JavaScript Objects and Object Oriented Programming

http://www.w3schools.com/js/js_obj_intro.asp

JavaScript Objects

JavaScript is an Object Oriented Programming (OOP) language. An OOP language allows you to define your own objects and make your own variable types.

JavaScript has a number of built-in objects*.

*an object is just a special kind of data

JavaScript Objects

Javascript is a prototype-based object oriented language.

Prototype - an initial object that can be cloned or modified to create new objects.

"Plans" for **class** constructs are underway for JS.

JavaScript Properties and Methods

Properties are the values associated with an object.

```
<script type="text/javascript">
var txt="Hello World!";
document.write(txt.length); //12
</script>
```

Methods are the actions that can be performed on objects.

```
<script type="text/javascript">
var str="Hello world!";
document.write(str.toUpperCase()); //HELLO WORLD
</script>
```

Reading Assignment

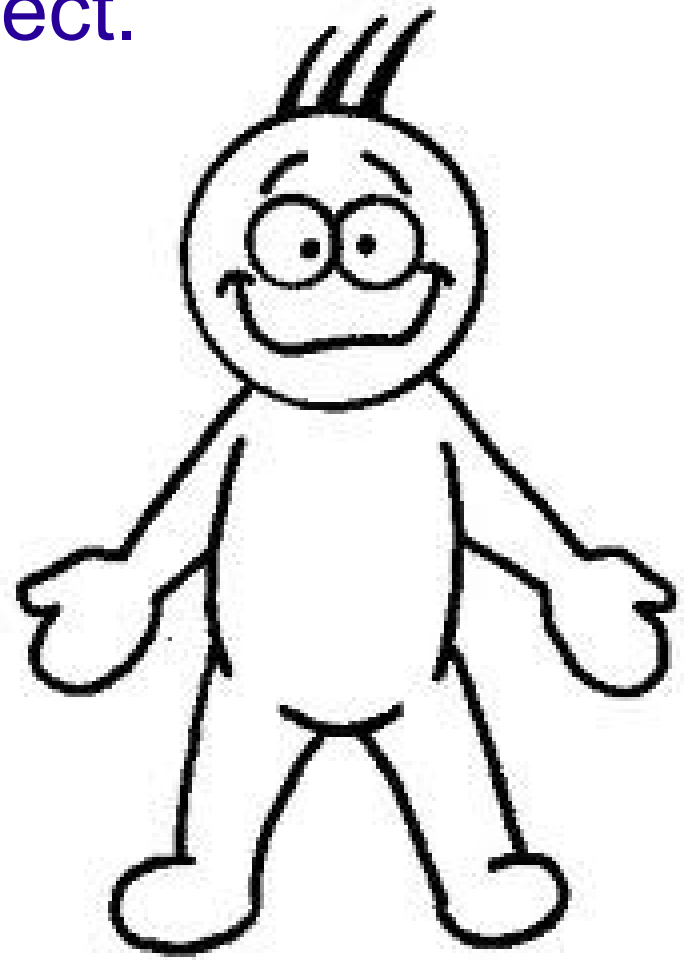
Javascript Built-in Objects
See link on our FB Group.

Objects in JavaScript

Let's say a Person is an object.

What are the possible properties for the object Person?
(name, age, height, etc)

What are the possible methods for the object Person?
(eat(), play(), sleep(), ...)



Objects in JavaScript

Properties

Syntax: objName.propName

Properties can be added to objects by simply giving them values. Assuming that personObj already exists...

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

document.write(personObj.firstname); //John
```

Objects in JavaScript

Methods

Syntax: objName.methodName()

Objects can also contain methods.

```
personObj.sleep();
document.write(personObj.firstname);
```

Creating Objects in JavaScript

Two different ways of creating a new object.

1. Create a direct instance of an object
2. Create an object constructor

Creating Objects in JavaScript

Create a direct instance of an object

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

Or using object literals

```
personObj={firstname:"John",lastname:"Doe",age:5
0,eyecolor:"blue"};
```

Creating Objects in JavaScript

Create an object constructor

```
function person(firstname,lastname,age,eyecolor)
{
this.firstname=firstname;
this.lastname=lastname;
this.age=age;
this.eyecolor=eyecolor; //notice the “this”
                        //keyword
}
```

Inside the function you need to assign things to this.propertyName. The reason for all the "this" stuff is that you're going to have more than one person object at a time (which person you're dealing with must be clear).

Creating Objects in JavaScript

Once you have the 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");
```

Creating Objects in JavaScript

Methods can also be added to the person object.

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

Methods are functions attached to objects.

```
function newlastname(new_lastname)
{
this.lastname=new_lastname;
}
//myMother.newlastname("Doe");
```

Adding Properties or Methods to an Existing object

```
myMother.husband = myFather;
myFather.hasMistress = false;
```

These add new properties to 'myMother' and 'myFather' only.
We can define new properties to existing
objects using this way BUT the property
is added to the target object only.

Adding a Property for an Object Type

The following will add new property that is shared by all objects of the specified type.

```
person.prototype.birthdate = null;
myMother.birthdate = new Date();
```