DOM-JAVASCRIPT

DOM

"It *is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.*"

Document Object Model - DOM

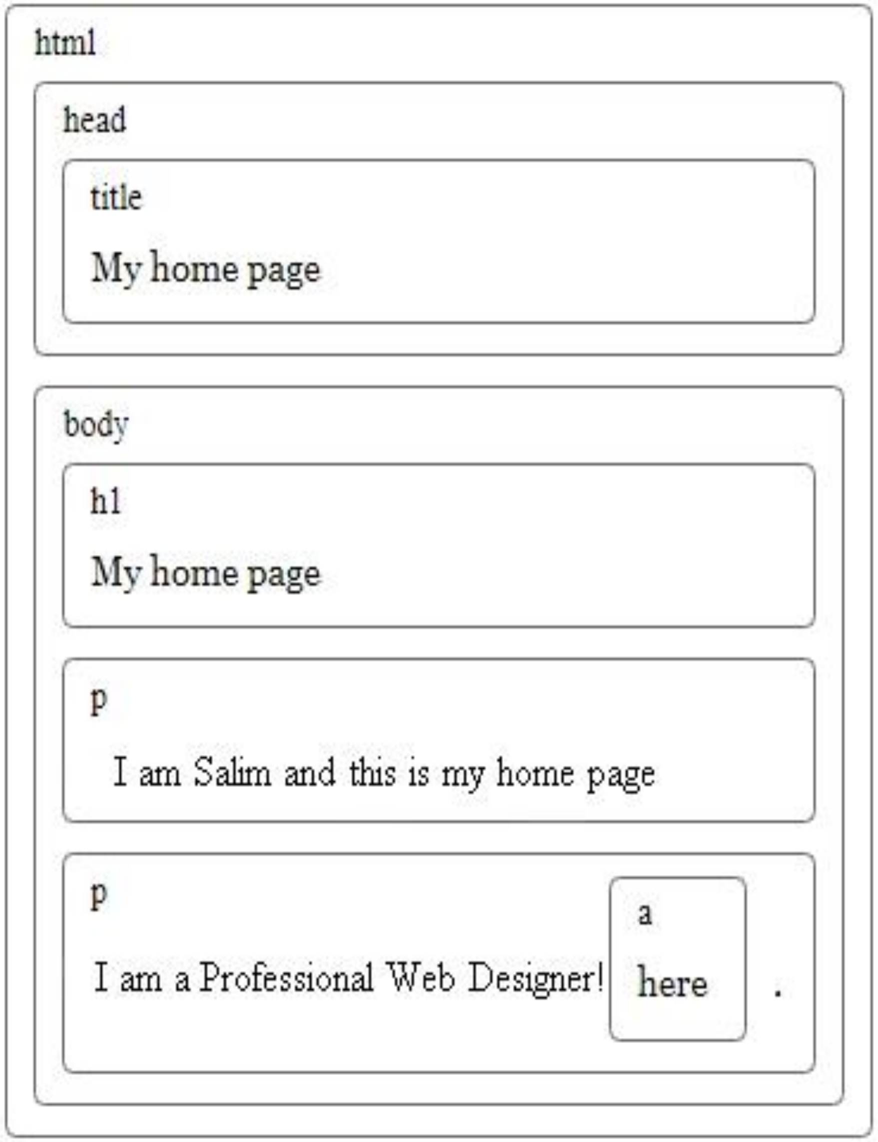
- DOM -is an essential part of making websites interactive.
- DOM is an interface that allows a programming language to manipulate the content, structure, and style of a website.
- JavaScript is the client-side scripting language that connects to the DOM in an internet browser.

Document Object Model - DOM

- **A Document object represents the HTML document that is displayed in that window. The Document object has various properties that refer to other objects, which allow access to modification of document content.**
- **In short,** The way a document content is accessed and modified is called the *Document Object Model*, or *DOM*

Imagine an HTML document as a nested set of boxes.

```
<!doctype html>
<html>
  <head>
      <title>My home page</title>
  </head>
  <body>
     <h1>My home page</h1>
     <p>Hello, I am ABC and this is
my home page.</p>
     <p>I am a Professional Web
Designer! Click here to  go to my
page
      <a
href="http://google.com">here</a
>.</p>
  </body>
</html>
```


html
head
title
My home page
body
h1
My home page
p
I am Salim and this is my home page
p
I am a Professional Web Designer!
a
here
.

HOW DOM WORKS ??

- When you open a web page in your browser, the browser retrieves the page's HTML text and parses it, browser builds up a model of the document's structure and uses this model to draw the page on the screen.

