Lecture 4.1

Window

Topics

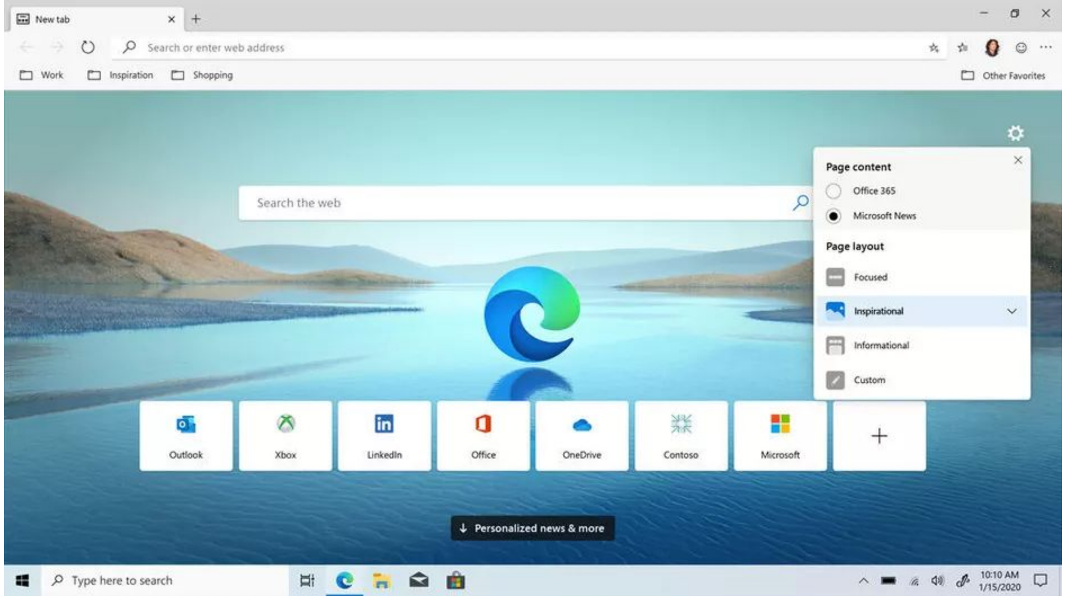
- Intro to the DOM

Window

Window Interface

The **Window** interface represents a window containing a DOM document; the document property points to the DOM document loaded in that window.

A global variable, window, representing the window in which the script is running, is exposed to JavaScript code.



