this

Keyword

Refers to the object itself

```
var width=600;
var shape={width:300};
var showWidth=function() {
document.write(this.width);
};
showWidth();
```

JavaScript Var Scope

Variable Scope

Local	inside the function	local scope	< memory
Global	outside of the function	global scope	>memory

```
function getArea (width, height) {
        var area=width*height;
        return area;
}
var wallOne=getArea(3,5);
Document.write(wallOne);
```

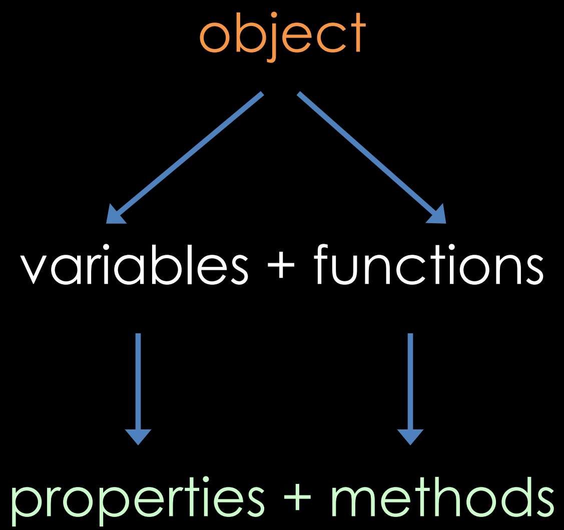
Variable Scope

Exercise:

Create a function to calculate and write in the body of the html document the full price, discount and the discounted price

Object

A group of variables and functions ex. hotel object



object

variables + functions

properties + methods

Object

Properties describe the objects (ex. name, # of rooms, etc.)

Methods describe tasks associated with the object (ex. check availability of hotel rooms)

Object

A group of variables and functions ex. hotel object

```
var hotel ={
name: ‘Quay’,
rooms: 40,
booked: 25,
checkAvailability: function() {
return.this.rooms – this.booked;
}
```

Accessing an Object

```
var hotelName= hotel.name;
Var roomsFree=hotel.checkAvailabitlity();


}
```

Accessing an Object

```
var hotel = {
name : 'Quay',
rooms : 40,
 booked : 25,
checkAvailability : function() {
return this.rooms - this.booked;
} }

document.write(hotel.checkAvailability());
```

Constructor

```
var hotel = new Object () ;                                   //properties
hotel.name : 'Quay',
hotel.rooms : 40,
hotel.booked : 25,
hotel.checkAvailability : function() {                        //method
return this.rooms - this.booked;
} }

document.write(hotel.checkAvailability());
```

Constructor

New keyword

Object () constructor

Constructor

hotel . name = ‘Park’;

Object . property = value

this

Keyword

Refers to the object itself

```
var width=600;
var shape={width:300};
var showWidth=function() {
document.write(this.width);
};
showWidth();
```

Constructor – function3.html

```
function Hotel(name, rooms, booked)
 { this.name = name;
this.rooms = rooms;
this.booked = booked;
this.checkAvailability = function() {
return this.rooms - this.booked; }; }

var quayHotel = new Hotel('Quay', 40, 25);
var parkHotel = new Hotel('Park', 120, 77);
```

Objects

Browser

object model

JavaScript

Global objects

Document

Object model

(DOM)

Representation

of the current

wen page

Browser object model

window	current browser window
document	current webpage
history	pages in browser history
location	URL of current page
navigation	info about browser
screen	device's display info

window.screen.width;	returns the width of the device's screen in pixels

Browser object model - properties

innerHeight	Returns the inner height of a window's content area
innerWidth	Returns the inner width of a window's content area
location	Returns the Location object for the window
pageXOffset	Returns the pixels the current document has been scrolled (horizontally) from the upper left corner of the window
pageYOffset	Returns the pixels the current document has been scrolled (vertically) from the upper left corner of the window

Browser object model - properties

screen	Returns the Screen object for the window
screenX	Returns the x coordinate of the window relative to the screen
screenY	Returns the y coordinate of the window relative to the screen