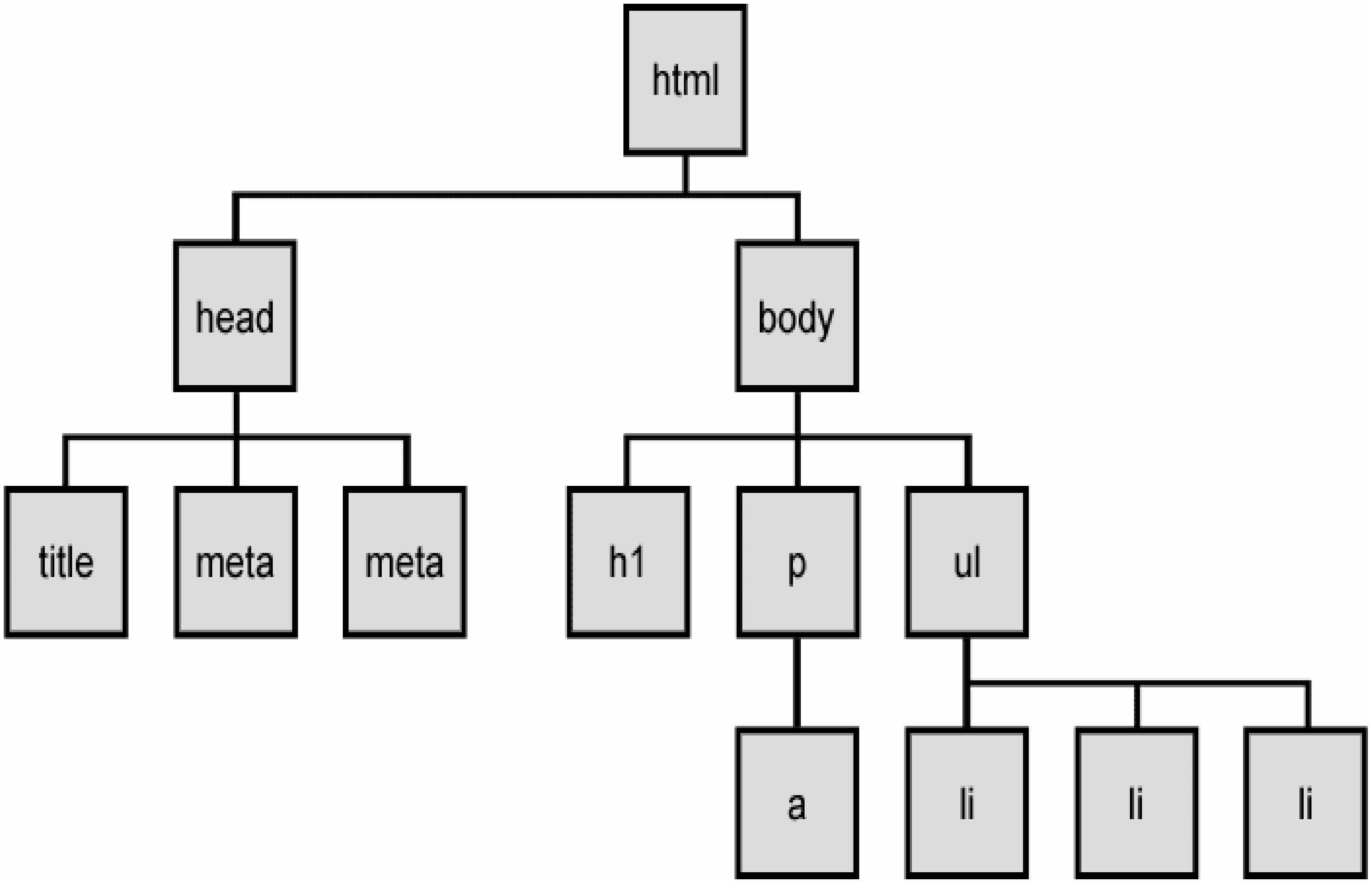
- This representation of the document is one of the toys that a JavaScript program has available in its sandbox.

How DOM works ??

