Table of Contents

1 Objects 1.1

2 Functions 1.2

1 So far, we have looked at different values like, **Numbers**, **Strings**, **Booleans**, and **Arrays**, which are a collection of other values.

Now we will look at another important data type in Javascript, called an **Object**.

2 Let's see what an Object looks like:

```
var myFirstObject = {
    name: "Jack The Rabbit",
    species: "Sylvilagus bachmani",
    speed: 62,
    points: 1206,
    age: 3,
    favouriteFoods: [ "Carrot Shake", "Oreos" ]
};
```

3 **Objects** are a collection of different values put together, much like Arrays. Except that values inside **Objects** aren't accessible by an **index**, like with **Arrays**, but they are accessible with **keys**.

4 What's a **key**? A **key** is the label that a **value** is stored with.

For example, in the **Object** above, what is the **key** that the value **3** is stored with? It's the **age** key.

5 What about the value `"Sylvilagus bachmani"` ?

6 Try the following and see what pops up.

```
alert( myFirstObject.name );
alert( myFirstObject[ "name" ] );
alert( myFirstObject[ "points" ] );
alert( myFirstObject.favouriteFoods );
alert( myFirstObject.favouriteFoods[ 1 ] );
```

7 Let's look at more example **Objects**.

```
var playerOne = {
      name: "Marjan",
      playerNum: 1,
      health: 100,
      level: 11,
      ability: "Fast Runner",
      items: [ "Flash light", "Stick" ]
};
```

```
var playerTwo = {
      name: "Stevie",
      playerNum: 2,
      health: 89,
      level: 3,
      ability: "Heavy Lifting",
      items: [ "Stone", "Rope", "Mug" ]
};
```

8 Notice, the value with the **items** key is an **Array**. You can store **Arrays** inside **Objects**!

9 Can you tell what the following line of code will display?

```
alert( playerTwo.health );
alert( playerOne.items[ 0 ] );
```

10 You can change the values inside an Object like this.

```
playerOne.health = 65;
```

11 Can you add more items to playerTwo's items array?

12 Can you remove playerOne's first item?

1 Functions are the last big topic we will cover in this series. Functions are extremely central to Javascript and all other programming languages.

2 We have used a few functions by now, but we have not talked about them. They are:

Function name	What it does
`alert( message )`	to display a pop-up with a message
`prompt( message )`	to display an input dialog for the user with a message
`Number.parseInt( num )`	to convert a **String** into a **Number**
`push( item )`	to add an item to an **Array**

3 These are all methods of **doing** things. You can think of them as **actions**.

For example, type out this simple function.

```
function basicAlert() {
    alert( "Hello from the basicAlert!" );
}
```

4 Congratulations! You just defined your first javascript function!

If you use this function, you get an alert pop up.

Try it. Execute it using the following code:

```
basicAlert();
```

5 You should see a pop up with the message `"Hello from the basicAlert!"`.

Let's make a new similar function with a new name and tweak it a little from the first one.

```
function customAlert( place ) {
 alert( "Hello from " + place );
}
```

6 Now try the following… what do you see?

```
customAlert( "CoderDojo" );
```

7 Let's examine the previous example more closely…

```
function customAlert( place ) {
 alert( "Hello from " + place );
}
```

8 Notice the different between the **basicAlert** function and the **customAlert** function.

`customAlert` has a special type of variable called **place**.

Here's another function has a special variable, but this time the special variable is called **num**, and it does something different with it.

```
function doubler( num ) {
 return num * 2;
}
```

9 Notice a new special word there? The **return** keyword does something special. It returns back a value to the person who executed that function!

10 Can you guess what the newNumber variable will be?

```
var newNumber = doubler( 12 );
```

Try it and see if your guess was right.