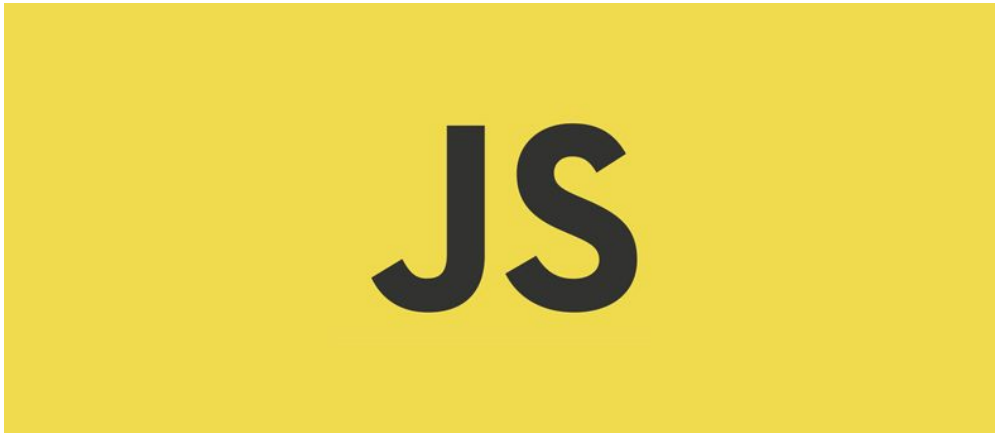


Javascript II : Object and Array

Arrays

- Arrays are used when we are storing a list of values that might be related to each other.
- You can mix different data types into a single array.

Try it: Array

```
// Create an array and assign it values.
var colors;
colors = ['white', 'black', 'custom'];

// Show the first item from the array.
console.log(colors[0]);
```

Important Methods in Array

Assessing an Array: You refer to an array element by referring to the **index number** written in **square brackets**.

This statement accesses the value of the first element in **colors** and changes the value of the second element.

colors[0] -> Get the first item in an array.

Index in array start with **0**.

Attempting to access an index outside of the array, returns the value **undefined**.

Important methods in Array

length : colors.length -> Get how many items are there in an array.

push() : Adds one or more elements to the end of an array and returns the new length of the array.

pop() : Removes the last element from an array and returns that element.

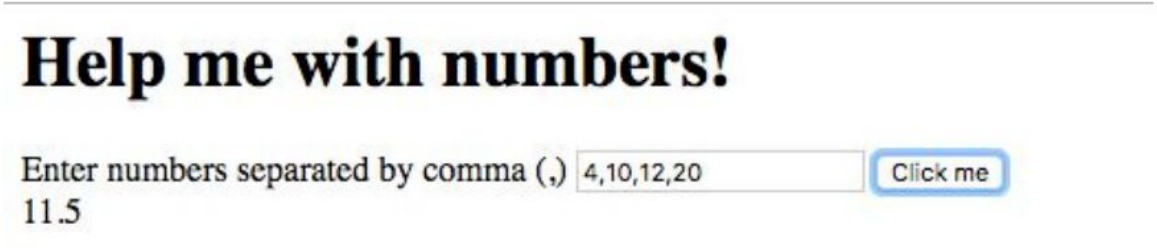
Try it: Array- Things to remember

- Create an array with 4 places that you have visited.
- Print the second item inside the array.
- Print the number of items inside the array using the appropriate method.
- Iterate the array and print all items using one of the control flow seen in class.

Other Array Methods

METHOD	DESCRIPTION
reverse()	Reverses the order of the elements of an array.
shift()	Removes the first element from an array and returns that element.
unshift()	Adds one or more elements to the front of an array and returns the new length of the array.
sort()	Sorts the elements of an array in place and returns the array.
indexOf()	Returns the first (least) index of an element within the array equal to the specified value, or -1 if none is found.

Exercise : Array of numbers



Exercise Instruction

1) Create a website that will contain a form and a button.

2) When the button is pressed, you will retrieve the value from the form, and return the lowest value, highest value and it’s average.

3) User will separate each number using ‘,’.