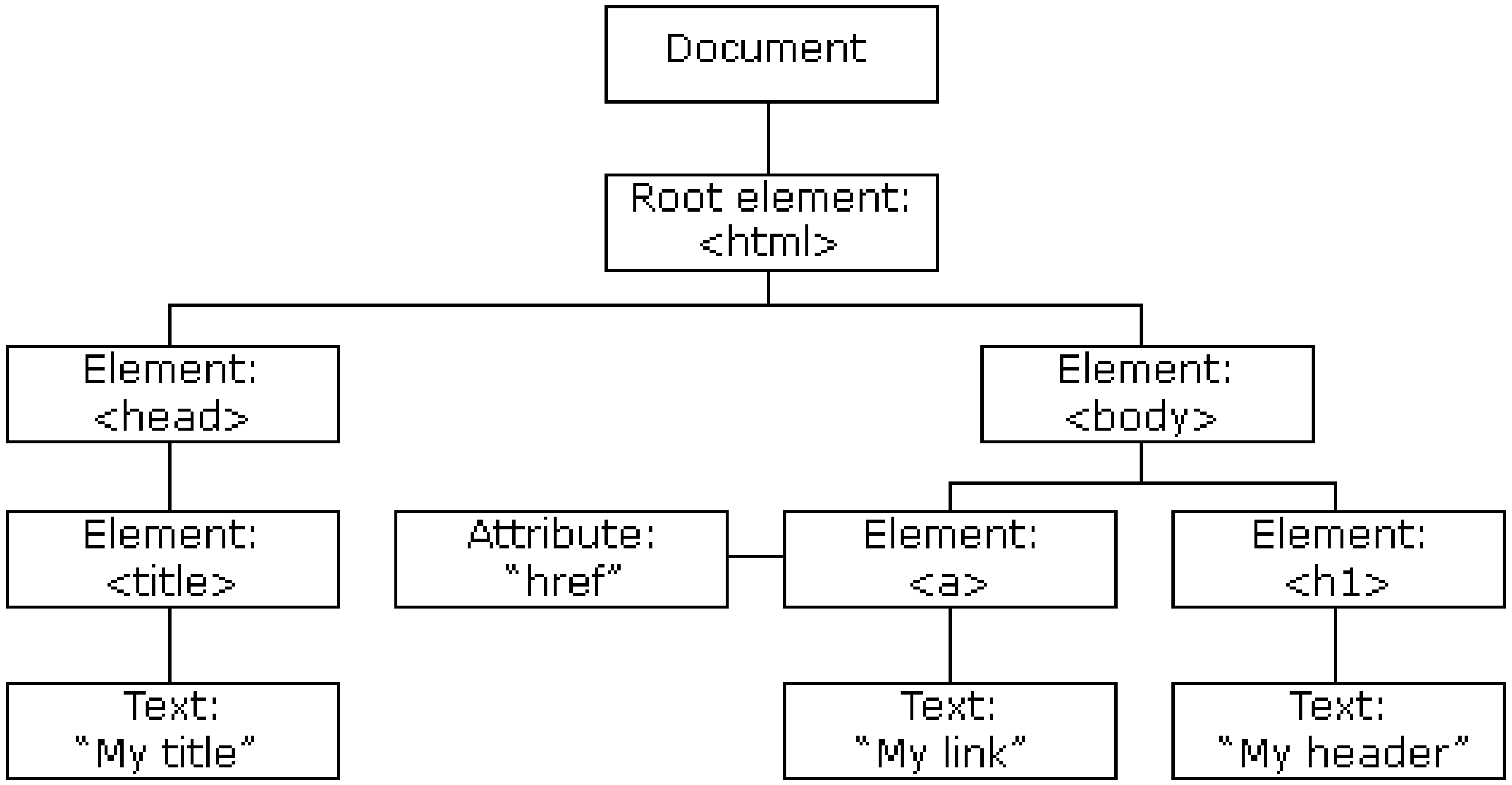
- It is a data structure that you can read or modify. It acts as a *live* data structure: when it's modified, the page on the screen is updated to reflect the changes.

