JavaScript en el Navegador: jQuery

Los programas javascript pensados para ejecutarse en el navegador web pueden estar:

En un fichero HTML: <head><script></script></head>
Fichero .js: <script src=""></script>
-Se pueden incluir varios scripts en un mismo documento
-La variables globales definidas en un script son accesibles desde los scripts incluidos posteriormente.
Para ver los resultados obtenidos Ctrl+Mayús+K

#NOTA: Las funciones específicas que se implementan para NodeJS y ExpressJS no son aplicables en el navegador [modulos NodeJS , require, npm fs, path y paquetes externos de NodeJS]

En cambio tenemos otras funciones como:

- Manipulación de elementos de la página web actualmente cargada.
- Funcionalidad relativa al historial de navegación.
- **Peticiones al servidor AJAX. (Tema 8)**

Objetos Predefinidos

- ***Variables y funciones Globales*** *window*
  window contiene todas las funciones y variables globales definidas, más algunos métodos predefinidos.

  #NOTA: toda variable y función global es implícitamente un atributo de *window*

1. ***Muestra una ventana emergente del navegador con un mensaje dado***: **alert("<mensaje>");**

   #Es una función **bloqueante:** el script no continua hasta que se pulse el boton Aceptar
2. ***Temporizadores***
   a. ***Ejecuta la función transcurrida una cantidad t de milisegundos:*** **setTimeout(fun,t);**
   b. ***Ejecuta la funcion cada t milisegundos:*** ***setInterval(fun,t);***
   c. Ambas funciones devuelven un número identificador que se pueden pasar para ***Cancelar los temporizadores Correspondientes:***
3. ***Abren una ventana emergente con la url dada:*** ***open("<url>");***

- ***Manejo de Historial*** *history*

  history proporciona una funcionalidad similar a la de los botones *Atrás y Adelante*, que permiten navegar por las últimas páginas visitadas. [**history::.go(<número>),. back(), .foward()**]

- ***Información del navegador*** *navigator*

  navigator proporciona información sobre el navegador **[navigator:: .appName, .languaje, .platform, .product, .userAgent**]

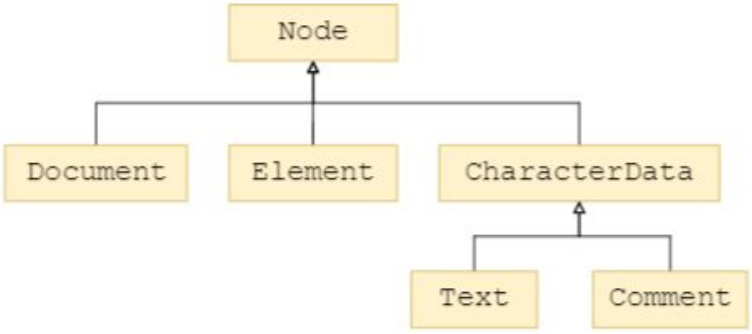
- ***Información sobre la pantalla*** *screen*

  screen contiene atributos sobre la resolución de pantalla.

  1. ***Tamaño total de pantalla*** [**screen:: .width, .height**]
  2. ***Tamaño disponible*** [**screen:: .availWidth, .availHeight**]

DOM

Un documento HTML consiste esencialmente en una serie de elementos anidados, que puede visualizarse como una estructura en forma de árbol . ***DOM*** es una interfaz que permite acceder, navegar y modificar este árbol.



document es una variable que contiene la raiz del arbol

1. ***Devuelve el objeto Node correspondiente al elemento HTML :*** **getElementById(id)**
2. ***Devuelve un NodeList con todos los "name" del documento:*** **getElementsByTagName(name)**
3. ***Devuelve un NodeList que contenga la clase name:*** **getElementsByClassName(name)**
4. ***Atributos para navegar por árbol DOM:***
   a. **childNodes(tipo NodeList)**
   b. **firstChild(tipo Node)**
   c. **lastChild(Tipo Node)**

- **Inconvenientes de utilizar DOM**
  - **Inconsistencias entre distintos navegadores**

JQuery

Es una librería javascript funciona en el lado del navegador web , que proporciona su propia API abstrayendo las diferencias entre los navegadores; ya que proporciona una capa de abstracción entre las funciones del DOM y el programador .