Hint:

1)Use button and apply onclick to the calculation method, eg:

<button onclick="submit()">Click me</button>

2) To get the value of a form, use value property.

3) Use Number() function to transform String into Number

Bring it all

Return the provided string with the first letter of each word capitalized. Make sure the rest of the word is in lowercase.

For the purpose of this exercise, you should also capitalize connecting words like "the" and "of".

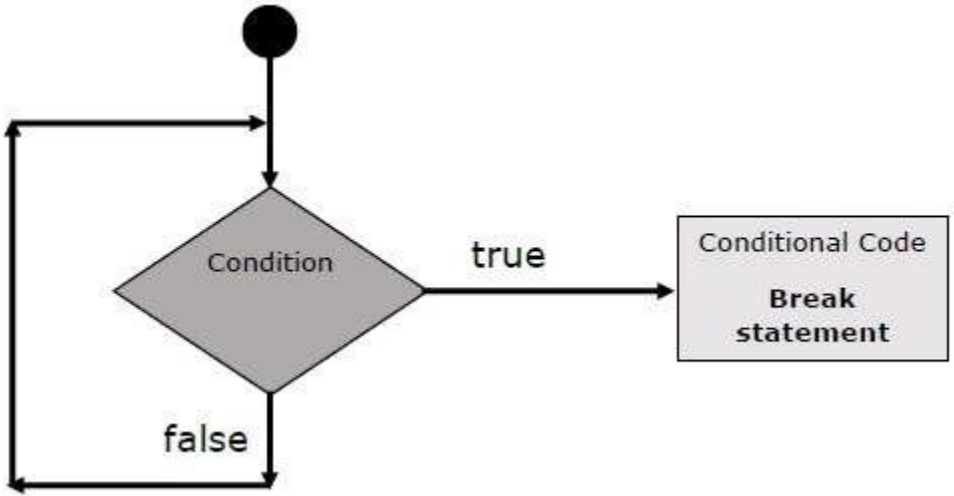
- `titleCase("I'm a little tea pot")` should return "I'm A Little Tea Pot".
- `titleCase("sHoRt AnD sToUt")` should return "Short And Stout".
- `titleCase("HERE IS MY HANDLE HERE IS MY SPOUT")` should return "Here Is My Handle Here Is My Spout".

Read last week String notes for reference.

While loop

The purpose of a **while** loop is to execute a statement or code block repeatedly as long as an **expression** is true. Once the expression becomes **false,** the loop terminates.

While loop : Control flow.



```
var i = 1;       // Set counter to 1
var msg = '';    // Message

// Store 5 times table in a variable
while (i < 10) {
  msg += i + ' x 5 = ' + (i * 5) + '<br />';
  i++;
}
console.log(msg);
```

Do While loop

The key difference between a **while** loop and a **do while** loop is that the statement in the code come before the condition.

The statement will at least be executed once even if the condition has never be met.

```
var i = 1;       // Set counter to 1
var msg = '';    // Message


do {
  msg += i + ' x 5 = ' + (i * 5) + '<br />';
  i++;
} while (i < 1);

console.log(msg);
```

What is an Object?

Object is a group of variable and functions to create a model of something that can be recognized from the real world.

If a variable is part of an object, it is called the property of the object.

If a function is part of an object, it is called the method of an object.

Creating an Object: Literal Notation

```
var hotel = {

name: ‘Quay’,

rooms: 40,

booked: 25,

checkAvailability : function () {

return this.rooms - this.booked;

}

};
```

Accessing an object and dot notation

There are two ways to access a property, using **dot** notation or using bracket syntax.

Eg:

var hotelName = hotel.name;

var roomsFree = hotel.checkAvailability;

var hotelName = hotel['name'];

var roomsFree = hotel['checkAvailability()'];

Updating an object

To update the value of properties, we use **dot notation** or **square brackets**.

Eg:

hotel.name = ‘Park’;

hotel[‘name’] = ‘Park’;

Creating many objects: Constructor Notation

You may create different object using the same blueprint/template.

Different object can be created using the template.