DOM = TREE

DOM Tree



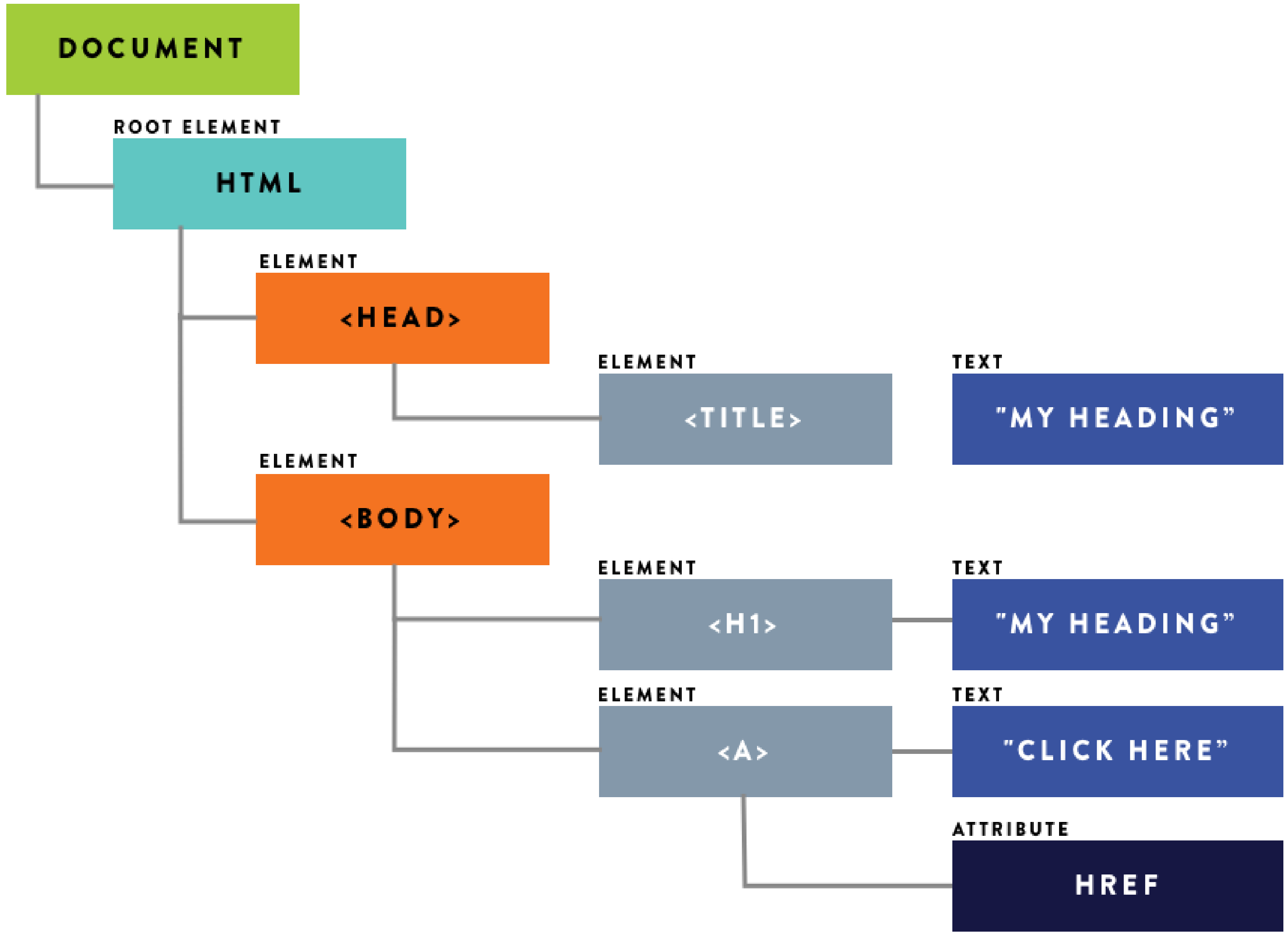
HTML DOM Tree of Objects



DOM - PARTS

- Core DOM
- XML DOM
- HTML DOM

Elements Access in Java Script



ELEMENTS ACCESS IN JAVASCRIPT

We can access the desired element by using from the web document using javascript method

- getElementById
- getElementByTagname
- getElementByClassname
- CSS Selector- Query selector

  document.querySelector(“tag.classname");
- HTML Object collection-Array

Var Dom_obj=document.getElementsById(“myinput”)

HTML DOM METHODS – [METHOD / PROPERTY]

- **getElementById** ----->Method

Most common way to access an HTML element is to use the id of the element.

- **innerHTML** ------> Property

1. InnerHTML property is used for getting or replacing the content of HTML Elements
2. InnerHTML property can be used to get or change any HTML element, including <html> and <body>

document.getElementById("demo").innerHTML= “welcome”;

Events

“Event is an activity that represents a change in the environment”

Event Handling

- Event Handler is a script that gets executed in response to these events.

- This event Handler enables the web document to respond the user activities through browser window.

- This special type of programming in which events may occur and these events get responded by executing the event handlers.

Programming - “ Event-driven programming”