- **Ventajas**
  - Eliminación de inconsistencias entre navegadores.
  - Código más claro y conciso.
  - Extendible mediante plugins.

- **Instalación**
  - Descargar Libreria**: <script src="jquery-3.2.1.min.js"></script>**
  - Incluir Libreria en un CDN: **<script src="**https://code.jquery.com/jquery-3.2.1.min.js**">**

- **La Variable** $
  Es la variable global nueva que contiene los métodos de jQuery.
  1. Usos:
     a. Como Función: **$(document), $("header")**
     b. Como Objeto: **$.ajax(), $.get()**

  - **Funciones Disponibles**
    - ***Inicialización de la página*:**
      Podemos asociar un callback a la inicialización del DOM y esta función callback será llamada cuando el DOM se haya cargado en memoria , para ello usamos **$(()=>{});**

      Múltiples Facetas de S
      1. **Acciones a realizar tras iniciar el DOM $(callback)**
      2. **Seleccionar un único elemento $(elemento)**
      3. **Seleccionar uno o varios elementos $(selector CSS)**
      4. **Crear nuevos elementos y seleccionarlos $(código HTML)**

    - **Búsqueda y navegación en el DOM**
      Para seleccionar los elementos del árbol a modificar: una ***selección*** es un conjunto de nodos del DOM; es decir son instancias de una determinada clase.

      La selección de elementos se puede realizar mediante selectores css: **$(<selector>)** [devuelve los elementos afectados por el selector]

Selectores Disponibles

1. **Todos los nodos del DOM** $("*")
2. **Todos los elementos de un tipo:** $(<tipo>)
3. **Todos los elementos de tipo x que tienen la clase y** $(x.y)
4. **Todos elementos con la clase** $(.clase)
5. **Elemento con id** $(#id)

Algunos métodos de los objetos seleccion permiten obtener otras selecciones.

1. **Hijos Directos** $(<tipo>).children
2. **Hijos de los elementos x que contenga la clase y** $(x).children(y)
3. **Descendientes (directos o indirectos) de los elementos x que contienen la clase y** $(x).find(y)
4. Padres de elemento seleccionado .parent()

Si al seleccionar devuelve más de un elemento se pueden utilizar la selección .

1. **Número de elementos seleccionados:** .length
2. **Elemento n-ésimo :** .eq(n)

- **Manipulación del DOM**

Las selecciones tienen métodos para manipular las propiedades y el estilo de todos los objetos seleccionados.

1. **Propiedad de los elementos seleccionados:**
   a. **Modificar: .prop(propiedad, valor)**
   b. **Obtiene el valor actual: .prop(propiedad)**
2. **Modificar Clases y Estilos:**
   a. **Agregar clase: .addClass(name)**
   b. **Remover clase : .removeClass(name)**
   c. **Activar Clase: .toggleClass(name)**
   d. **Tiene Clase: .hasClass(name)**
3. **Propiedades CSS:**
   a. **Modificar: .css(propiedad, valor)**
   b. **Obtiene valor actual:: .css(propiedad)**
4. **Mostrar/Ocultar Elementos: .show() / .hide()**
5. **Obtener Contenido de texto:**
   a. **Obtener : .text()**
   b. **Modificar: .text(nuevo Texto)**
6. **Si queremos que el contenido interprete etiquetas HTML: .html()**
7. **Valor de una propiedad personalizada de un elemento:**
   a. **Obtener valor: .data(atributo)**

b. **Cambiar valor : .data(atributo, valor)**

- **Creación de nuevos elementos en el DOM**
  1. **Añadir Nodos al DOM:**
     a. **Último hijo (derecha): .append(elem)**
     b. **Primer hijo (izquierda): .prepend(elem)**
     c. **Hermano Mayor: .before(elem)**
     d. **Hemano Menor: .after(elem)**
  2. **Eliminar Objeto seleccionado: .remove()**
  3. **Hacer una copia del objeto seleccionado: .clone()**
  4. **Envuelve los objetos seleccionados en el objeto pasado como parámetro : .wrap(elem)**