Try It: Creating Object Constructor

```
function Hotel(name, rooms, booked) {

this.name = name;

this.rooms = rooms;

this.booked = booked;

this.checkAvailability = function() {

return this.rooms - this.booked;

};

}
```

Object Manipulation

Creating new object:

var quayHotel = new Hotel(‘Quay’, 40, 25);

var parkHotel = new Hotel(‘Park’, 120, 77);

Reading values:

console.log(quayHotel.name)

console.log(parkHotel.name)

Object Manipulation

Adding properties

quayHotel.gym = true;
quayHotel.pool false;

Removing properties

delete hotel.booked;

this Keyword

The keyword **this** is used inside functions and objects.

When the function **this** is created at the top level of a script, then it is in the **global scope.**

If the function is defined inside a function, the **this** refers to the containing object.

The default object is the **window** object. When **this** is used inside the global context, it refers to the **window** object.

Example : this Keyword

```
var shape = {

width: 600,

height: 400,

getArea: function() {

return this.width * this.height;

}

}
```

Recap: Arrays, Objects, Arrays of Objects, Objects of Array

An Object:

```
costs = {

room1: 420,

room2: 460,

room3: 230,

room4: 620

};
```

Recap: Arrays, Objects, Arrays of Objects, Objects of Array

Objects of Array

var info =

{name: ‘Casey’, rate: 70, active:true , emergency : [

{name: ‘Jason’, phone:’0123456789’},

{name: ‘Joseph’, phone:’01298765432’’}

]}

Recap: Arrays, Objects, Arrays of Objects, Objects of Array

An Object:

```
costs = {

room1: 420,

room2: 460,

room3: 230,

room4: 620

};
```

Exercise : Object Oriented programming

1) Create a Book Class from the model presented on previous page.

2) function read will increase the number of pages.

3) function buy will show the following message in browser “<Book title>, by

<Author> has been purchased!”

4) Create a form to add the book into an array.

5) Create a button that will add the book in Array and another button to show it inside your website.

Exercise: Object Oriented Programming

What are Built-in Objects?

Browsers come with a set of built-in objects that you can manipulate. You can take advantage of these objects as a toolkit to create an interactive web pages.

Browser Object Model: Contains objects that represent current browser window or tab, such as browser history and device's screen.

Document Object Model: Contains representation of current page.

Global Javascript Objects: Things that JavaScript language needs to create model of.