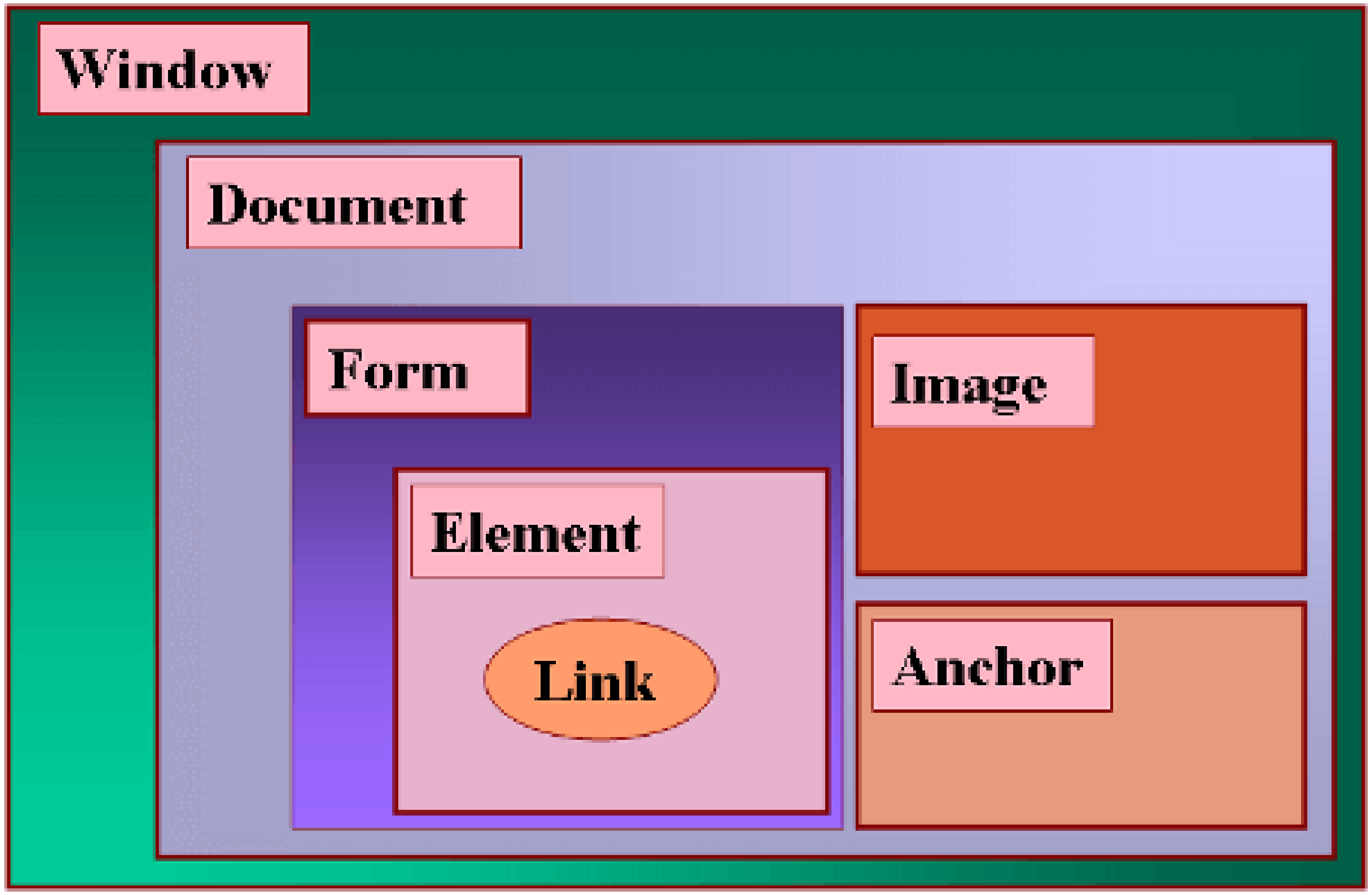
EVENTS

EVENTS	INTRINSIC EVENT ATTRIBUTE	MEANING
blur	onblur	Losing the focus
change	onchange	On occurrence of some change
click	onclick	When user click the mouse button
dbclick	ondbclick	When user double click the mouse button
focus	onfocus	When user requires input focus
Keyup	onkeyup	When user releases the key from the keyboard
keydown	onekeydown	When user presses the key down
mouseover	onmouseover	When user moves the mouse away over some element

EVENTS

EVENTS	INTRINSIC EVENT ATTRIBUTE	MEANING
keypress	onkeypress	When user presses the key
mousedown	onmousedown	When user clicks the left mouse button
mouseup	onmouseup	When user releases the left mouse button
mouseout	onmouseout	When user moves the mouse away from some element
load	onload	After getting the document loaded
reset	onreset	When the reset button is clicked
Submit	onsubmit	When the submit button is clicked
Select	onselect	On selection
unload	onunload	When user exits the document

WINDOW OBJECTS



WINDOW OBJECTS

Methods

- alert(message)
- prompt(message, default_text)
- confirm(message)
- close()
- open(url, window name)
- Set TimeOut()

WINDOW OBJECTS

Properties

- closed
- defaultstatus
- status
- frame
- location
- Name
- Self
- parent

THE JAVASCRIPT LANGUAGE

WHAT IS JAVASCRIPT?

HTML	- Content Layer
CSS	-Presentation layer
JS	-Interactive Layer

HOW JS INTERACTS WITH HTML?

- Document model-Everything is represented as objects.
- JS can easily interpret those objects.

LANGUAGE ELEMENTS

- Variables
- Literals
- Operators
- Control Structures
- Functions
- Objects

JAVASCRIPT VARIABLES

- Untyped- any data type.
- Can be declared with var keyword:

```
var name;
```

- Can be created automatically by assigning a value:

```
val=1;     call="Hi John";
```

VARIABLES (CONT.)

- Using **`var`** to declare a variable results in a *local* variable (inside a function).

- If you don't use **`var`** – the variable is a global variable.

LITERALS

- The typical bunch:
  - Numbers `17` `123.45`
  - Strings `"Hello Dave"`
  - Boolean: `true` `false`
  - Arrays: `[1,"Hi John",17.234]`

Arrays can hold anything!

ARRAYS

- Arrays are actually Javascript Objects.
- The only thing special in the language to support arrays is the syntax for literal.

OPERATORS

- Arithmetic, comparison, assignment, bitwise, boolean (pretty much just like C++).

```
+ - * / % ++ -- == != > <
&& || ! & | << >>
```

DIFFERENT THAN C++

- The + operator is used for addition (if both operands are numbers)

-or-

- The + operator means string concatenation (if either one of the operands is not a number)

CONTROL STRUCTURES

- Again – pretty much just like C:

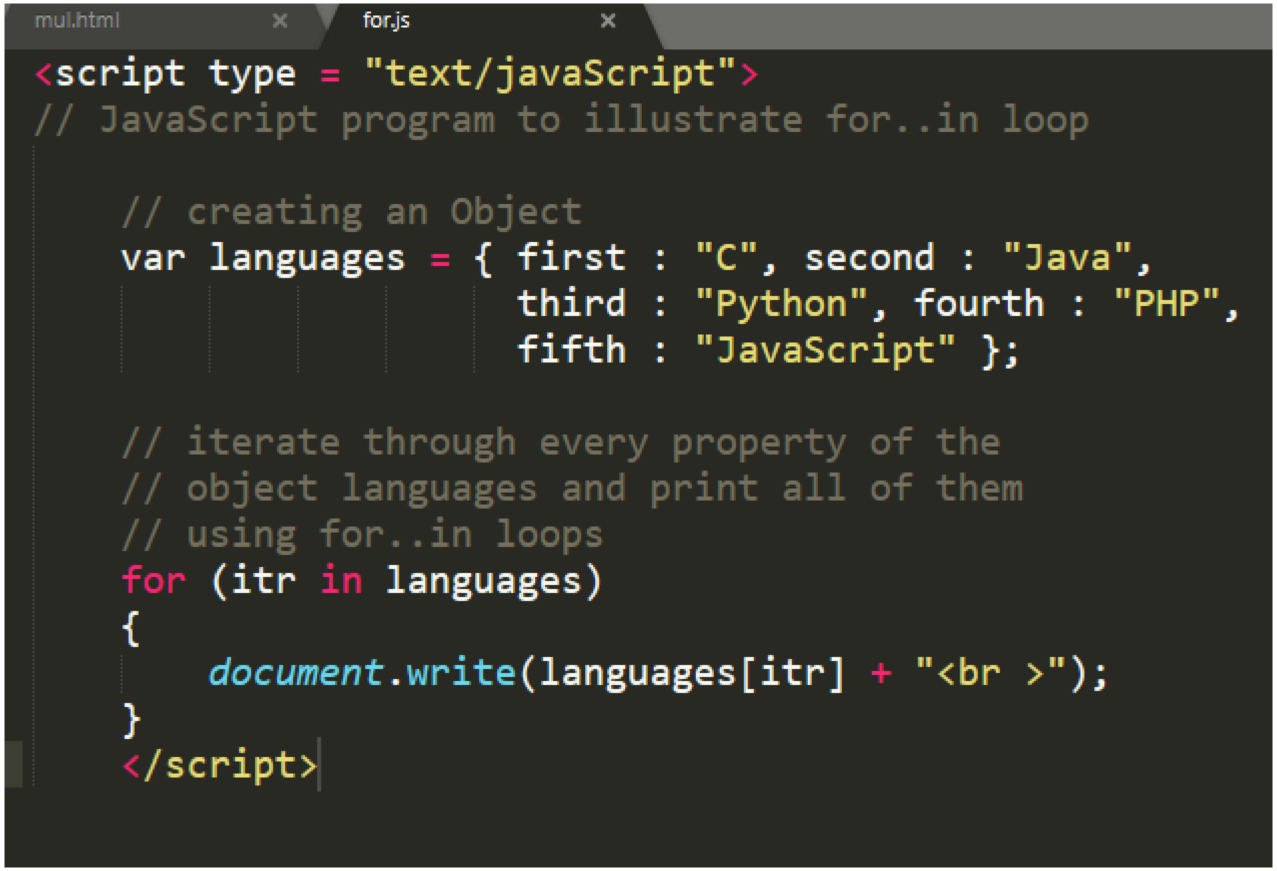
```
        if   if-else   ?:  switch

          for   while do-while
```

- And a few not in C

```
          for (var in object)
```

EXAMPLE



```
<script type = "text/javaScript">
// JavaScript program to illustrate for..in loop

    // creating an Object
    var languages = { first : "C", second : "Java",
                      third : "Python", fourth : "PHP",
                      fifth : "JavaScript" };

    // iterate through every property of the
    // object languages and print all of them
    // using for..in loops
    for (itr in languages)
    {
        document.write(languages[itr] + "<br >");
    }
    </script>
```

