CS 106: Web Technology - I Assignment #3

Anup Adhikari
anup.adhikari@gandakiuniversity.edu.np

Gandaki University January 30, 2023

1 Introduction to Javascript

1.1 What is Scripting?

What is scripting?

A script is program code that doesn't need pre-processing (e.g. compiling) before being run. In the context of a Web browser, scripting usually refers to program code written in JavaScript that is executed by the browser when a page is downloaded, or in response to an event triggered by the user.

Scripting can make Web pages more dynamic. For example, without reloading a new version of a page it may allow modifications to the content of that page, or allow content to be added to or sent from that page. The former has been called DHTML (Dynamic HTML), and the latter AJAX (Asynchronous JavaScript and XML).

Beyond this, scripts increasingly allow developers to create a bridge between the browser and the platform it is running on, making it possible, for example, to create Web pages that incorporate information from the user's environment, such as current location, address book details, etc.

This additional interactivity makes Web pages behave like a traditional software application. These Web pages are often called Web applications and can be made available either directly in the browser as a Web page, or can be packaged and distributed as Widgets.

1.2 What scripting interfaces are available ?

The most basic scripting interface developed at W3C is the DOM, the Document Object Model which allows programs and scripts to dynamically access and update the content, structure and style of documents. DOM specifications form the core of DHTML.

Modifications of the content using the DOM by the user and by scripts trigger events that developers can make use of to build rich user interfaces.

A number of more advanced interfaces are being standardized, for instance:

- XMLHttpRequest makes it possible to load additional content from the Web without loading a new document, a core component of AJAX,
- the Geolocation API makes the user's current location available to browser-based applications,
- several APIs make the integration of Web applications with the local file system and storage seamless.

DOM

The Document Object Model (DOM) is an application programming interface (API) for valid HTML. It defines the logical structure of documents and the way a document is accessed and manipulated. In the DOM specification, the term "document" is used in the broad sense - increasingly, XML is being used as a way of representing many different kinds of information that may be stored in diverse systems, and much of this would traditionally be seen as data rather than as documents. Nevertheless, XML presents this data as documents, and the DOM may be used to manage this data.

```
<TABLE>
<TBODY>
<TR>
<TD>Shady Grove</TD>
<TD>Aeolian</TD>
</TR>
<TR>
<TD>Over the River, Charlie</TD>
<TD>Dorian</TD>
</TR>
</TBODY>
</TABLE>
```

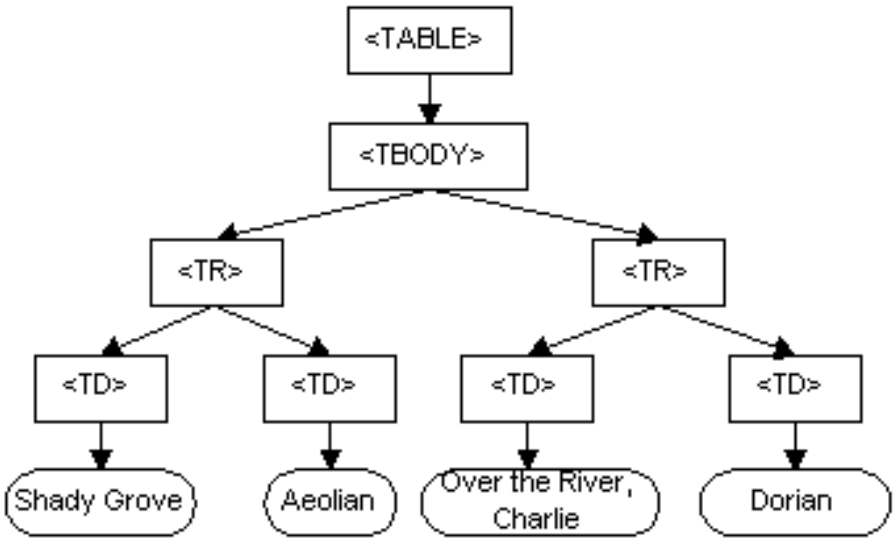

Figure 1: DOM of the example code

1.3 Infrastructure of DOM

1.3.1 Trees

A tree is a finite hierarchical tree structure. In tree order is preorder, depth-first traversal of a tree.

An object that participates in a tree has a parent, which is either null or an object, and has children, which is an ordered set of objects. An object A whose parent is object B is a child of B.

The root of an object is itself, if its parent is null, or else it is the root of its parent. The root of a tree is any object participating in that tree whose parent is null.

An object A is called a descendant of an object B, if either A is a child of B or A is a child of an object C that is a descendant of B.

An inclusive descendant is an object or one of its descendants.

An object A is called an ancestor of an object B if and only if B is a descendant of A.

An inclusive ancestor is an object or one of its ancestors.

An object A is called a sibling of an object B, if and only if B and A share the same non-null parent.

An inclusive sibling is an object or one of its siblings.

An object A is preceding an object B if A and B are in the same tree and A comes before B in tree order.

An object A is following an object B if A and B are in the same tree and A comes after B in tree order.

The first child of an object is its first child or null if it has no children.

