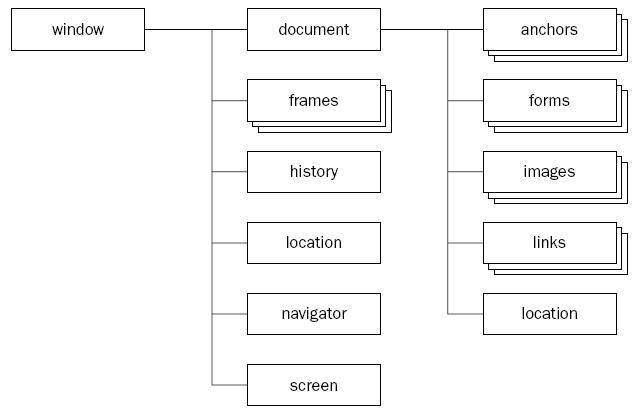
**Pràctica 17**

**L’objecte window**

L’objecte window l’hem estat utilitzant constantment. Representa la finestra del navegador. A continuació podem veure objectes que col·laboren directament amb l'objecte windows:



Window és un objecte global i fins ara hem vist els següents mètodes:

* alert(): Mostra un diàleg d'alerta amb un missatge.
* prompt(): Mostra un diàleg per a l'entrada d'un valor de tipus string.

Aquestes funcions les podem utilitzar anteposant el nom de l'objecte window al nom del mètode:

var valor = window.prompt("Introduïu un valor");

O en forma resumida indicant només el nom del mètode (com ho hem estat fent):

var valor = prompt("Introduïu un valor");

Això és possible ja que l'objecte window és l’objecte de màxim nivell. Per reduir la quantitat de caràcters teclejats normalment trobarem els programes codificats de la segona forma.

Hi ha d’altres mètodes de l’objecte window que encara no hem vist i que detallem a continuació:

* confirm(): Mostra un diàleg de confirmació amb els botons Acceptar i Cancel·lar. El diàleg mostra el text indicat dins dels parèntesis del confirm(). Si l’usuari pren ‘Acceptar’, la funció retorna ‘true’, en canvi, si pren ‘Cancel·lar’, retorna ‘false’.

Exemple:

var res = confirm("Ets major d'edat?");

console.log(res); // retorna ‘true’ si pren Acceptar i ‘false’ si pren Cancel·lar

* open() i close(): Obre o tanca una nova finestra del navegador. Podem especificar la mida de la finestra, el seu contingut, etc. La sintaxi és la següent:

       [var variable =] [window.]open(URL, nom de la finestra, propietats)

Si volem tenir accés a la nova finestra des de la finestra on la vam crear, haurem assignar la nova finestra a una variable, sinó simplement invoquem el mètode: el navegador automàticament sabrà que pertany a l'objecte window.

El paràmetre URL és una cadena que contindrà la direcció de la finestra que estem obrint: si està en blanc, la finestra s'obrirà amb una pàgina en blanc.

Les propietats són una llista, separada per comes, d'alguns dels següents elements:

* width = píxels (Mínim valor és 100px)
* height = píxels (Mínim valor és 100px)
* toolbar [= yes | no] (Només vàlid per a Internet Explorer i Firefox)
* location [= yes | no] (Només vàlid per a Opera)
* status [= yes | no]
* menubar [= yes | no]
* scrollbars [= yes | no] (Només vàlid per a Internet Explorer, Firefox i Opera)
* resizable [= yes | no] (Només vàlid per a Internet Explorer)

A continuació mostrem un exemple per obrir una finestra:

var finestra =open("https://www.w3schools.com/jsref/met\_win\_open.asp",

"","width=500px,height=500px,resizable=no,toolbar=yes");

**PROBLEMA**

1. Desenvolupar una pàgina que permeti obrir una altra finestra quan es pressiona un botó. Aquesta finestra ha de tenir una mida de 500x500px, i ha de mostrar el menú i la barra d'eines. A més, hi haurà un botó que en polsar-ho es tancarà la finestra oberta.

**Propietat location de l'objecte window**

Quan li assignem una nova direcció a la propietat location de l'objecte window, el navegador redirecciona a aquesta pàgina.