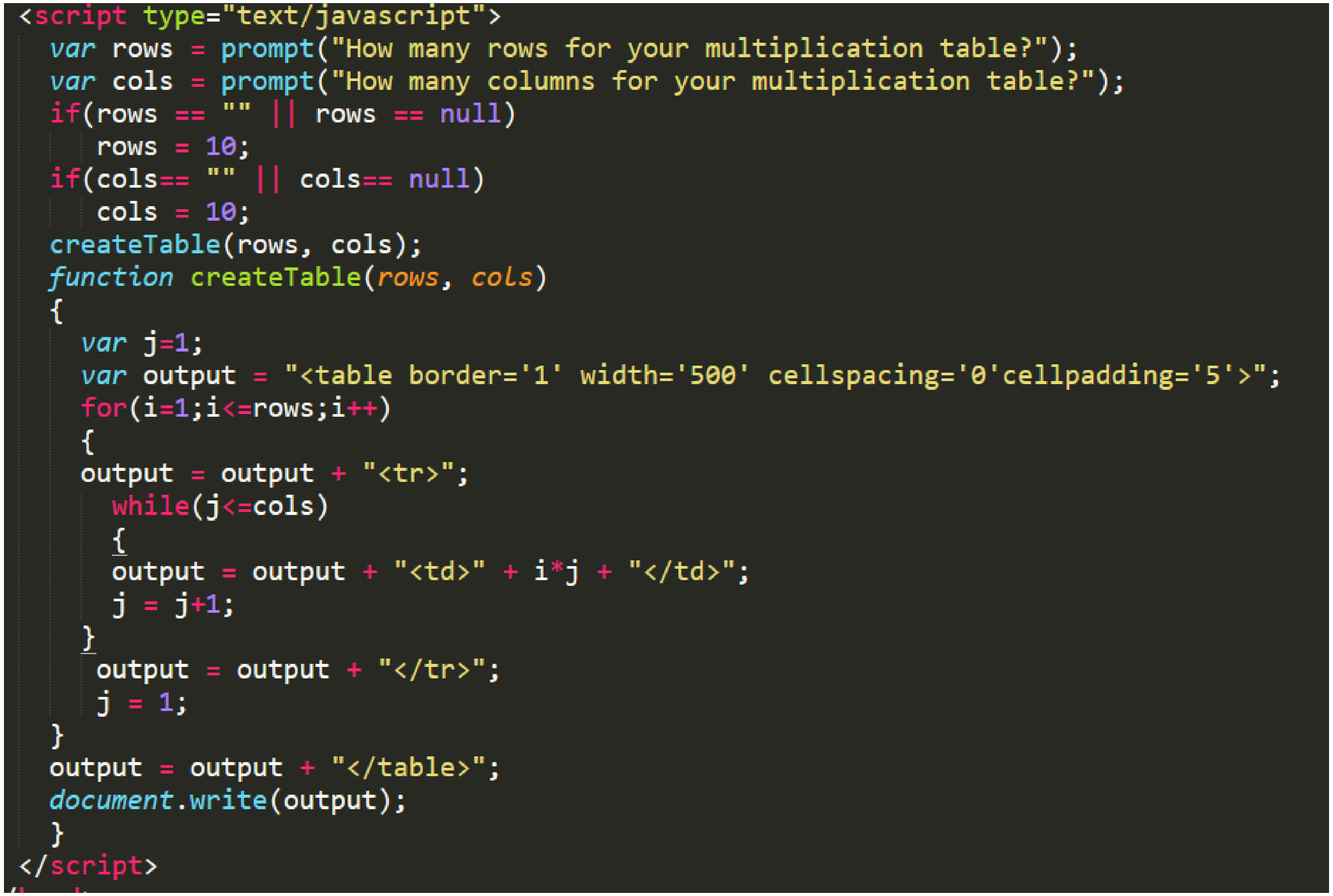
JAVASCRIPT FUNCTIONS

- The keyword **`function`** is used to define a function (subroutine):

```
function add(x,y) {
  return(x+y);
}
```

- No type is specified for arguments!

JAVASCRIPT PROGRAM



```
<script type="text/javascript">
  var rows = prompt("How many rows for your multiplication table?");
  var cols = prompt("How many columns for your multiplication table?");
  if(rows == "" || rows == null)
     rows = 10;
  if(cols== "" || cols== null)
     cols = 10;
  createTable(rows, cols);
  function createTable(rows, cols)
  {
    var j=1;
    var output = "<table border='1' width='500' cellspacing='0'cellpadding='5'>";
    for(i=1;i<=rows;i++)
    {
    output = output + "<tr>";
      while(j<=cols)
      {
      output = output + "<td>" + i*j + "</td>";
      j = j+1;
    }
     output = output + "</tr>";
     j = 1;
  }
  output = output + "</table>";
  document.write(output);
  }
</script>
```

RECURSION IS SUPPORTED

```
function factorial(x) {
   // use <= 0 instead of < 0
   // to avoid problems with neg numbers

   if (x<=0)
       return(1);
   else
       return( x * factorial(x-1));
}

document.write("<H3>11! = " +
  factorial(11) + "</H3>");
```

OBJECTS

- Objects have attributes and methods.
- JavaScript is template based not class based.
- Here, we don't create class to get the object. But, we directly create objects.
- **Helps to bundle all the entity in single name.**

CREATING OBJECTS

- There are 3 ways to create objects.
- By object literal
  - emp={id:102,name:"SSS",salary:40000}
- By creating instance of Object directly (using new keyword)
  - var emp=new Object();
    - emp.id=101;
    - emp.name=“AAA";
    - emp.salary=50000;
- By using an object constructor (using new keyword)

EXAMPLE

```
function emp(id,name,salary){
   this.id=id;
   this.name=name;
   this.salary=salary;
         }
  e=new emp(103,“AAA",30000);
```

ARRAYS

- **JavaScript array** is an object that represents a collection of similar type of elements.
- There are 3 ways to construct array in JavaScript

1. By array literal

```
<script>
var emp=["Sonoo","Vimal","Ratan"];
for (i=0;i<emp.length;i++){
document.write(emp[i] + "<br/>");
}
</script>
```

2.By creating instance of Array directly (using new keyword)

```
<script>
var i;
var emp = new Array();
emp[0] = "Arun";
emp[1] = "Varun";
emp[2] = "John";
for (i=0;i<emp.length;i++){
document.write(emp[i] + "<br>");
}
</script>
```

3. By using an Array constructor (using new keyword)

```
<script>
var emp=new Array("Jai","Vijay","Smith");
for (i=0;i<emp.length;i++){
document.write(emp[i] + "<br>");
}
</script>
```

THE DOCUMENT OBJECT

- Many attributes of the current document are available via the **`document`** object:

`Title`	`Referrer`
`URL`	`Images`
`Forms`	`Links`
`Colors`	

DOCUMENT METHODS

- **`document.write()`** like a print statement – the output goes into the HTML document.
- **`document.writeln()`** adds a newline after printing.

```
document.write("My title is" +
  document.title);
```

EXAMPLE

```
<HEAD>
<TITLE>JavaScript is Javalicious</TITLE>
</HEAD>
<BODY>
<H3>I am a web page and here is my
  name:</H3>
<SCRIPT>
document.write(document.title);
</SCRIPT>
<HR>
```

