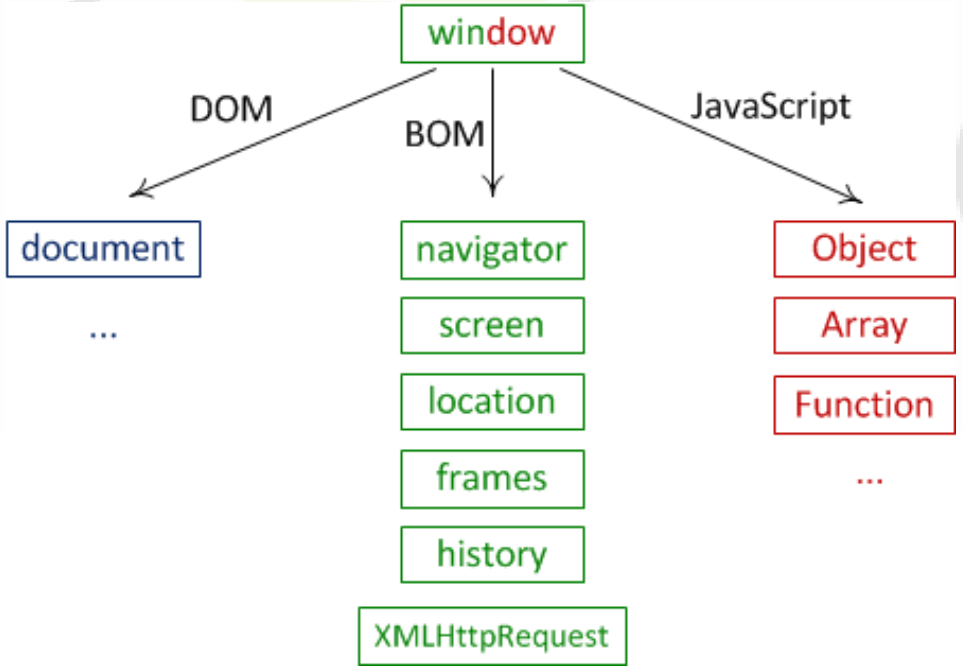
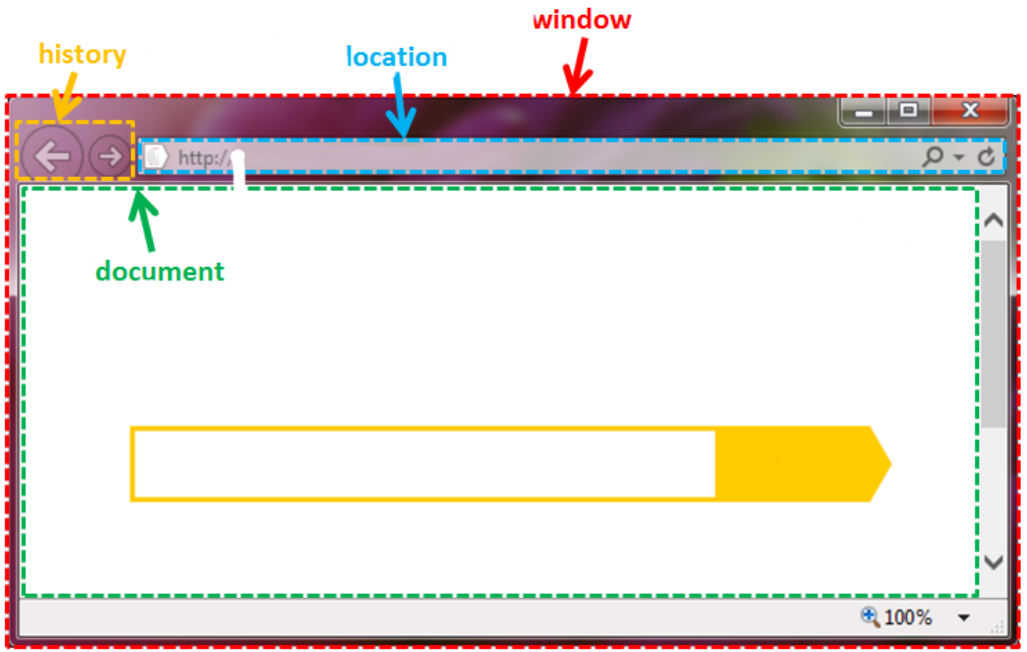
The Browser Object Model

