

12-11-2023

Modelo de objetos predefinidos en JavaScript

Tarea 03

ROBERTO RODRÍGUEZ JIMÉNEZ
roberto.rodjim.1@educa.jcyl.es

Contenido

Tarea online DWEC02.....	2
¿Qué contenidos o resultados de aprendizaje trabajaremos?.....	2
Resultados de Aprendizaje.....	2
Contenidos.....	2
1.- Descripción de la tarea	3
Caso práctico	3
¿Qué te pedimos que hagas?	3
2.- Información de interés.....	4
Recursos necesarios	4
Consejos y recomendaciones	4
3.- Evaluación de la tarea	4
Criterios de evaluación implicados.....	4
¿Cómo valoramos y puntuamos tu tarea?	4
RESOLUCIÓN.....	5
Abra una nueva ventana no redimensionable.....	5
Hacer una función y dentro de esa función:.....	7
Y ahora fuera del código de la función que siga haciendo lo siguiente:	10
Código.....	14
index.html	14
aplicacion.js	14
Recursos.....	18

Tarea online DWEC02

Título de la tarea: Estructura del lenguaje JavaScript

Unidad: DWEC02

Ciclo formativo y módulo: Desarrollo de Aplicaciones Web - Desarrollo web en entorno cliente

Curso académico: 2020/2021

¿Qué contenidos o resultados de aprendizaje trabajaremos?

Resultados de Aprendizaje

- ✓ RA 3: Escribe código, identificando y aplicando las funcionalidades aportadas por los objetos predefinidos del lenguaje.

Contenidos

Modelo de objetos predefinidos en JavaScript.

1. Objetos de más alto nivel en Javascript.
 1. Objeto `window`.
 1. Gestión de ventanas.
 2. Propiedades y métodos.
 2. Objeto `location`.
 3. Objeto `navigator`.
 4. Objeto `document`.
2. Objetos nativos en Javascript.
 1. Objeto `String`.
 1. Propiedades y métodos del objeto `String`.
 2. Objeto `Math`.
 3. Objeto `Number`.
 4. Objeto `Boolean`.
 5. Objeto `Date`.

1.- Descripción de la tarea

Caso práctico

Antonio ha completado correctamente la fase de introducción y fundamentos básicos del lenguaje JavaScript, y ahora comienza a investigar en las características de los objetos predefinidos en JavaScript.

Estos objetos le van a permitir gestionar ventanas, marcos, propiedades de los navegadores, de las URL, etc. en JavaScript.

Además, también va a poder realizar operaciones matemáticas, de fecha y de cadenas, con otros tantos objetos nativos del lenguaje JavaScript.

Antonio tiene una pequeña reunión con Ada y con su tutor Juan, para comentar los progresos realizados hasta este momento y se pone manos a la obra con esta nueva sección.

¿Qué te pedimos que hagas?

Realizar una pequeña aplicación en JavaScript que realice lo siguiente:

- ✓ Abra una nueva ventana no redimensionable.
 - Hacer una función y dentro de esa función:
 - Escribir en la nueva ventana `<h3>Ejemplo de Ventana Nueva</h3>`
 - URL Completa: XXXXX
 - Protocolo utilizado: XXXXX
 - Nombre en código del navegador: XXXXX
 - Que detecte si está JAVA disponible en esa ventana del navegador y si es así que escriba:
 - Java SI disponible en esta ventana, o bien.
 - Java NO disponible en esta ventana.
 - Que abra un iframe con el contenido de www.google.es y de 800x600.