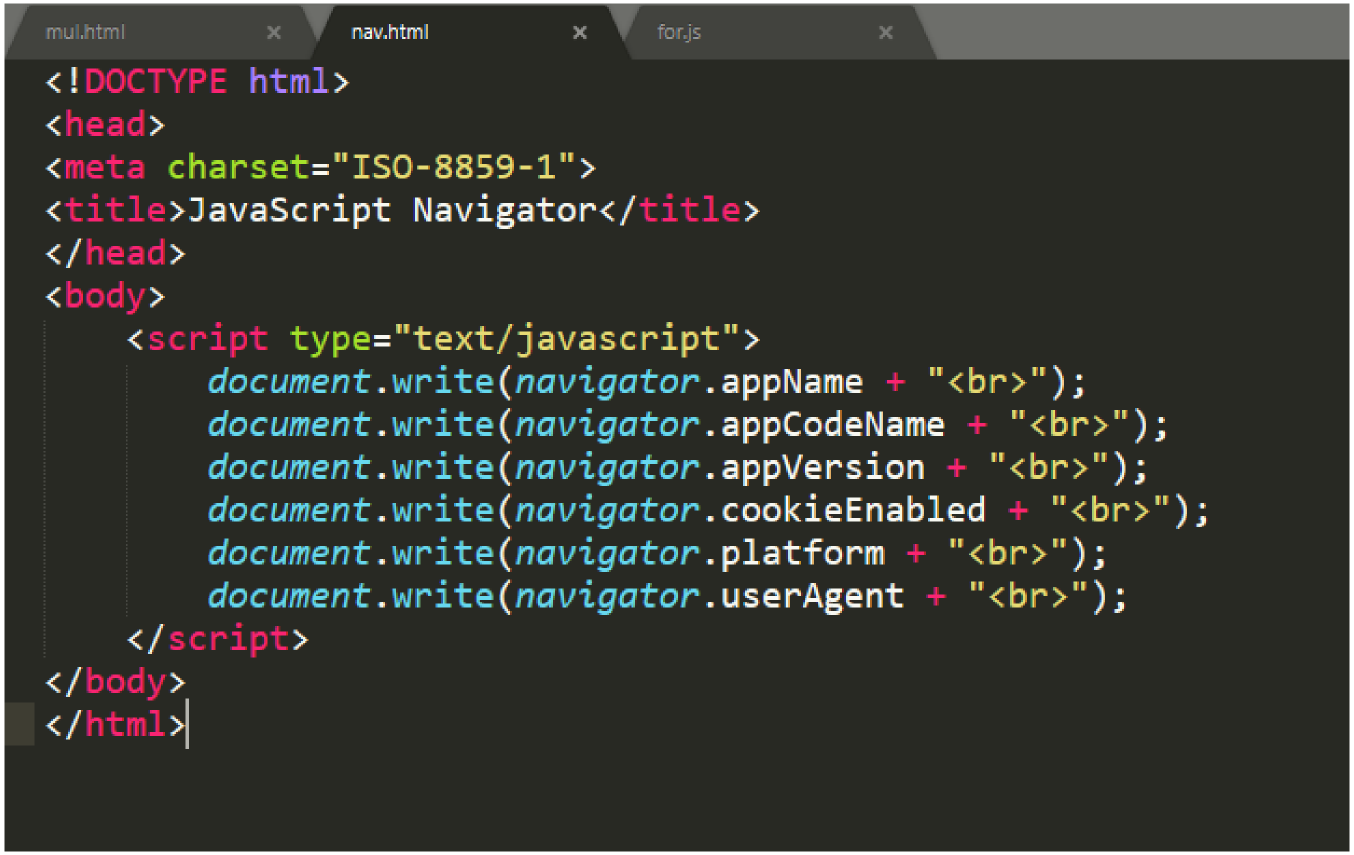
THE NAVIGATOR OBJECT

- Represents the browser. Read-only!
- Attributes include:

```
appName
appVersion
platform
```

often used to determine what kind of browser is being used (Netscape vs. IE)

NAVIGATOR EXAMPLE



```
<!DOCTYPE html>
<head>
<meta charset="ISO-8859-1">
<title>JavaScript Navigator</title>
</head>
<body>
    <script type="text/javascript">
        document.write(navigator.appName + "<br>");
        document.write(navigator.appCodeName + "<br>");
        document.write(navigator.appVersion + "<br>");
        document.write(navigator.cookieEnabled + "<br>");
        document.write(navigator.platform + "<br>");
        document.write(navigator.userAgent + "<br>");
    </script>
</body>
</html>
```

THE `WINDOW` OBJECT

- Represents the current window.

- There are possible many objects of type **`Window`**, the predefined object **`window`** represents the current window.
- Access to, and control of, a number of properties including position and size.

WINDOW ATTRIBUTES

- `document`
- `name`
- `status` the status line
- `parent`

Some window Methods

```
alert()
close()
prompt()
moveTo()  moveBy()
open()
scroll()  scrollTo()
resizeBy()    resizeTo()
```

THE MATH OBJECT

- Access to mathematical functions and constants.
- Constants: **`Math.PI`**
- Methods:

  `Math.abs(), Math.sin(), Math.log(), Math.max(), Math.pow(), Math.random(), Math.sqrt(), …`

MATH OBJECT IN USE

```
// returns an integer between 1 and 6
function roll() {
  var x = Math.random();

  // convert to range [0,6.0)
  x = x * 6;
  // add 1 and convert to int
  return parseInt(1+x );
}

document.writeln("Roll is “ + roll() );
```

ARRAY OBJECTS

- Arrays are supported as objects.
- Attribute **`length`**
- Methods include:

  `concat join pop push reverse sort`

SOME SIMILARITY TO C++

- Array indexes start at 0.
- Syntax for accessing an element is the same:

```
          a[3]++;
      blah[i] = i*72;
```

NEW STUFF (DIFFERENT THAN C++)

- Arrays can grow dynamically – just add new elements at the end.
- Arrays can have *holes*, elements that have no value.
- Array elements can be anything
  - numbers, strings, or arrays!