Browser Object Model

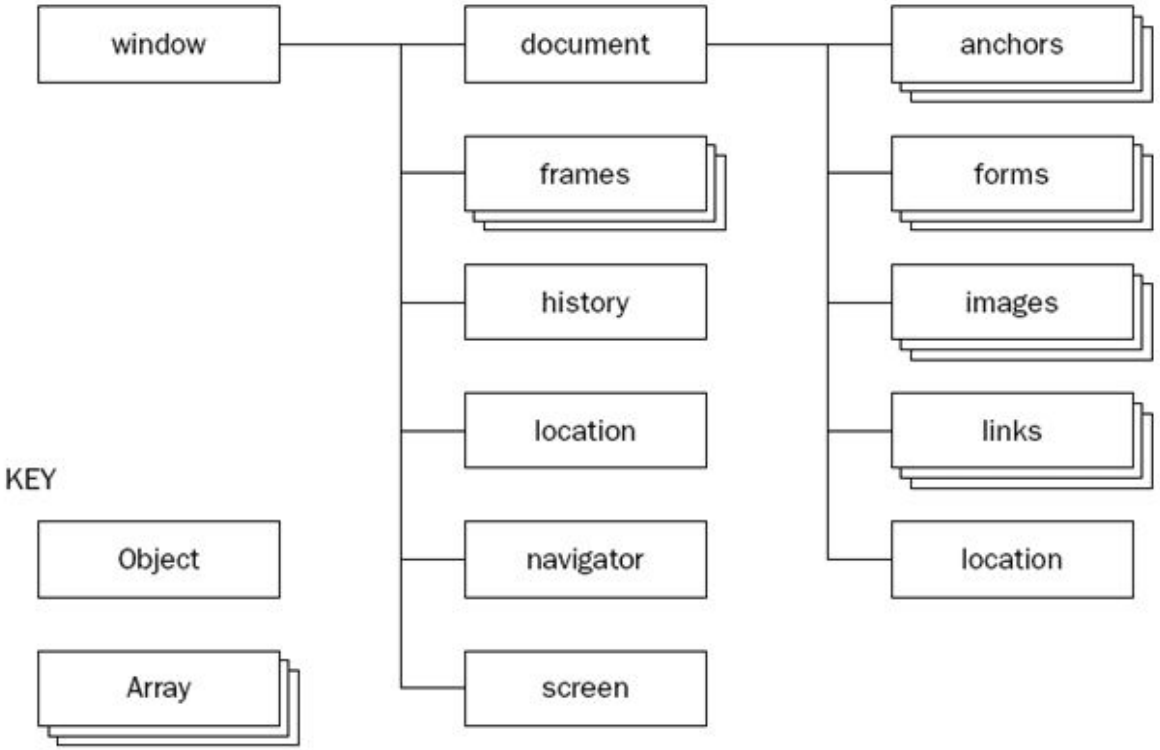


Figure 5-3

Browser Object Model (property)

Property	Description
window.innerHeight/innerWidth	Get the window's height and width (NOT including toolbars/scrollbars)
window.pageXOffset/pageYOffset	The pixels the current document has been scrolled from the upper left corner of the window, horizontally and vertically.
screenX, screenY	Return the x and y coordinates of the new window relative to the screen:
window.location	Current URL of window object
window.history	Reference to history object for browser window or tab
window.screen	Reference to screen object, you may use width and height property to get the value of the window.

Browser object Model (method)

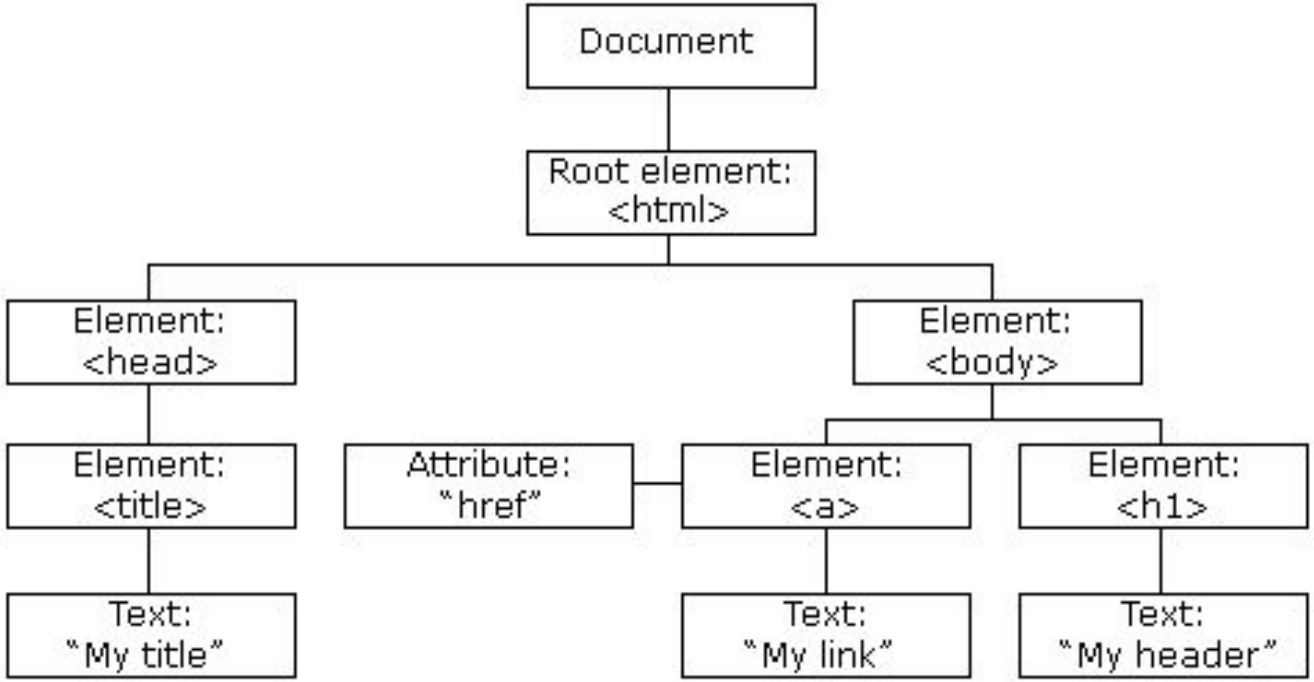
Method	Description
window.alert()	Create dialog box with message
window.open()	Opens new browser specifies with the URL.
window.print()	Print the current page.
window.prompt()	Prompt an alert for user to fill in with information.