Page content

○ Office 365

● Microsoft News

Page layout

Focused

Inspirational

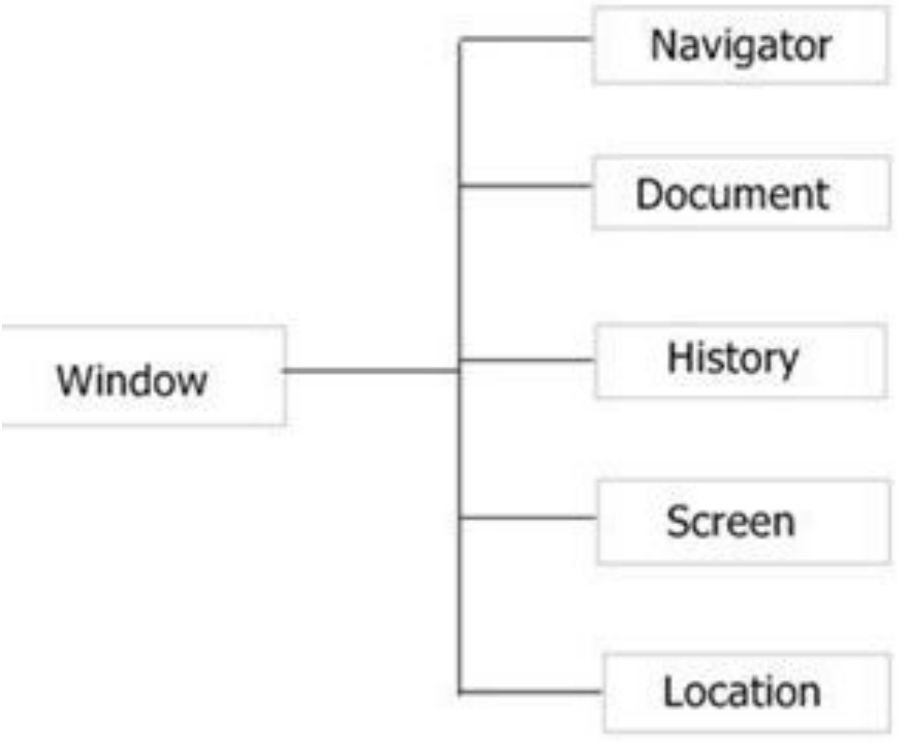
Informational

Custom

Window Object

The Window object is used to open a window in a browser to display the Web page.

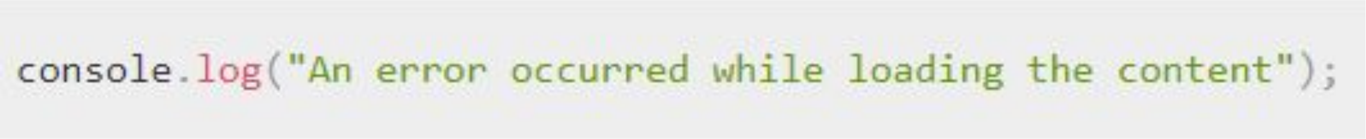
The following figure shows the Window object in the hierarchy of Browsers objects.



Window Object Properties

console

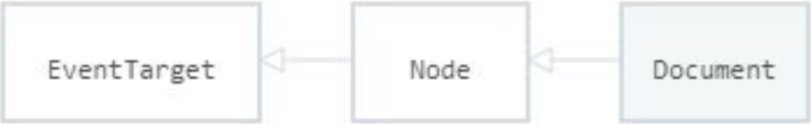
The **Window.console** property returns a reference to the Console object, which provides methods for logging information to the browser's console. These methods are intended for debugging purposes only and should not be relied on for presenting information to end users.

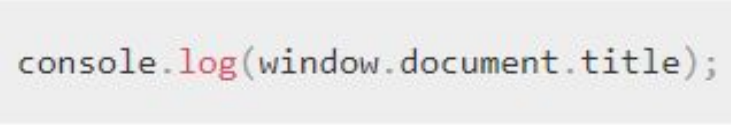


```
console.log("An error occurred while loading the content");
```

document

window.document returns a reference to the document contained in the window.

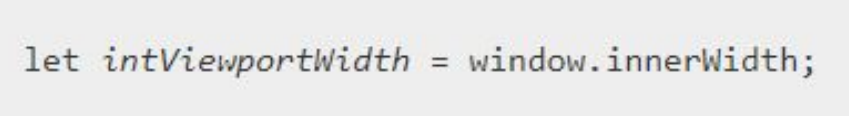




```
console.log(window.document.title);
```

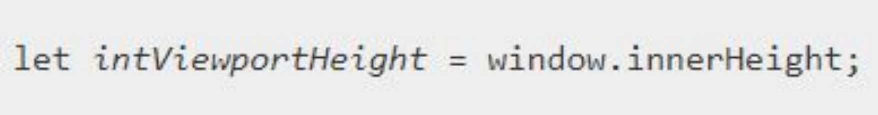
Inner Height and Width

The read-only Window property **innerWidth** returns the interior width of the window in pixels.



```
let intViewportWidth = window.innerWidth;
```

The read-only **innerHeight** property of the Window interface returns the interior height of the window in pixels, including the height of the horizontal scroll bar, if present.