- **Posicionamiento y dimensiones**
  - **Dimensiones** (Devuelve los valores del objeto seleccionado)**:**
    - **Altura:**
      - **Consultar: .height()**
      - **Modificar: .height(ancho)**
    - **Anchura:**
      - **Consulta: .width()**
      - **Modificar: .width(ancho)**
    - **Obtienen/ Modifican las dimensiones sin tener encuenta el borde ni los márgenes (si padding) : .innerWidth()/ .innerHeight()**
    - **Obtienen/ Modifican las dimensiones tomando en cuenta padding,bordes y margenes: .outerWidth() / .outerHeight()**
  - **Posicionamiento:**
    - **Devuelve la posición con respecto del documento: .offset()** esta función permite cambiar la posición de un objeto
    - **Devuelve la posición con respecto al elemento padre: .position()**
      - **Coordenada x: left**
      - **Coordenada y: top**

- **Manejo de Eventos**

Asociar un evento a una selección: .on (tipoEvento, ()=>{})

- **Eventos de Ratón**
  1. **Pulsar Botón Izquierdo: click / dblclick**
  2. **Pulsar Botón Derecho: contextmenu**
  3. **Pulsar +Liberar un Botón: mousedown, mouseup**
  4. **Entrada/Salida del puntero en un area: mouseenter, mouseleave,mousemove**

- **Eventos de Teclado**
    1. **Pulsación o Liberación de una tecla: keydown , keyup**
    2. **Pulsación + Liberación : keypress**
- **Objeto** *event*

A veces es necesario conocer más información sobre el *event*, para ello el método on recibe un objeto de la clase Event que tiene información del evento como:

1. **Elemento que produjo el evento: target**
2. **Hora del Evento: timeStamp**
3. **Boton/Tecla pulsado: which**
4. **Indicación de Tecla Auxiliar: ctrlKey, metaKey, shiftKey, altKey**
5. **Coordenadas del ratón respecto al HTML: pageX, pageY**

Los eventos pueden trabajar con un html que se carguen de forma estática y/ o dinámica

- **Eventos Directos :** afectas a los elementos que ajusten con el selector CSS en el momento de ejecutar la llamada
  $(selectorCSS).on (tipoEvento, (event)=>{})

- **Eventos Delegados :** afectan a los elementos que ajusten con el selectorCSS2 , afecta tanto los elementos presentes como a los futuros
  $(selectorCSS).on (tipoEvento,selectorCSS2 (event)=>{})

- **Propagación de Eventos**

Sabemos que los elementos de HTML pueden estar anidados unos dentros de otros, y es por eso que los eventos se propagan de manera ascendentes en el DOM, desde los más internos hasta los externos .

1. **Impedir la propagación del evento hacia los elementos superiores del árbol: event.stopPropagation()**

- **Acciones por Defecto**

Algunos tipos de eventos llevan implícita una determinada acción por parte del navegador, es posible evitar la ejecución de estas acciones utilizando preventDefault()

Casos de Implementación

1. *Evitar el envio del formulario en caso de no ser validado*

NOTA: La validación del lado del cliente no exime de la validación en el lado del servidor

- **Animaciones y Efectos**

Es posible utilizar efectos visuales para mostrar y ocultar elementos

1. **Desaparicion/ Aparicion Gradual: fadeOut([duracion]), fadeIn([duracion]),fadeTo(duracion,opacidad)**
2. **Desaparición/Aparición con desplazamiento: slideDown([duration]), slideUp([duracion])**
3. **Animaciones Personalizadas: animate(propiedades, t)** modificar propiedades CSS de manera gradual para que alcance los valores indicados en propiedades en t milisegundos