Try It: Browser Object Model

Show the following

1) Height and Width of the browser window.
2) Height and Width of the screen.
3) Number of items in history
4) Show an alert of current URL

Document Object Model



Document Object Model : Property

Property/Method	Description
document.title	Return title of current document
document.lastModified	Return date it was last modified
document.URL	Return URL string of current document
document.domain	Return domain of current document.

Document Object Model : Method

Property/Method	Description
document.write()	Writes text to document
document.getElementById()	Returns element with specified ID
document.querySelectorAll()	Returns a list elements that match a selector

Try It: Document Object Model

Do the following operation

1) Get the title of the page
2) The Page address
3) The last modified date.

More Numbers

Property/Method	Description
isNan()	Checks if the value is not a number
toFixed()	Rounds to specified number of decimal places (returns a string)
toPrecision()	Rounds to total number of places (returns a string)
toExponential()	Returns a string to total number of places

Try It : More Numbers

Create a number, eg: 10.23456789;

1) Get the number to 3 decimal places.
2) Return the number to 3 number of places.

Date and Time

To work with dates, you create an instance of the **Date** object constructor.

var today = new Date();

You may also set the date and time according to format wishes:

var dob = new Date(1996,11,26,15,45,55);

var dob = new Date('Dec 26, 1996 15:45:55');

var dob = new Date(1996,11,26);

Date and Time Object

Method	Description
References	
getDate()	Get the day as a number (1-31)
getDay()	Get the weekday as a number (0-6)
getFullYear()	Get the four digit year (yyyy)
getHours()	Get the hour (0-23)
getMilliseconds()	Get the milliseconds (0-999)
getMinutes()	Get the minutes (0-59)
getMonth()	Get the month (0-11)
getSeconds()	Get the seconds (0-59)
getTime()	Get the time (milliseconds since January 1, 1970)

Try It: Date and Time

```
var today = new Date();

var year = today.getFullYear();

var firstClass = new Date('Aug 13, 2017 09:00:00')

var difference = today.getTime()-firstClass.getTime();

 var differenceInDays = difference/86400000

var msg = Math.floor(differenceInDays) + ‘ since we start coding!’;

document.write(msg);
```

Exercise

Create a function that will greet you based on the current time on the system.

If it is between 7 am to 1 pm, you will greet “Good Morning”

From 1pm to 5 pm, you will greet “Good Afternoon”

From 5 pm to 7 pm , you will greet “Good Evening”

Other than that, “Good Night”

Variable scope

- The location where you declare a variable will affect where it can be used within the code.
- If the variable is declared within a function, it can only be used within the function.
- This is also known as variable scope.

Local variables vs Global Variable

Local Variable

If a variable is created inside a function, it can only be used inside that function. It is called **local variable** and have **local scope**. It cannot be accessed outside of the function.

Global Variable

If a variable is created outside a function, it can be used anywhere within the script. It is called **global variable** and has **global scope.**

Try this: Local vs global variable

```
var x = "global";
function f() {
    var x = "local";
    console.log(x); // local
}
f();
console.log(x);
```

Surprise Quiz!

Create an element that will show dynamically the Copyright year in the footer , eg this year it will show Copyright 2017, next year automatically change to 2018