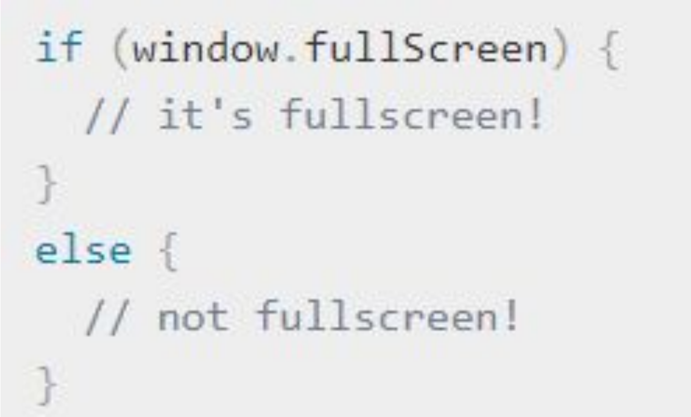
```
let intViewportHeight = window.innerHeight;
```

window.isFullScreen

The **fullScreen** property of the Window interface indicates whether the window is displayed in full screen mode or not.

A boolean. Possible Values:

- true: The window is in full screen mode.
- false: The window is not in full screen mode.

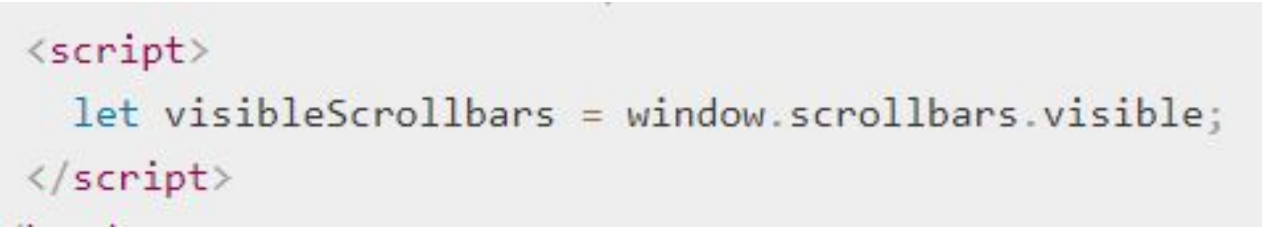


```
if (window.fullScreen) {
  // it's fullscreen!
}
else {
  // not fullscreen!
}
```

window.scrollbars

The **Window.scrollbars** property returns the scrollbars object, whose visibility can be checked.

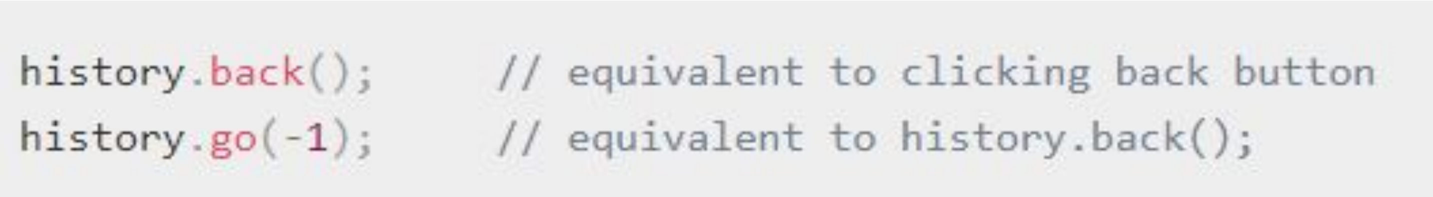
The following complete HTML example shows how the visible property of the scrollbars object is used.



```
<script>
  let visibleScrollbars = window.scrollbars.visible;
</script>
```

window.history

The **Window.history** read-only property returns a reference to the History object, which provides an interface for manipulating the browser *session history* (pages visited in the tab or frame that the current page is loaded in).



```
history.back();     // equivalent to clicking back button
history.go(-1);     // equivalent to history.back();
```

Window Object Methods

Window Location Methods

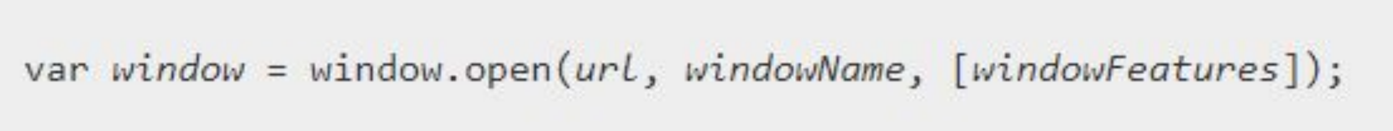
The window.location object can be used to get information on the current page address (URL) and to redirect the browser to a new page.

