



Linkia FP

DAW	M06	T02	Material	Objetos Predefinidos
-----	-----	-----	----------	----------------------

Concepto de Objeto

- Un objeto es la definición de una estructura de datos formada por un conjunto de atributos y funciones.
- Podemos crear distintos objetos definiendo distintas estructuras.
- Al definir una nueva variable podemos indicar que utiliza una estructura en concreto.

Iniciar Objetos en JS

- Una vez definido un Objeto, podemos crear una variable para que utilice su estructura :

```
let fecha = new Date();  
let nombre_gato = new String();
```

- Una vez definida una variable con una estructura concreta, podemos asignar un valor a las variables de la estructura y llamar a sus funciones:

```
fecha.getDate();  
nombre_gato.concat("hi!");
```

Objetos JavaScript

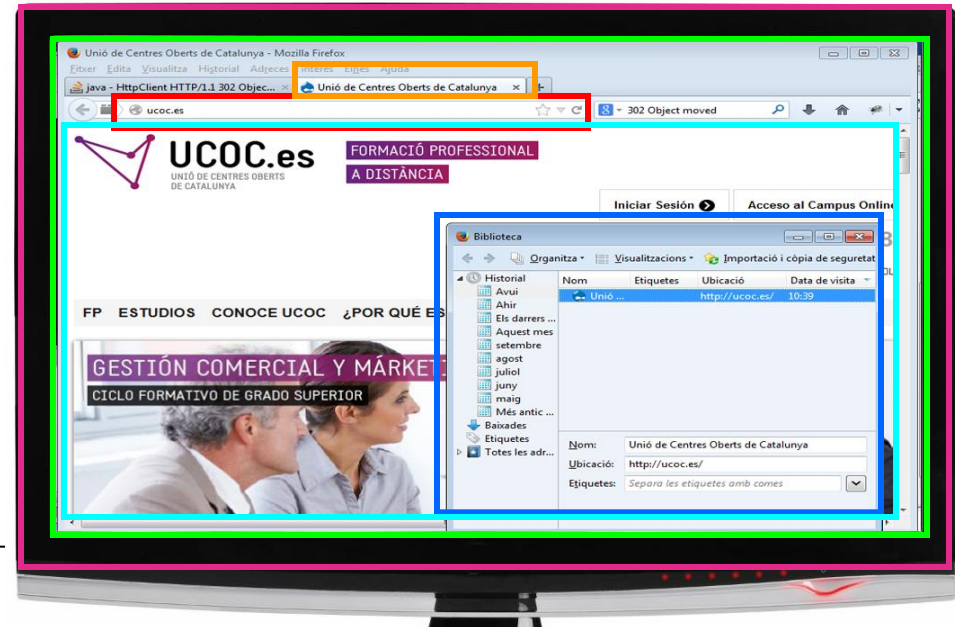
- Objetos con atributos y funciones lógicas, numéricas y de tipo de datos.
- Objetos ligados al Modelo y al Controlador.

- Global
- Boolean
- Math
- String
- Array
- Date
- Number
- RegExp

Objetos del Navegador

- Objetos con atributos y funciones sobre el navegador, la visualización i la interpretación del código HTML.
- Objetos ligados a la Vista.

- Window
- Document
- Navigator
- Screen
- History
- Location



- El Objeto más general en JS.
- Contiene funciones para codificar/decodificar texto, convertir variables , evaluar i ejecutar código.
- Podemos llamar a sus funciones y atributos directamente.

PROPIEDADES

- Infinity : valor infinito
- NaN : valor no numérico
- undefined : valor que tiene una variable antes de asignarle un valor

FUNCIONES

- isNaN() : indica si una variable es un numero.
- parseInt() : transforma una variable a Number.
- eval() : ejecuta un texto como si fuera un script
- isFinite(): indica si el argumento es un numero finito

- Permite almacenar y manipular cadenas de texto.
- Atributos :length.

METODOS

- **var str = "hola";**
- **str.indexOf("a");** posición de **a** dentro de **str**
- **str.substring(1,2);** extrae de str el texto entre la posición 1 y 2
- **str.substr(1,2);** extrae del str 2 caracteres desde la posición 1.
- **str.charAt(3)** , **str[3]** : obtiene el 3er caracter
- **str.toUpperCase();** retorna una String con el texto de **str** en mayúsculas.
- **str.length;** propiedad con la longitud del string.
- **str.split(",");** Corta la String por el carácter pasado por parámetro y con cada trozo genera un array de strings
-

NUMBER

- Información sobre números en JS. Almacena y manipula números.
 - **Propiedades:** calcular el máximo /mínimo .
 - `Number.MAX_VALUE`;
 - **Funciones:** convertir a exponencial, definir el número de dígitos, convertir a string.
 - `Number.isInteger(34.4)`;
 - `num.toExponential()`;

MATH

- Operaciones y variables matemáticas.

PROPIEDADES

- `PI` : número pi
- `LN2` : logaritmo de 2
- ...

FUNCIONES

- `Math.floor()`; redondea un numero a entero.
- `Math.random()`; número aleatorio entre 0 y 1
- `Math.abs()`; valor absoluto de un número
- `cos(x)`; calcula el coseno de x
- ...