- BOM is browser API through which you can manage browser (current window or tab)
- There is no standard for BOM – each browser has it's own implementation
- BOM can be accessed through main global object Window





Window

- It’s references browser window (tab) or frame
- **Window object** is main **root** JavaScript object.
- All other JavaScript objects, all variables and functions defined by user are stored in Window object.
- If you have reference to **another window** then you can use it to access objects of that window as well.
- You can **skip “window” keyword** during manipulating it’s properties and method

Window - properties

closed - a value indicating whether a window has been closed or not
defaultStatus - Sets or returns the default text in the statusbar of a window
frameElement - the <iframe> element in which the current window is inserted
frames - all <iframe> elements in the current window
innerHeight - the inner height of a window's content area
innerWidth - the inner width of a window's content area
length - the number of <iframe> elements in the current window
localStorage - a reference to the local storage object used to store data. Stores data with no expiration date
name - Sets or returns the name of a window
opener - a reference to the window that created the window
outerHeight - the outer height of a window, including toolbars/scrollbars
outerWidth - the outer width of a window, including toolbars/scrollbars
pageXOffset - the pixels the current document has been scrolled (horizontally) from the upper left corner of the window
pageYOffset - the pixels the current document has been scrolled (vertically) from the upper left corner of the window
parent - the parent window of the current window
screenLeft - the horizontal coordinate of the window relative to the screen
screenTop - the vertical coordinate of the window relative to the screen
screenX - the horizontal coordinate of the window relative to the screen
screenY - the vertical coordinate of the window relative to the screen
sessionStorage - a reference to the local storage object used to store data. Stores data for one session (lost when the browser tab is closed)
scrollX - An alias of pageXOffset
scrollY - An alias of pageYOffset
self - the current window
status - Sets or returns the text in the statusbar of a window
top - the topmost browser window

Window - methods

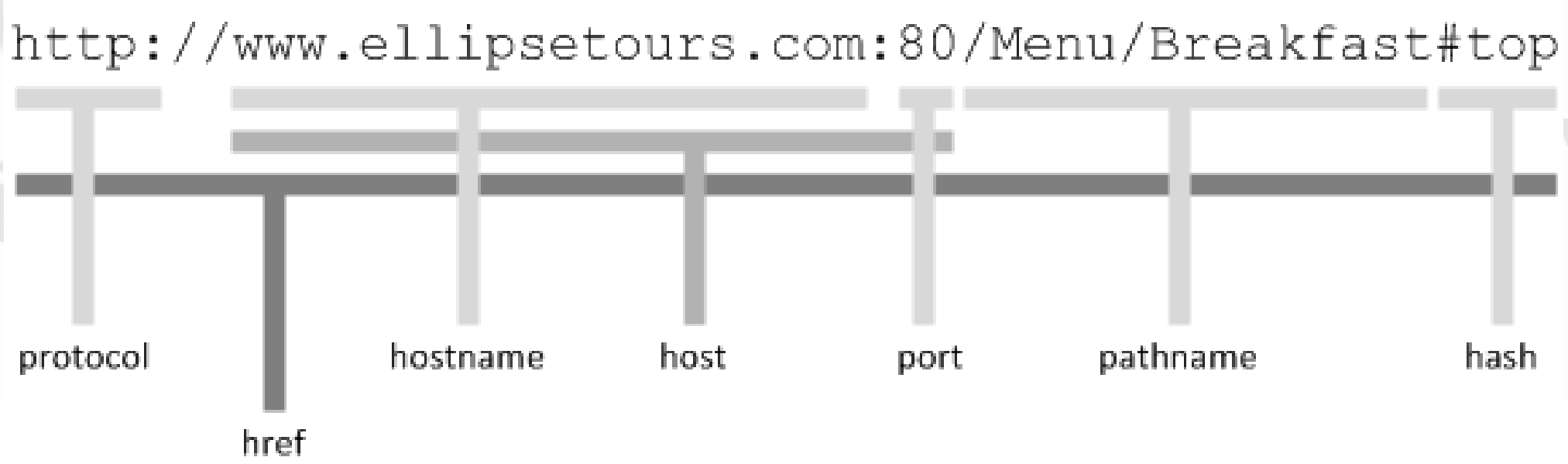
alert() - Displays an alert box with a message and an OK button
atob() - Decodes a base-64 encoded string
blur() - Removes focus from the current window
btoa() - Encodes a string in base-64
clearInterval() - Clears a timer set with setInterval()
clearTimeout() - Clears a timer set with setTimeout()
close() - Closes the current window
confirm() - Displays a dialog box with a message and an OK and a Cancel button
createPopup() - Creates a pop-up window
focus() - Sets focus to the current window
getComputedStyle() - Gets the current computed CSS styles applied to an element
getSelection() - Returns a Selection object representing the range of text selected by the user
matchMedia() - Returns a MediaQueryList object representing the specified CSS media query string
moveBy() - Moves a window relative to its current position
moveTo() - Moves a window to the specified position
open() - Opens a new browser window
print() - Prints the content of the current window
prompt() - Displays a dialog box that prompts the visitor for input
resizeBy() - Resizes the window by the specified pixels
resizeTo() - Resizes the window to the specified width and height
scrollBy() - Scrolls the document by the specified number of pixels
scrollTo() - Scrolls the document to the specified coordinates
setInterval() - Calls a function or evaluates an expression at specified intervals (in milliseconds)
setTimeout() -Calls a function or evaluates an expression after a specified number of milliseconds
stop() - Stops the window from loading