Browser object model - methods

alert()	opens dialog box
open()	opens new browser window
print()	prints content of the webpage

Browser object model – window_object.html and .js

.js	external JavaScript file
.css	external styles file
.html	HTML document

Document object model

document current webpage

<html>

<head>

<body>

<p>

document.getElementById() method gets element by the value of its ID attribute

Document object model - properties

document.ttitle	title of the doc
document.lastModified	date on which dopc was last modified
Document.URL	returns string with URL of the doc
document.domain	returns domain of the current doc

Document object model - methods

document.write()	writes text to docuemnt
document.getElementById()	returns element with matching ID
document.createElement()	creates new element
document.createTextNode()	creates new text node

Document object model – document_object.html &js

Document – object.html

. Js

Global JavaScript objects

String	for working with string values
Math	for working with numbers and calculations
Date	to represent and handle dates

Math.PI();	Math’s object PI property will return the value of PI

Global JavaScript objects - String

```
var saying='Home sweet home';
```

toUpperCase()	changes string to upper case
toLowerCase()	changes string to lower case

```
saying.toUpperCase();       'HOME SWEET HOME'
document.write(saying);
```

JavaScript String Reference W3Schools.com

http://www.w3schools.com/jsref/jsref_obj_string.asp

Global JavaScript objects - String

Property:

length	Returns the length of a string

Methods:

charAt()	Returns the character at the specified index (position)
concat()	Joins two or more strings, and returns a new joined strings
fromCharCode()	Converts Unicode values to characters

Global JavaScript objects - Math

Property:

PI	Returns pi

Methods:

Math.round()	rounds number to the nearest integer
Math.sqrt()	returns square root of the positive number
Math.ceil()	rounds number up to the nearest integer
Math.floor()	rounds number down to the nearest integer
Math.random()	generates a random number between 0 & 1

JavaScript Math Reference   W3Schools.com

Global JavaScript objects - Math

```
// Create a variable to hold a random number between 1 and 10
var randomNum = Math.floor((Math.random() * 10) + 1);


// Create a variable called el to hold the element whose id attribute
has a value of info
var el = document.getElementById('info');
// Write the number into that element
el.innerHTML = '<h2>random number</h2><p>' + randomNum +
'</p>';
```

Global JavaScript objects – Math- random.html

```
<html>
<head>
<title>Random Script</title>
<script>
var myPix = new Array(”image1.jpg",”image2.jpg",”image3.jpg”)
```

School of Design

Global JavaScript objects – Math- random.html

```
function choosePic() {
     if(document.images) {
       randomNum = Math.floor(Math.random()*myPix.length)
       document.sample.src=myPix[randomNum]
          }
    }
</script>
```

Global JavaScript objects – Math- random.html

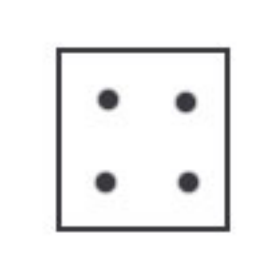
```
</head>
<body onLoad="choosePic()">


<img src="image1.jpg" name =”sample">
</body>
</html>
```

Global JavaScript objects – dice.html

Write a script that simulates rolling a pair of dice when the user clicks on the “roll the dice” link.

You can simulate rolling one die by choosing one of the images 1, 2, 3, 4, 5, or 6 at random. The number you pick represents the number on the die after it is rolled.



Functions : bgcolor.html

```
<html>
<head>
<title>bgcolor_change</title>
<script language="JavaScript”>
function newbg(thecolor)
{
document.bgColor=thecolor;
}
</script>
</head>
```

Functions : bgcolor.html

```
<body textcolor="black" link="black" alink="black”>
<center>
<a href="#" onmousedown="newbg(’olive');"> olive</a><br />
<a href="#" onmousedown="newbg('blue');"> blue</a><br />
<a href="#" onmousedown="newbg('Beige');"> beige</a><br />
<a href="#" onmousedown="newbg('yellow');">yellow</a><br />
</center>
</body>
</html>
```