Y ahora fuera del código de la función que siga haciendo lo siguiente:

- Que escriba en la ventana principal `<h1>TAREA DWEC03</h2><hr />`
- Que solicite: introduzca su nombre y apellidos.
- Que solicite: introduzca DIA de nacimiento.
- Que solicite: introduzca MES de nacimiento.
- Que solicite: introduzca AÑO de nacimiento.
- Una vez solicitados esos datos imprimirá en la ventana principal:
 - Buenos días XXXXX
 - Tu nombre tiene XX caracteres, incluidos espacios.
 - La primera letra A de tu nombre está en la posición: X
 - La última letra A de tu nombre está en la posición: X
 - Tu nombre menos las 3 primeras letras es: XXXXXXXX
 - Tu nombre todo en mayúsculas es: XXXXXXXX
 - Tu edad es: XX años.
 - Naciste un feliz XXXXXX del año XXXX.
 - El coseno de 180 es: XXXXXXXXXX
 - El número mayor de (34,67,23,75,35,19) es: XX

- Ejemplo de número al azar: XXXXXXXXXX

Donde aparecen las XXXX tendrá que aparecer el cálculo o texto que corresponda.

2.- Información de interés

Recursos necesarios

Editor de texto para teclear el código de la aplicación y un navegador web.

Consejos y recomendaciones

Mirar las referencias y códigos de ejemplo de la página W3Schools:

[Página de referencia de w3schools.](#)

3.- Evaluación de la tarea

Criterios de evaluación implicados

- a) Se han identificado los objetos predefinidos del lenguaje.
- b) Se han analizado los objetos referentes a las ventanas del navegador y los documentos web que contienen.
- c) Se han escrito sentencias que utilicen los objetos predefinidos del lenguaje para cambiar el aspecto del navegador y el documento que contiene.
- d) Se han generado textos y etiquetas como resultado de la ejecución de código en el navegador.
- e) Se han escrito sentencias que utilicen los objetos predefinidos del lenguaje para interactuar con el usuario.
- f) Se han utilizado las características propias del lenguaje en documentos compuestos por varias ventanas y marcos.
- g) Se han utilizado “cookies” para almacenar información y recuperar su contenido.
- h) Se ha depurado y documentado el código.

¿Cómo valoramos y puntuamos tu tarea?

Rúbrica de la tarea

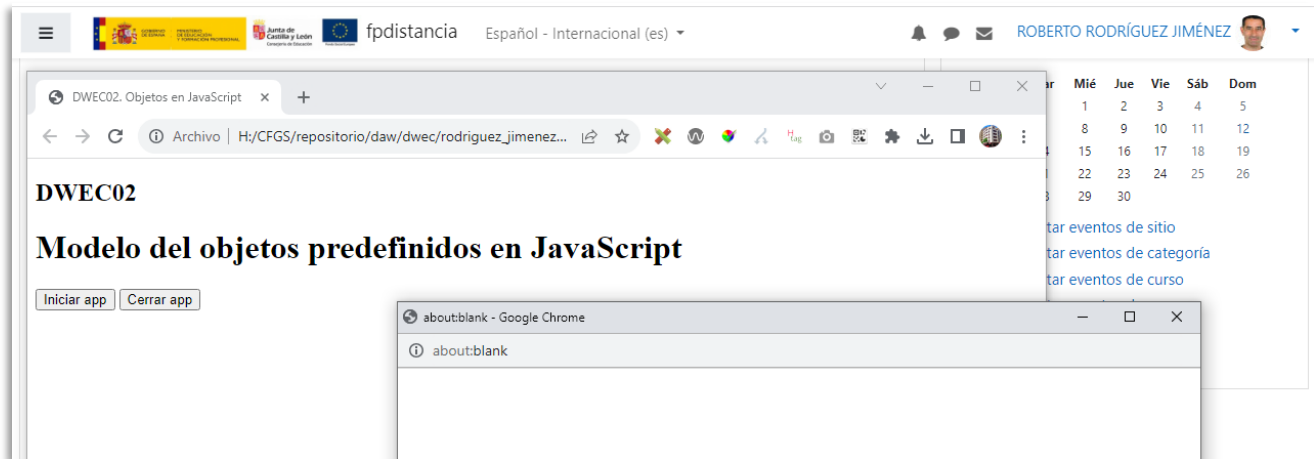
Creación de la función que se pide.	5 puntos.
Petición de los datos.	2 puntos.
Visualización de los mismos.	2 puntos.
Claridad y presentación del código del ejercicio, comentarios en el código y su indentación.	1 punto.