The last child of an object is its last child or null if it has no children.

The previous sibling of an object is its first preceding sibling or null if it has no preceding sibling.

The next sibling of an object is its first following sibling or null if it has no following sibling.

The index of an object is its number of preceding siblings, or 0 if it has none.

Features of DOM Module

1. Core Module
2. XML Module
3. HTML Module
4. Views Module
5. StyleSheets Module
6. CSS Module
7. *Events Module*
8. *User Interface Events Module*
9. *Mouse Events Module*
10. *Mutation Events Module*
11. Range Module
12. Traversal Module

1.4 Getting started with Javascript

1.4.1 Using Console Tab of Browsers

All the popular web browsers have built-in JavaScript engines. Hence, you can run JavaScript on a browser. To run JavaScript on a browser,

1. Open your favorite browser (here we will use Google Chrome).
2. Open the developer tools by right clicking on an empty area and select Inspect. Shortcut: F12
3. On the developer tools, go to the console tab. Then, write JavaScript code and press enter to run the code.

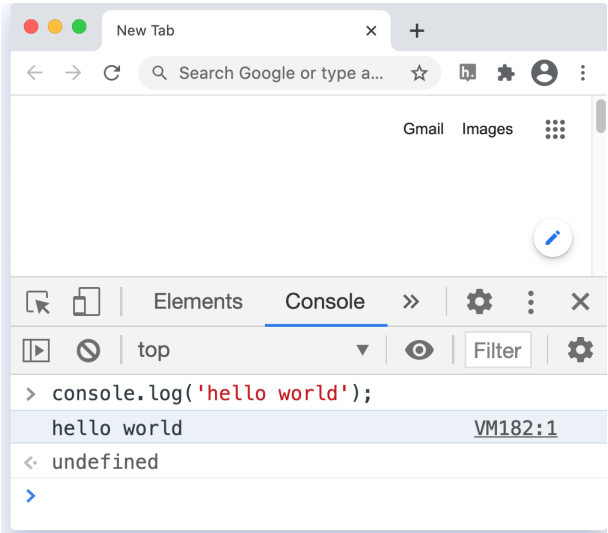

Figure 2: Developer Tools Colsole

1.4.2 By Creating Web Pages

JavaScript was initially created to make web pages interactive, that's why JavaScript and HTML go hand in hand. To run JS from a webpage, follow these steps:

1. Write a HTML document. (index.html)
2. Write a script tag.

```
<script> console.log("Hello BIT2022"); </script>
```

3. Write a script source file (run.js) in script tag.

```
<script src="run.js"></script>
```

run.js

```
1        console.log('Hello BIT2022');
```

2 Lab Objectives

1. To be acquainted with basic JavaScript techniques of writing applications.

3 Lab Exercise

1. Work with JavaScript variable declarations **(var, let)**
2. Initialize Variable with values.
3. Changing the values of variable.
4. Javascript Constants using **const**
5. Using operators to find new values

```
1  =
2  +=
3  -=
4  *=
5  /=
6  %=
7  **=
8  ==
9  !=
10 ===
11 !==
12 >
13 >=
14
15 && Logical AND
16 ||
17 !
18
19 &    Bitwise AND
20 |    Bitwise OR
21 ^    Bitwise XOR
```

```
~   Bitwise NOT
<<  Left Shift
>>  Sign-propagating right shift
>>> Zero-fill right Shift
```

variablesAndConstants.js

```
let x;
x = 5;
let x = 5, y = 6, z = 7;
console.log(x);

x=3;
console.log(x);
```

4 Questions

Question 1

1. Declare a variable x and assign it the value 10. Declare another variable y and assign it the value 5. Write a single line of code that uses the + operator to add x and y and store the result in a new variable z.
2. Write a line of code that uses the - operator to find the difference between x and y.
3. Write a line of code that uses the * operator to multiply x and y.
4. Write a line of code that uses the / operator to divide x by y.
5. Write a line of code that uses the % operator to find the remainder when x is divided by y.
6. Write a line of code that uses the ++ operator to increment the value of x by 1.
7. Write a line of code that uses the – operator to decrement the value of y by 1.
8. Write a line of code that uses the += operator to add 5 to the value of x.
9. Write a line of code that uses the -= operator to subtract 3 from the value of y.
10. Write a line of code that uses the == operator to compare the values of x and y for equality.
11. Write a line of code that uses the === operator to compare the values of x and y for strict equality (meaning that they are both the same value and the same type).
12. Write a line of code that uses the != operator to compare the values of x and y for inequality.
13. Write a line of code that uses the !== operator to compare the values of x and y for strict inequality (meaning that either the values are different or the types are different).
14. Write a line of code that uses the > operator to compare the value of x to the value of y and determine if x is greater than y.
15. Write a line of code that uses the < operator to compare the value of x to the value of y and determine if x is less than y.
16. Write a line of code that uses the >= operator to compare the value of x to the value of y and determine if x is greater than or equal to y.
17. Write a line of code that uses the <= operator to compare the value of x to the value of y and determine if x is less than or equal to y.