JavaScript Basics: How to create a Dictionary with Key/Value pairs

Sep 5, 2015

In statically typed programming languages a Dictionary (Key/Value pair collection) object can be very useful at times. While JavaScript doesn't natively include a type called "Dictionary", it does contain a very flexible type called "Object". The JavaScript "Object" type is very versatile since JavaScript is a dynamically typed language. This flexibility allows for the "Object" type to be used in ways that might seem strange when compared to statically typed languages such as C#.

Creating a Dictionary in JavaScript

While there's no "Dictionary" type in JavaScript, it's actually really easy to create and use a dictionary object. The following steps show how to go about creating and populating a dictionary with Key/Value pairs:

Step 1: Create a new Object

```
var dict = new Object();

// or the shorthand way
var dict = {};
```

You can also initialize the Dictionary with Key/Value pairs when creating it if you are using the shorthand method.

```
var dict = {
  FirstName: "Chris",
  "one": 1,
  1: "some value"
};
```

Step 2: Populate Values on the Object

This can be done by either setting the values using the Object's Indexer property, or just calling it directly as if it were a standard property on the object. As you'll see below the Key values can be any object and are not limited to Strings.

```
// using the Indexer
dict["one"] = 1;
dict[1] = "one";

// add new or update property
dict["Age"] = 42;

// direct property by name
// because it's a dynamic language
dict.FirstName = "Chris";
```

Iterating Key/Value Pairs

A simple JavaScript "for" loop can be used to iterate through your new dictionary.

```
for(var key in dict) {
  var value = dict[key];

  // do something with "key" and "value" variables
}
```

Recent Posts

On Finding a Business Partner
10 Oct 2020

Loop through JavaScript Arrays using for, forEach, and map functions
01 Jul 2020

JavaScript: Convert String to Number
06 Apr 2020

Encryption and Code are Greater Than Guns and Bombs
11 Mar 2020

macOS: Monitor CPU Usage and Frequency Info
15 Feb 2020

Related Posts

Loop through JavaScript Arrays using for, forEach, and map functions
01 Jul 2020

JavaScript: Convert String to Number
06 Apr 2020

Basics of JavaScript Class Inheritance
02 Jan 2020

Call Functions in JavaScript
24 Jul 2019

Recent on Build5Nines.com

Latest Cloud News: IoT, Security, Azure Sphere, and more! (December 4, 2020 Build5Nines Weekly)
04 Dec 2020

Latest Cloud News: Apple on K8s, IoT, Microsoft Pluton and more! (November 20, 2020 Build5Nines Weekly)
20 Nov 2020

Latest Cloud News: .NET 5 Released, Apple Silicon M1 CPU, and more! (November 12, 2020 Build5Nines Weekly)
12 Nov 2020

By using the “for(var key in dict)” method of iterating on the object you are able to easily access all the key/value pairs in contains. This method will iterate with each “key” value being the Index on the Object that can be used to access the associated Value.

Access Key/Value Pairs Directly

The Key/Value pairs can be accessed directly on the dictionary Object either through the indexer or as if it’s directly a property on the object.

```
// using indexer
var name = dict["FirstName"];

// as property
var name = dict.FirstName;
```

Functions can be Key or Value Too!

Since JavaScript is a functional language, functions are objects too. As a result, Functions can also be used as either Key and/or Value on your dictionary. For the Dictionary Key, this really does make sense since the Dictionary is really just an Object which can have functions/methods on it. However using Functions as the value may not be the most obvious for developers most comfortable with non-functional programming languages.

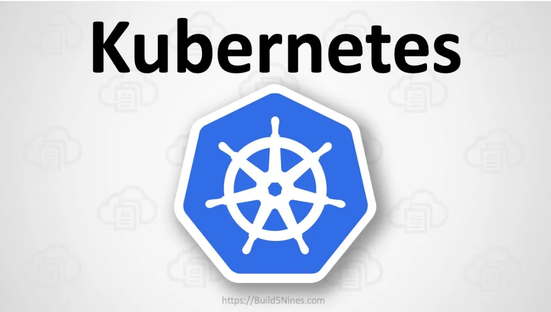
Here are some examples of using a Function in both Key and Value of a Dictionary:

```
var dict = {};

var f = function() {
  // do something
};

// setup Function as Value
dict['method'] = f;

// setup Function as Key
dict[f] = 'some value';

// execute Function from Value
dict['method']();

var method = dict.method;
method();

// get value for Key
var val = dict[f];
```

Hopefully this offers some additional clarification and/or insights into how JavaScript works, and how to make it work how you want/need it to. Happy coding!

Related Posts

- Loop through JavaScript Arrays using for, forEach, and map functions
  01 Jul 2020
- JavaScript: Convert String to Number
  06 Apr 2020
- Basics of JavaScript Class Inheritance
  02 Jan 2020
- Call Functions in JavaScript
  24 Jul 2019



Fix Kubernetes Dashboard Strange 401 Unauthorized, 503 Service Unavailable Errors

11 Nov 2020

Latest Cloud News: Kubernetes, Terraform, Teams Multi-Login and more! (November 5, 2020 Build5Nines Weekly)

05 Nov 2020

Legal Disclaimer

This site is a participant in the Amazon Services LLC Associates Program, an affiliate advertising program designed to provide a means for sites to earn advertising fees by advertising and linking to Amazon.com. We also participates in affiliate programs with Udemy, and other sites. This site is compensated for referring traffic and business to these companies.