RESOLUCIÓN

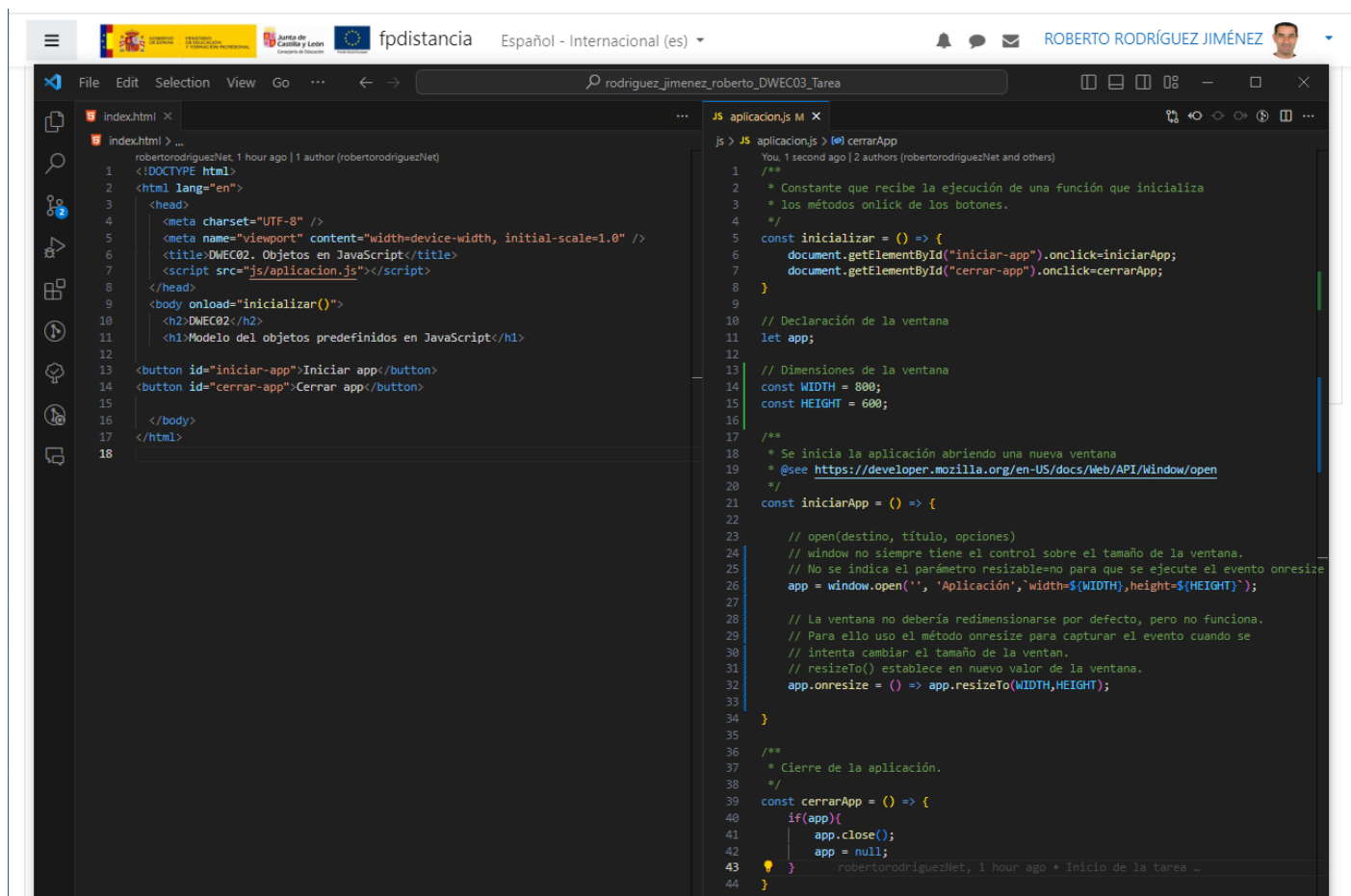
Abra una nueva ventana no redimensionable.

index.html contiene los controles para iniciar y cerrar la aplicación.

js/aplicacion.js es el script que ejecuta la aplicación.



