a car attribute, we were actually creating a JSON Car Object. There are certain rules that need to be followed at the same time as we create a JSON structure, we will learn about these rules while discussing the key value pairs. So, to create a JSON, the first thing we're going to need is an attribute. Here we create an Employee JSON object. The next thing we need is to specify the properties of the object, let's assume our co-worker has a first name, last name, employee ID and designation. These properties for the employee are represented as Keys in the JSON structure. Let's create a JSON object:Everything within the braces is known as JSON Employee Object.A basic JSON object is represented by Key-Value pairs. In the previous Example, we used a JSON to represent an employee's data. And we have represented different characteristics for the employee; First name, Last name, employment ID, and designation. Each of these keys has a value in JSON. For example, First name is represented by a value Sam. Similarly, we have also represented other keys by using different values. Rules to be followed while while a JSON:JSON Objects should start and end with braces { }. Key fields are included in the double quotes. Values are represented by putting : colon between them and the keys. JSON key-value pair is separated by a comma,. Values can be of any data type such as String, Integer, Boolean etc. A little exercise for you. Try to create a sample JSON that describes an Employee with your own set of Keys and Values.By now, you must have had a basic understanding of what is JSON? Use of JSON and How it looks? Now, let's dive deeper into more complex JSON structures. JSON ArraysArrays in JSON are similar to those found in any programming language, the json array is also an orderly collection of data. The matrix starts with a left bracket [and ends with the right square bracket]. The values inside the array are separated by a comma. There are some basic rules that must be followed if you are going to use an array in a JSON. Let's take a look at a sample JSON with an Array. We will use the same Employee object that we used before. We will add another property like Language Expertise. An employee can have expertise in multiple programming languages. So, in this case, we can use an array to offer a better way to register multiple language values. As we have already discussed, there are also few rules that must be followed, while maintaining a matrix in a JSON. They are:A matrix in JSON will start with a left bracket and will end with a right bracket. Values inside the array will be separated by a comma. Objects, Key-value pairs, and Arrays make different components of JSON. These can be used together to record all the data in a JSON. Now, as we have already discussed the basic structure of JSON lets start working on a more complex JSON structure. Earlier in this tutorial, we gave you two Examples of JSON as shown below. Employed JSONCar JSON Now, let's assume that there is more than 1 employee and they also have a car. So, we will need to arrange the data in such a way that the car JSON should also be included in employee JSON to make the entry a complete. This means that we will need to create a nested Car JSON object inside Employee JSON. To include the car in Employee JSON, initially, we need to include a Key as car in JSON. Something like this:Once we have added the car key to the employee JSON, we can then pass the value directly to the car JSON. { FirstName: Sam, LastName: Jackson, employeeID: 5698523, Designation : Manager, LanguageExpertise : [Java, C#, Python] Car : { Make&Model: Maruti Suzuki Swift, MakeYear: 2017, Color: Red, Type: Hatchback, } } This way we can create a Nested JSON. Let's adopt a situation where there are multiple employees, so we need to create a JSON that can hold data for multiple employees. { FirstName: Sam, LastName: Jackson, employeeI-D: 5698523, Designation: Manager, LanguageExpertise: [Java, Python], Car: { Make&Model: Maruti Suzuki Swift, MakeYear: 2017, Color: Red, Type: Hatchback } }, { FirstName: Tam, LastName: Richard, employeeID: 896586, Designation: Senior Manager, LanguageExpertise: [Ruby, C#], Car: { Make&model: Hyundai Verna, MakeYear: 2015, Color: Black, Type: Sedan } } In the above Example, you can clearly see that we have included the data for two employees. Again, there are few considerations while creating this kind of complex JSON structures. First, remember to include all json structure inside a bracket [ ]. A comma is used to separate the two different set of data in a JSON, whether it is a key value pair or a JSON object. When we get to the end of the tutorial, here's a little exercise for all of you. Create a json company with different kpis. Given below are the steps you will need to follow:#1) Open a notepad or some text editor.#2) Create a company JSON with different key-value pairs.#3) Add data for at least two companies.#4) Include an array field in JSON.#5) Use a nested JSON.#6) Now navigate JSON Validator.#7) Paste your JSON structure inside the text area and click validate to validate your JSON. Make sure you follow all of the above procedures and rules while creating a JSON. Here is the validation of the employee JSON that we created earlier with JSON Validator.ConclusionJSON is one of the most popular data transition formats. It is mostly used for data transition between different networks. The text-based structure allows JSON to be read and deconstruct into individual data easily either by a user or by any machine. JSON but sometimes described as the subclass of JavaScript, can be read/modified by all programming languages. JSON files have an extension of .json and can be created using any programming language. We can create a simple JSON by directly assigning Key value pairs, or we can use arrays to assign multiple values to a key. Other than the simple structure, JSON can also have a nested structure, which means that a JSON can have another JSON object described inside it as a key. This allows the user to transfer more complex data through the format. Please let us know if you have any questions or if you need more clarification. Next tutorial #2: Create JSON objects with C# (part 1) 1)