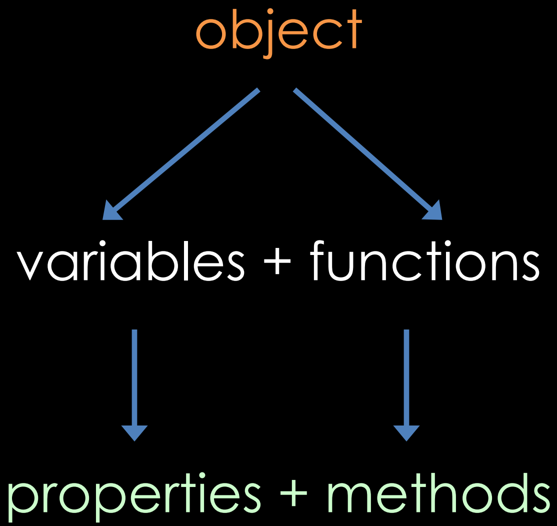
Math object - exercise

Write a script that randomly changes back ground color on the webpage when the user clicks on the “change background color” link.

Use hexadecimal color system and Math object random function

Object

A group of variables and functions ex. hotel object



object

variables + functions

properties + methods

Object

Properties describe the objects (ex. name, # of rooms, etc.)

Methods describe tasks associated with the object (ex. check availability of hotel rooms)

Objects

Browser

object model

JavaScript

Global objects

Document

Object model

(DOM)

Representation

of the current

wen page

Global JavaScript objects

String	for working with string values
Math	for working with numbers and calculations
Date	to represent and handle dates

Math.PI(); Math's object PI property will return the value of PI

Date object

Date to represent and handle dates

```
var today = new Date();  //holds today's date and the current time
```

School of Design

Date object - methods

getDate()	Get the day as a number (1-31)
getDay()	Get the weekday 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)

Date object - methods

getTimezoneOffset()	Returns time zone offset in mins for locale
toDateString()	Returns “date” as a human-readable string
toTimeString()	Returns “time” as a human-readable string
toString()	Returns a string representing a specified date

Date object - methods

toDateString() Returns “date” as a human-readable string

Mon Sept 15 2014.

Date object - methods

```
// Create a variable to hold a new Date object (defaults to now)
 var today = new Date();
// Create a variable to hold the year this year
var year = today.getFullYear();
// Create a variable called el to hold the element whose id attribute
//has a value of footer
var el = document.getElementById('footer');
// Write the year into that element.
el.innerHTML = '<p>Copyright &copy;' + year + '</p>';
```

Date object - methods

```
<!DOCTYPE html>
<html>
  <head>
    <title>Objects - Date Object</title>
    <link rel="stylesheet" href="css/c03.css" />
  </head>
  <body>
    <h1>TravelWorthy</h1>
    <div id="footer"></div>
    <script src="js/date-object.js"></script>
  </body>
</html>
```

Date object - methods

Exercise – add copyright note to your website using Date object

Date object - methods

```
// Create a variable to hold a new Date object (defaults to now)
var today = new Date();
// Get the year this year
var year = today.getFullYear();
// Set the date that the company was established
var est = new Date('Apr 16, 1996 15:45:55');
// Get difference between then & now in milliseconds
var difference = today.getTime() - est.getTime();
// Divide by number of milliseconds to get years
difference = (difference / 31556900000);
```

Date object - methods

```
// Create a variable called elMsg to hold the element whose id
//attribute has a value of message
var elMsg = document.getElementById('message');

// Write the message into that element.
elMsg.textContent = Math.floor(difference) + ' years of online travel advice';
```

Date object - methods

```
<!DOCTYPE html>
<html>
  <head>
    <title>Date Object Difference</title>
    <link rel="stylesheet" href="css/c03.css" />
  </head>
  <body>
    <h1>TravelWorthy</h1>
    <div id="message">Established 1945</div>
    <script src="js/date-object-difference.js"></script>
  </body>
</html>
```

Date object - methods

Exercise – write a function to return your friend’s age

Objects

Functions are set of related statements together that represent a single task

Functions can take parameters (information required to do their job) and may return a value

An object is a series of variables and functions grouped together that represent something from the world around you

In an object, variables are known as properties and functions as methods of the object

Objects

Web browsers implement objects that represent both the browser window and the document loaded into the browser window

JavaScript has several built-in objects such as String, Math, Number and Date.