Window – Lab Work – 20 mins

- !! Each action should be done as button.
- Display **all window properties** on the page using **document.write**
- Open new window in new tab: **window.open(*URL,name,specs,replace*)**
- Use code from first point to display properties of new window in itself.
- Resize new window to size 500x500: ***window*.resizeTo(*width,height*)**
- Reduce current window size: ***window*.resizeBy(*width,height*)**
- Scroll new window by 100px horizontally: **window.scrollBy(*xnum,ynum*)**
- Scroll current window to point 200x250: **window.scrollTo(*xpos,ypos*)**
- Give focus to new window and in 3 sec blur focus from it.
- Close new window and try to print current one.

Location

- Current URL address of the document
- window.location === document.location;
- Properties:



- Method **window.location.toString()** returns property **href**

Location - methods

Method	Description
assign()	Loads a new document
reload()	Reloads the current document
replace()	Replaces the current document with a new one

Location – Lab Work

```
function urlArgs() {
        var args = {};
        var query = location.search.substring(1);
        var pairs = query.split("&");

        for(var i = 0; i < pairs.length; i++) {
                var pos = pairs[i].indexOf('=');
                if (pos == -1) continue;
                var name = pairs[i].substring(0,pos);
                var value = pairs[i].substring(pos+1);
                value = decodeURIComponent(value);
                args[name] = value;
        }

        return args;
}
```

Usage:

var args = urlArgs();

Display args on the page in next format:

<full url>

<arg1> – <value1>

<arg2> – <value2> …etc.

Try with 5 different urls.

History

- You are not supposed to mess with browser history!
- **history.length** – a property containing the number of URLs help in the history
- **history.go()** – takes a positive or negative number as an argument to specify how many steps back or forward in the history to go. For example history.go(-4) is equivalent to pressing the *Back* button in the browser 4 times
- **history.back()** - equivalent to pressing the *Back* button in the browser (loads the previous page from the history list)
- **history.forward()** - equivalent to pressing the *Forward* button in the browser (loads the next page from the history list if it exists)

- Note: Since HTML5 however this node contains some interesting new methods like pushState() and popState(), which can be used to maintain browsing history logic in rich Javascript applications.

Navigator

- Tells information about your browser
- Client-sniffing:

```
var browser   = navigator.appName;
var b_version = navigator.appVersion;
var version   = parseFloat(b_version);
```

```
document.write("Browser name: "+ browser);
document.write("<br />");
document.write("Browser version: "+ version);
```

- Properties:
  - **appCodeName**: Returns the code name of the browser.
  - **appName**: Returns the name of the browser .
  - **appVersion**: Returns the version information of the browser.
  - **cookieEnable**: Determines whether cookies are enabled in the browser .
  - **platform**: Returns for which platform the browser is compiled
  - **userAgent**: Returnsthe user-agent header sent by the browser to the server.

Screen

- Properties:

Property	Description
availHeight	Returns the height of the screen (excluding the Windows Taskbar)
availWidth	Returns the width of the screen (excluding the Windows Taskbar)
colorDepth	Returns the bit depth of the color palette for displaying images
height	Returns the total height of the screen
pixelDepth	Returns the color resolution (in bits per pixel) of the screen
width	Returns the total width of the screen

- Methods:

Screen.lockOrientation

Lock the screen orientation (only works in fullscreen or for installed apps)

Screen.unlockOrientation

Unlock the screen orientation (only works in fullscreen or for installed apps)