- Permite almacenar y manipular fechas

FUNCIONES

- `var d= new Date();` genera un objeto Date con la fecha actual.
- `var d= new Date(año, mes, día, hora, minutos);` genera un objeto Date con la fecha indicada.
- `d.getDate();` retorna el día almacenado en d.
- `d.getHours();` retorna la hora almacenada en d.
- `d.getMonth();` retorna el mes almacenado en d.
- `d.getFullYear();` retorna el año almacenado en d.
- `d.setDate();` establece el día almacenado en d.
- `d.setHours();` establece la hora almacenado en d.
- ...

Ejemplo: programa muestre la fecha y genere un número aleatorio entre el 0 y el día de la semana en el que estamos.

NAVIGATOR

- Información sobre el navegador que contiene el documento HTML en el que se ejecuta JS.

- **navigator.userAgent**: nombre del navegador
- **navigator.platform** : sistema operativo
- **navigator.cookieEnabled**; cookies habilitadas
- **navigator.language**; lenguaje por defecto
- ..

LOCATION

- Contiene información sobre la URL de la web en la que se ejecuta el JavaScript.

- **location.href**; la URL completa. Cambiar ésta propiedad provoca navegar a la nueva URL
- **location.pathname**; Retorna la / inicial seguido por la ruta de la URL
- **location.port**; Retorna el número de puerto en la URL

- **location.replace("https://linkiafp.es");** cambia la página actual por la pasada por parámetro. No guarda el cambio en el historial

DOCUMENT

• Contiene la estructura de datos que representa la página web. Nos permite acceder y modificar a los elementos que forman la web.

PROPIEDADES

- `document.title;`
- `document.URL;`
- `document.cookie;`
- `document.forms;`

FUNCIONES

- `document.getElementById();`
- `document.getElementsByTagName();`
- ..

- Variables vinculadas a un sitio web almacenadas en el navegador y el servidor.
- Se envían del navegador al servidor y viceversa.
- Se puede acceder siempre desde el servidor, y en ocasiones desde el navegador con JS.

Crear / Modificar /Borrar / Leer una Cookie de forma nativa

```
document.cookie='nom=James';
```

```
document.cookie='apellido=Bond';
```

```
document.cookie='nom=Jaime';
```

```
document.cookie='nom=James;expires=Thu, 2 Aug 2013 20:47:11 UTC;' ;
```

```
textoGuardado=hola; nom=Jaime; apellido=Bond
```

- Información sobre [la ventana](#) que contiene el documento HTML en el que se ejecuta el JS.

ATRIBUTOS

- | | |
|---|---|
| <ul style="list-style-type: none">a) <code>window.location</code>b) <code>window.screenLeft;</code>c) <code>window.screenTop;</code>d) <code>window.document;</code> | <ul style="list-style-type: none">a) Información sobre la dirección del documento HTML mostrado en la Window.b) Distancia desde la izquierda de la pantalla hasta la ventana que contiene la web.
<ul style="list-style-type: none">3) Cada 3 segundos muestra un alerta con el texto "Hello".4) En 5 segundos muestra una alerta "Wake Up!" |
|---|---|

FUNCIONES

- 1) `window.alert("Hola!");`
- 2) `window.prompt("que tal?");`
- 3) `setInterval(function(){ alert("Hello"); }, 3000);`
- 4) `setTimeout(function(){ alert("Wake Up!"); }, 5000);`



ATRIBUTOS Y FUNCIONES PARA CREAR Y CERRAR VENTANAS

- a) `window.open ("url.html","Nombre","scrollbars=no,width=200");`
- b) `window.focus();`
- c) `window.opener;`
- d) `window.close();`
- e) `window.closed;`

EJEMPLO

`var miPopup = window.open("url.html","Nombre","scrollbars=no");`

`miPopup.focus();` → Referencia a la nueva ventana

`window.focus();` → Muestra delante de todo la nueva ventanada creada (url.html)

`miPopup.close();` → Muestra delante de todo la ventana inicial (en la que se ha ejecutado el JS). *DESHABILITADO POR LOS NAVEGADORES*

→ Cierra la nueva ventana creada(url.html)

- `window.alert("PopUp ");`
- `var fname = window.prompt("Pregunta","valor por defecto");`
- `var miPopup = window.open("url.html", "Nombre", "scrollbars=no");`
`miPopup.focus();`
`miPopup.resizeTo(250, 250);`
`miPopup.close();`

Referencia al window abierto

`{`
`_blank`
`_self`
`..`
`}`

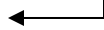
index.html

`miPopup = window.open("url.html", "miWindow", "scrollbars=no");`

`{`
`window.focus();`
`window.close();`
`}`

url.html

`origen = window.opener`



- **var** posX = window.screenX;
- **var** ventanaPadre = window.opener; Referencia a la ventana padre
- **var** estaCerrada = index.closed;
- **var** doc = window.document;

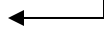
index.html

```
miPopup = window.open("url.html", "miWindow", "scrollbars=no");
```

```
{
  window.focus();
  window.close();
}
```

url.html

```
origen = window.opener
```



Conceptos clave para la actividad

- Crear código HTML “complejo” desde JS
 - Acceder a propiedades y métodos de objetos nativos JS
 - Buscar información sobre métodos y objetos nativos
 - Generar un número aleatorio
 - Acceder y modificar el valor de una cookie
 - Obtener y modificar el valor de un input
 - Ejecutar un código pasado X tiempo
 - Ejecutar un código cada X tiempo
 - Abrir una nueva ventana.
 - Cerrar una ventana abierta.
 - Acceder desde una ventana padre a el contenido de una hija y viceversa
-



Linkia FP

Formación Profesional
Oficial a Distancia