Object

- In JavaScript an object is a composite type which aggregates several properties and functions (methods).
- Property – Entity with a name and a value.
- Object properties work like variables and fundamentally are variables. You can assign values to them and read values from them.
- A property value can be any data type we have seen, including objects.
- We access object properties and methods using a period (e.g., window.prompt).

JavaScript Window Object

- Represents the window displaying the document (either the web browser window or a frame within a window).
- Some Window Object Properties:
  - status – read/write string representing browser's status line.
  - location - location object for the window (or frame). It represents the URL of the loaded document.
  - closed – represents whether the window has been closed.
  - document – read-only value representing the document object that describes the document associated with the window (or frame).
  - history –read-only reference to object that represents the browsing history.
- **Example:** *WindowObjectI.html*
- Some Window Object methods
  - alert – displays a message in a dialog box.
  - prompt – to input string through dialog box.
  - confirm – yes-or-no question.
  - open – creates a new window.
  - print – equivalent to clicking on the "Print" button.
- **Example:** *WindowObjectII.html*

JavaScript Document Object

- This object represents the HTML document displayed in the window.
- Some Document Object Properties:
  - title
  - lastModified
  - URL
  - forms[] – array of form objects present in the document.
  - images[] – array of Image objects present in the document and associated with the <img> tag.
- **Example:** DocumentObject.html
- Some Document Object methods
  - writeln
  - Several methods to deal with Events
  - getElementById(Id) – returns element with the specified id or null (element does not exist).
  - getElement
- It is through the document object that we will be able to access/modify an html document.
- The way the document content is accessed/modified is referred to as the Document Object Model (DOM).
- Several DOMs ☹
  - W3C DOM
  - IE4 DOM

Forms

- **Forms** - means by which information passes from the user to a server.
- For now we will use forms to read values to be processed by our JavaScript programs.
- **<form> tag**
  - Defines the form.
  - It has two attributes: action and method
  - **action** – indicates where the form contents will be sent when the form is submitted.
  - **method** – defines how the contents will be sent (post/get).
- **<input>tag**
  - Appears inside of the <form> tag.
  - Defines several input data alternatives.
  - The general format is: **<input type="ALTERNATIVE" />**
  - **ALTERNATIVE** can be text, password, checkbox, radio, file, submit, image, button, reset, hidden
- We have can several forms in our document.
- **Example:** Forms.html

Additional Form Elements

- Reset
  - Allow us to restore the default values of form elements.
- Password
  - Similar to a text box but text is hidden.
- Example: ResetPassword.html

Timer

- setInverval
  - Allow us to schedule an activity at a particular interval
- **Example:** TestudoAnimation.html

Global Variables

- Global Variables
  - Variables defined outside of any function.
- We want to avoid using global variables. Why?
- **Example:** TestudoAnimation.html
- **Example:** TestudoSizeAnimation.html

Conventions to Use From Now On

- **Variable/Function names**
  - We will use lowercase.
  - If multiple words are associated with a variable name then capitalize the first letter of second word on.
    - waterTemperature
    - globalWarmingIndex
- **Curly Brackets**
  - Use a particular style.
- **Comparisons**
  - Use === rather than ==
- **No global variables!**
- **Good indentation**

HTML Forms

- Forms - means by which information passes from the user to a server.
- <form> tag - defines the form. It has two attributes:
  - **action** – indicates where the form contents will be sent when the form is submitted
  - **method** – defines how the contents will be sent when the submit button is clicked on
    - post – contents sent using the HTTP POST method . Content is "hidden".
    - get – contents sent using the HTTP GET method. Contents included in the URL.
- $_POST or $_GET
  - Allow us to retrieve form information in the target script.
- **Example:** formsSummaryPost.html, formsSummaryGet.html
- Advantages/Disadvantages of POST and GET

Form Processing

- As we have seen, we need a server script/program to process the data we collect in a form.
- If you know the script/program to use you can develop your own customized form to provide a service (e.g., Google search).
- Make sure you name your form elements after the variables required by the script/program.
- Notice we are not using any JavaScript at the moment. We are focusing on what we can do with forms by relying on the server script/program for processing.
- **Example:** OurSearchBox.html

Passing Data To Functions

- Reviewing passing by value with numbers.
  - **Example:** PassByValue.html
- How is an array passed to a function?
  - **Example:** PassingArrays.html
- Let's see a Memory Diagram.
- Memory Diagram
  - Tool we will use to illustrate the associations between variables and entities (e.g., objects, arrays, etc.).