Página con los botones y ventana nueva abierta.



Código para index.html y aplicacion.js

Problemas encontrados

El objeto `window` no siempre tiene el control sobre el tamaño de la ventana. Los navegadores dan el control al usuario, que es quien decide si redimensiona o no la ventana.

Las ventanas creadas con `window.open()` no pueden, por defecto, redimensionarse a no ser que se indique con el parámetro `resizable=yes`. También es posible indicar `resizable=no`.

Para mantener el tamaño original, que no evitar el redimensionamiento, se escucha el evento `onresize` y se llama a la función `resizeTo(w,h)` que establece un nuevo valor para la dimensión de la ventana, en este caso, la original.

Sin embargo, si se establece `resizable=no`, se evita el redimensionamiento (aunque no sea efectivo) por lo que no se invoca el evento `onresize` y, por lo tanto, la ventana puede cambiar sus dimensiones.

Solución

No impedir el redimensionamiento en la construcción de la ventana y capturar ese redimensionamiento para poder devolver el valor mediante `resizeTo()`.

Fuentes:

<https://developer.mozilla.org/en-US/docs/Web/API/Window/open>

<https://es.javascript.info/popup-windows>

```
const inicializar = () => {
  document.getElementById("iniciar-app").onclick=iniciarApp;
  document.getElementById("cerrar-app").onclick=cerrarApp;
}

let app;

const WIDTH = 800;
const HEIGHT = 600;

const iniciarApp = () => {
  app = window.open('', 'Aplicación', `width=${WIDTH},height=${HEIGHT}`);
  app.onresize = () => app.resizeTo(WIDTH,HEIGHT);
}

const cerrarApp = () => {
  if(app){
    app.close();
    app = null;
  }
}
```

Código para aplicacion.js

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>DWEC02. Objetos en JavaScript</title>
    <script src="js/aplicacion.js"></script>
  </head>
  <body onload="inicializar()">
    <h2>DWEC02</h2>
    <h1>Modelo del objetos predefinidos en JavaScript</h1>

    <button id="iniciar-app">Iniciar app</button>
    <button id="cerrar-app">Cerrar app</button>

  </body>
</html>
```

Código para index.html

Hacer una función y dentro de esa función:

Función `crearContenido()`

```
function crearContenido() {
  // Obtener los valores de los objetos.
  // Algunas de estas propiedades, y algún método, están obsoletas (noviembre de 2023).
  let documento = app.document;
  let url = location.href;
  let protocolo = location.protocol;

  // Para obtener la información del navegador se usa userAgent,
  // que contiene la información que da appCodeName() (obsoleta).
  let navegador = navigator.userAgent;

  // javaEnabled() en un método obsoleto.
  // No se recomienda su uso y la alternativa es consultar
  // la compatibilidad del navegador.
  let java = navigator.javaEnabled()? "Sí" : "NO";

  // Header con el título.
  return (
    `<h3>Ejemplo de nueva ventana</h3>
    <dl>
      <dt>URL:</dt><dd>${url}</dd>
      <dt>Protocolo:</dt><dd>${protocolo}</dd>
      <dt>Navegador:</dt><dd>${navegador}</dd>
      <dt>Java habilitado:</dt><dd>Java ${java} disponible en esta ventana</dd>
```



```

    </dl>

    <iframe id="iframe-goole" title="Google" width="800" height="600"
src="www.google.es"></iframe>`
    );
}

```

La función devuelve el código HTML, no lo escribe, de esta manera puede ser utilizado en diferentes lugares y en diferentes momentos.

La escritura del código con el contenido se realiza siguiendo el hilo de ejecución de la función `iniciarApp()`.