Arrays and objects can be used to create complex data sets

Objects – Day of the week example

```
<!DOCTYPE html>
<html>
<body>
<p>You can use an array to display the name of the weekday:</p>
<p id="demo"></p>
<script>
var d = new Date();
var days =
["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"]
;
document.getElementById("demo").innerHTML = days[d.getDay()];
</script>
</body>
</html>
```

Objects – example.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Objects - Example</title>
    <link rel="stylesheet" href="css/c03.css" />
  </head>
  <body>
    <h1>TravelWorthy</h1>
    <div id="info">
      <h2>latest hotel offer</h2>
```

Objects – example.html

```
<div id="hotelName"></div>
      <div id="roomRate"></div>
      <div id="specialRate"></div>
      <p>room rate when you book 2 or more nights</p>
      <div id="offerEnds"></div>
    </div>
    <script src="js/example.js"></script>
  </body>
</html>
```

Objects – example.html

```
(function() {
 // PART ONE: CREATE HOTEL OBJECT AND WRITE OUT THE OFFER DETAILS
 // Create a hotel object using object literal syntax
 var hotel = {
   name: 'Park',
   roomRate: 240, // Amount in dollars
   discount: 15,  // Percentage discount
   offerPrice: function() {
     var offerRate = this.roomRate * ((100 - this.discount) / 100);
     return offerRate;
   }
 }
```

Objects – example.html

```
 // Write out the hotel name, standard rate, and the special rate
 var hotelName, roomRate, specialRate;
// Declare variables
 hotelName = document.getElementById('hotelName');
 // Get elements
 roomRate = document.getElementById('roomRate');
 specialRate = document.getElementById('specialRate');
 hotelName.textContent = hotel.name;
// Write hotel name
 roomRate.textContent =  '$' + hotel.roomRate.toFixed(2);
// Write room rate
 specialRate.textContent = '$' + hotel.offerPrice();     // Write offer price
```

Objects – example.html

```
// PART TWO: CALCULATE AND WRITE OUT THE EXPIRY DETAILS FOR THE OFFER
var expiryMsg; // Message displayed to users
var today;     // Today's date
var elEnds;    // The element that shows the message about the offer ending
function offerExpires(today) {
  // Declare variables within the function for local scope
  var weekFromToday, day, date, month, year, dayNames, monthNames;
  // Add 7 days time (added in milliseconds)
  weekFromToday = new Date(today.getTime() + 7 * 24 * 60 * 60 * 1000);
```

Objects – example.html

```
// Create arrays to hold the names of days / months
  dayNames = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday',
'Friday', 'Saturday'];
  monthNames = ['January', 'February', 'March', 'April', 'May', 'June', 'July',
'August', 'September', 'October', 'November', 'December'];
  // Collect the parts of the date to show on the page
  day = dayNames[weekFromToday.getDay()];
  date = weekFromToday.getDate();
  month = monthNames[weekFromToday.getMonth()];
  year = weekFromToday.getFullYear();
  // Create the message
  expiryMsg = 'Offer expires next ';
  expiryMsg += day + ' <br />(' + date + ' ' + month + ' ' + year + ')';
  return expiryMsg
```

Objects – example.html

```
today = new Date();                              // Put today's date in variable
elEnds = document.getElementById('offerEnds');  // Get the offerEnds element
elEnds.innerHTML = offerExpires(today);         // Add the expiry message

// Finish the immediately invoked function expression
}());
```

Objects – example.html

Exercise- write a script to display the restaurant schedule for next week using Date and Time object

Functions, Variables, Objects

Functions are set of related statements together that represent a single task

Functions can take parameters (information required to do their job) and may return a value

An object is a series of variables and functions grouped together that represent something from the world around you

In an object, variables are known as properties and functions as methods of the object

Functions, Variables, Objects

Web browsers implement objects that represent both the browser window and the document loaded into the browser window

JavaScript has several built-in objects such as String, Math, Number and Date.

Arrays and objects can be used to create complex data sets