- window.location.href returns the href (URL) of the current page
- window.location.hostname returns the domain name of the web host
- window.location.host returns both the host name and any associated port
- window.location.pathname returns the path and filename of the current page
- window.location.protocol returns the web protocol used (http: or https:)
- window.location.assign() loads a new document

Window Open Method

The Window interface's **open()** method loads the specified resource into the new or existing browsing context (window, <iframe> or tab) with the specified name.

If the name doesn't exist, then a new browsing context is opened in a new tab or a new window, and the specified resource is loaded into it.



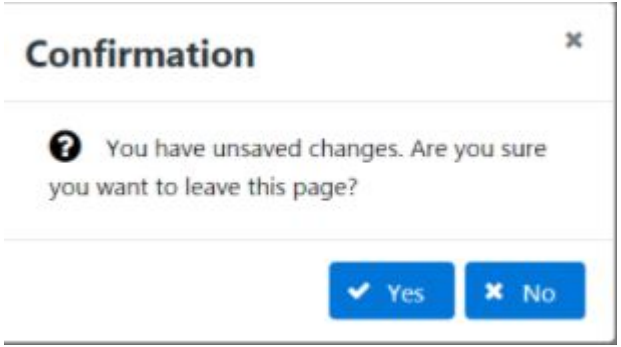
```
var window = window.open(url, windowName, [windowFeatures]);
```

Popup Boxes

Popup boxes (or dialog boxes) are modal windows used to notify or warn the user, or to get input from the user.

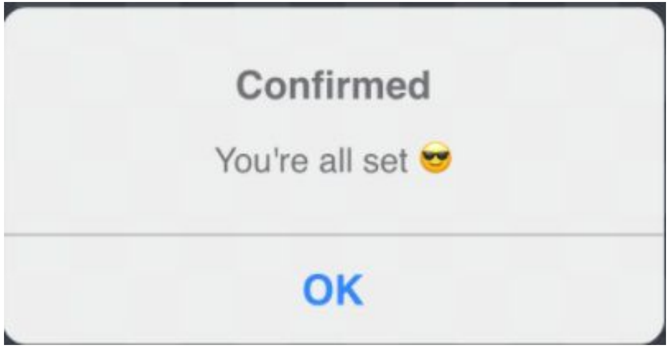
Popup boxes prevent the user from accessing other aspects of a program until the popup is closed, so they should not be overused.

There are three different kinds of popup methods used in JavaScript: window.alert(), window.confirm() and window.prompt().



Confirmation

You have unsaved changes. Are you sure you want to leave this page?

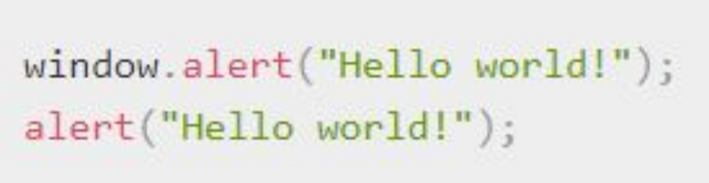


Confirmed

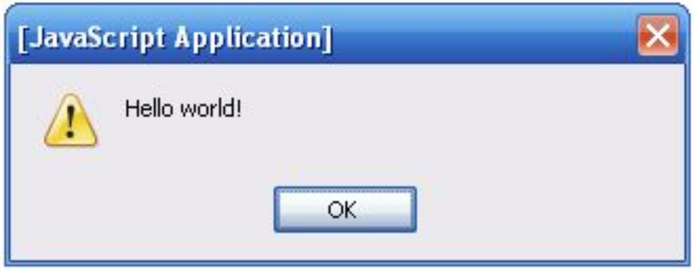
You're all set 😎

Window Alert Method

The alert method displays messages that don’t require the user to enter a response. Once this function is called, an alert dialog box will appear with the specified (optional) message. Users will be required to confirm the message before the alert goes away.