Implementarem un petit exemple per veure la utilitat d'aquesta propietat: Suposem que tenim un hipervincle que en ser pressionat mostra una finestra de confirmació on es pregunta si l’usuari és major d’edat. En cas que l’usuari premi el botó ‘Acceptar’, redireccionarem a una altra pàgina, en cas contrari mostrarem un ‘alert’:

<!DOCTYPE html>  
<html>

<head>

<title>Pràctica 17</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script>

function verificaMajorEdat(){

var res = confirm("Ets major d'edat?");

if(res){

window.location = 'https://www.w3schools.com/jsref/obj\_location.asp';

}else{

window.alert("Podrà accedir-hi quan sigui major d'edat.");

}

}

</script>

</head>

<body>

<h1>Propietat location de l'objecte window</h1>

<a href="javascript:verificaMajorEdat();">link per majors d'edat</a>

</body>

</html>

El primer que hem d'indicar és que per cridar a una funció de javascript des d'un hipervincle hem de posar la paraula javascript seguida de dos punts i finalment, el nom de la funció:

<a href="javascript:verificaMajorEdat();">

La funció ‘verificMajorEdat()’ mostra la finestra amb els botons ‘Accepta’r i ‘Cancel·lar’. Si es pressiona el botó ‘Acceptar’, la funció ‘confirm()’ retorna ‘true’ i per tant s'executa el codi que conté ‘window.location’. En cas contrari es mostra el ‘alert’.

**PROBLEMA**

1. Desenvolupar una pàgina que tingui un enllaç. Quan es pressioni aquest enllaç, s’haurà de generar un valor aleatori entre 0 i 2. Si es genera el 0 s’haurà de redireccionar a la pàgina de Outlook, si es genera 1, es redireccionarà a Gmail, en cas que sigui 3, a yahoo.  
   Per generar un valor aleatori utilitzar la funció random de l'objecte Math.

**Propietat history l'objecte window**

L'objecte history emmagatzema totes les pàgines que visitem. Després, amb una sèrie de funcions, podem extreure de la memòria de l'ordinador les pàgines ja visitades, sense haver de demanar-les novament al servidor.

Compta amb les següents funcions:

* window.history.back(); // Retrocedeix a la pàgina anterior
* window.history.forward(); // Avança a la pàgina següent
* window.history.go(); // Avança o retrocedeix a la llista de pàgines visitades.

Trucar a les funcions ‘back()’ o ‘forward()’, té el mateix comportament que prémer el botó "Enrere" o "Següent" del navegador.

La funció ‘go()’ rep com a paràmetre ‘-1’ per retrocedir una pàgina a l’historial o ‘+1’ per avançar una pàgina en la memòria de l’historial.

El següent exemple ens permet carregar una segona pàgina i després retrocedir a la primera pàgina sense haver de sol·licitar-la novament al servidor:

<!DOCTYPE html>

<html>

<head>

<title>Pràctica 17</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script>

function avançar(){

window.history.go(+1);

}

</script>

</head>

<body>

<h1>Propietat location de l'objecte window: Pàgina 1</h1>

<a href="pagina2.html">Anar a la segona pàgina</a>

<a href="javascript:avançar();">avançar</a>

</body>

</html>

En aquesta primera pàgina, la primera vegada, hem de carregar la segona pàgina seleccionant l'hipervincle pàgina2.

La segona pàgina:

<!DOCTYPE html>

<html>

<head>

<title>Pràctica 17</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script>

function retrocedir(){

window.history.go(-1);

}

</script>

</head>

<body>

<h1>Propietat location de l'objecte window: Pàgina 2</h1>

<a href="javascript:retrocedir();">retrocedir</a>

</body>

</html>

A la segona pàgina, mitjançant el mètode go i passant-li un valor negatiu, retrocedim a la primera pàgina sense tenir la necessitat de recarregar-la.

