19. JavaScript Objects

"Everything" in JavaScript is an Object: a String, a Number, an Array, a Function....

In addition, JavaScript allows you to define your own objects.

19.1. JavaScript Objects

JavaScript has several **built-in** objects, like String, Date, Array, and more.

An object is just a special kind of data, with **properties** and **methods**.

a) Accessing Object Properties

Properties are the values associated with an object.

The syntax for accessing the property of an object is:

objectName.propertyName

This example uses the length property of the String object to find the length of a string:

```
var message="Hello World!";
var x=message.length;
```

The value of x, after execution of the code above will be:

```
12
```

b) Accessing Objects Methods

Methods are the actions that can be performed on objects.

You can call a method with the following syntax:

```
objectName.methodName()
```

This example uses the toUpperCase() method of the String object, to convert a text to uppercase:

```
var message="Hello world!";
var x=message.toUpperCase();
```

The value of x, after execution of the code above will be:

```
HELLO WORLD!
```

19.2. Creating JavaScript Objects

With JavaScript you can define and create your own objects.

There are 2 different ways to create a new object:

- 1. Define and create a direct instance of an object.
- 2. Use a function to define an object, then create new object instances.

a) Creating a Direct Instance

This example creates a new instance of an object, and adds four properties to it:

Example

```
person=new Object();
person.firstname="John";
person.lastname="Doe";
person.age=50;
person.eyecolor="blue";
```

Alternative syntax (using object literals):

Example

```
person={firstname:"John",lastname:"Doe",age:50,eyecolor:"blue"};
```

b) Using an Object Constructor

This example uses a function to construct the object:

Example

```
function person(firstname,lastname,age,eyecolor)
{
```

```
this.firstname=firstname;
this.lastname=lastname;
this.age=age;
this.eyecolor=eyecolor;
}
```

The reason for all the "this" stuff is that you're going to have more than one person at a time (which person you're dealing with must be clear). That's what "this" is: the instance of the object at hand.

19.3. Creating JavaScript Object Instances

Once you have a object constructor, you can create new instances of the object, like this:

```
var myFather=new person("John","Doe",50,"blue");
var myMother=new
person("Sally","Rally",48,"green");
```

a) Adding Properties to JavaScript Objects

You can add new properties to an existing object by simply giving it a value.

Assume that the personObj already exists - you can give it new properties named firstname, lastname, age, and eyecolor as follows:

```
person.firstname="John";
person.lastname="Doe";
person.age=30;
person.eyecolor="blue";

x=person.firstname;
```

The value of x, after execution of the code above will be:

```
John
```

b) Adding Methods to JavaScript Objects

Methods are just functions attached to objects.

Defining methods to an object is done inside the constructor function:

```
function person(firstname,lastname,age,eyecolor)
{
this.firstname=firstname;
this.lastname=lastname;
this.age=age;
this.eyecolor=eyecolor;

this.changeName=changeName;
function changeName(name)
{
this.lastname=name;
}
}
```

The changeName() function assigns the value of name to the person's lastname property.

Now You Can Try:

```
myMother.changeName("Doe");
```

JavaScript knows which person you are talking about by "substituting" **this** with **myMother**.

19.4. JavaScript Classes

JavaScript is an object oriented language, but JavaScript does not use classes.

In JavaScript you don’t define classes and create objects from these classes (as in most other object oriented languages).

JavaScript is prototype based, not class based.

a) JavaScript for...in Loop

The JavaScript for...in statement loops through the properties of an object.

Syntax

```
for (variable in object)
  {
  code to be executed
  }
```

Note: The block of code inside of the for...in loop will be executed once for each property.

Example

Looping through the properties of an object:

```
var person={fname:"John",lname:"Doe",age:25};

for (x in person)
  {
  txt=txt + person[x];
  }
```