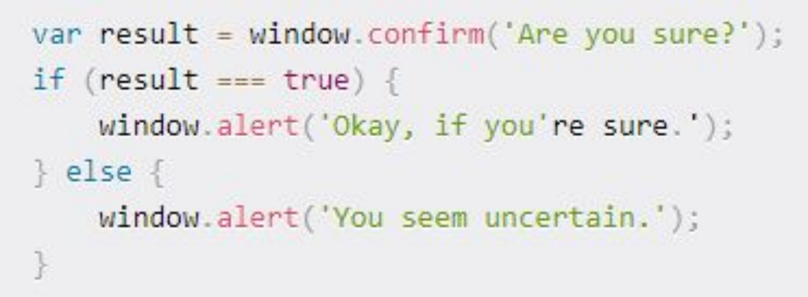
```
window.alert("Hello world!");
alert("Hello world!");
```



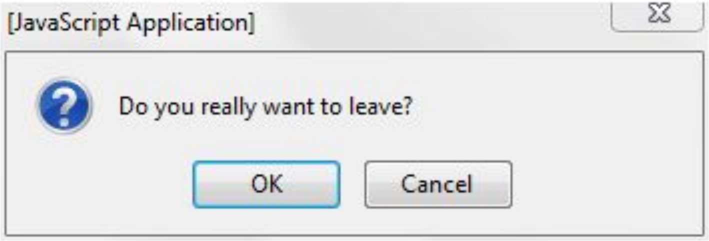
Hello world!

Window Confirm Method

The confirm method is similar to window.alert(), but also displays a cancel button in the popup. The buttons return boolean values: true for OK and false for Cancel.



```
var result = window.confirm('Are you sure?');
if (result === true) {
    window.alert('Okay, if you're sure.');
} else {
    window.alert('You seem uncertain.');
}
```

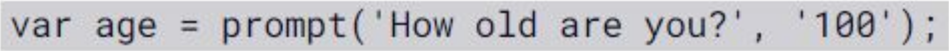


[JavaScript Application]

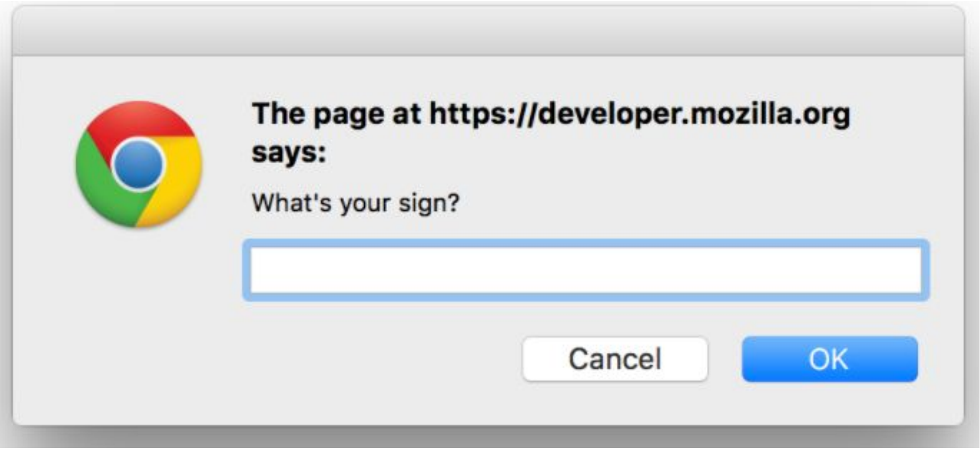
Do you really want to leave?

Window Prompt Method

The prompt method is typically used to get text input from the user. This function can take two arguments, both of which are optional: a message to display to the user and a default value to display in the text field.



```
var age = prompt('How old are you?', '100');
```



The page at https://developer.mozilla.org says:

What's your sign?