Podem millorar l'exemple accedint a l'atribut length (emmagatzema la quantitat de pàgines de la llista) de l'objecte history:

if(window.history.length > 1){

window.history.go(+1);

}else{

alert('no hi ha cap pàgina a la memòria cap endavant');

}

**PROBLEMA**

1. Confeccionar tres pàgines. Disposar de 2 hipervincles a cada pàgina per anar directament a les altres dos pàgines. A cada pàgina també hi haurà 2 hipervincles per anar endavant o enrere.

**Propietat screen de l'objecte window**

L'objecte screen ofereix informació sobre el monitor on s'està executant el navegador.  
La propietats principals de l'objecte screen són:

* availHeight: L'alt de la pantalla en píxels disponible per al navegador.
* availWidth: L'amplada de la pantalla en píxels disponible per al navegador.
* colorDepth: Representa el nombre de bits usats per representar els colors.
* height: L'alt de la pantalla en píxels.
* width: L'amplada de la pantalla en píxels.

El següent programa mostra el valor emmagatzemat en les cinc propietats que té l'objecte screen:

<!DOCTYPE html>

<html>

<head>

<title>Practica 17</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script>

function inici(){

console.log('availHeight:' + screen.availHeight);

console.log('availWidth:' + screen.availWidth);

console.log('height:' + screen.height);

console.log('width:' + screen.width);

console.log('colorDepth:' + screen.colorDepth);

}

window.onload=inici;

</script>

</head>

<body>

<h1>Propietat screen de l'objecte window</h1>

</body>

</html>

El resultat obtingut és el següent:

availHeight:974

availWidth:1280

height:1024

width:1280

colorDepth:24

**PROBLEMES**

1. Volem confeccionar una pàgina que obrirà una finestra nova i la volem redimensionar de tal forma que quedi centrada deixant 50px amunt, avall, a l’esquerra i a la dreta. Per aconseguir-ho farem una pàgina amb quatre botons. El primer botó haurà d’obrir una nova finestra de 500x500 px, el segon botó haurà de tancar la nova finestra oberta, el tercer boto haurà de redimensionar la nova finestra y el quart botó haurà de moure la nova finestra. Per redimensionar una finestra podem utilitzar el mètode ‘resizeTo()’ de l’objecte Windows:

Window.resizeTo(screen.width- ?,screen.height- ?);

Hauràs de calcular el valor a restar per tal de que quedi amb la mida correcta. Per moure la finestra podem utilitzar el mètode ‘moveTo()’ de l’objecte Windows (Investiga per internet què paràmetres espera aquest mètode).

**Propietat navigator l'objecte window**

Conté informació sobre el navegador web. La implementació d'aquest objecte varia entre navegadors (IExplorer, FireFox, Opera, etc.).

Les propietats comunes a aquests navegadors són:

* appCodeName: emmagatzema el codi del nom del navegador.
* appname: emmagatzema el nom oficial del navegador.
* appVersion: emmagatzema la versió del navegador.
* cookieEnabled: emmagatzema si les ‘cookies’ estan actives en el navegador.
* platform: emmagatzema la plataforma on el navegador s'està executant.
* userAgent: emmagatzema una cadena de text representant l’agent que està utilitzant el navegador. S’utilitza per identificar el navegador utilitzat.

Aquest petit programa mostra els valors de les propietats abans anunciades:

<!DOCTYPE html>

<html>

<head>

<title>Practica 17</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script>

function inici(){

console.log('appCodeName:' + navigator.appCodeName);

console.log('appName:' + navigator.appName);

console.log('appVersion:' + navigator.appVersion);

console.log('cookieEnabled:' + navigator.cookieEnabled);

console.log('platform:' + navigator.platform);

console.log('userAgent:' + navigator.userAgent);

}

window.onload=inici;

</script>

</head>

<body>

<h1>Propietat navigator de l'objecte window</h1>

</body>

</html>

El resultat obtingut en un navegador Firefox és el següent:

appCodeName:Mozilla

appName:Netscape

appVersion:5.0 (Windows)

cookieEnabled:true

platform:Win64

userAgent:Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:61.0) Gecko/20100101 Firefox/61.0

**PROBLEMA**

1. Pintar el fons de la pàgina de verd si el navegador és Firefox o vermell si és qualsevol altre. Investiga per internet com utilitzar ‘userAgent’ per identificar el navegador.