```

const iniciarApp = () => {
  // Cerramos la ventana si ya tenemos una instancia
  if (app != null) {
    app.close();

    app = window.open("", "Aplicación", `width=${WIDTH},height=${HEIGHT}`);

    app.onresize = () => app.resizeTo(WIDTH, HEIGHT);

    // -- CONTENIDO PARA LA APLICACIÓN -----
    // Llamar a la función que creará el contenido
    app.document.write(crearContenido());

    . . .
  }
}

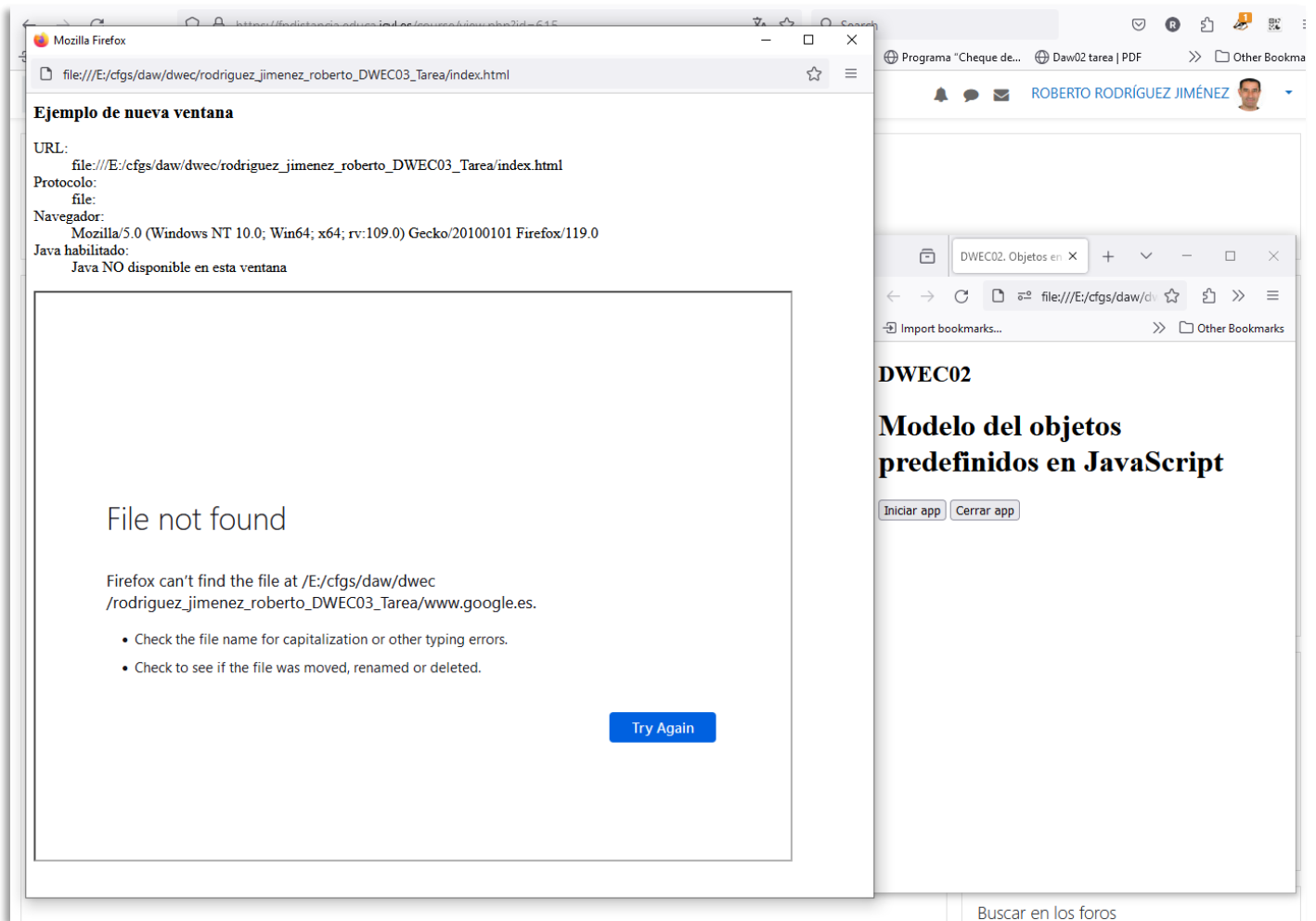
```

Todos los datos se asignan previamente a variables para que sean procesadas si es necesario, como ocurre con la habilitación de Java.

Por otro lado, `javaEnabled()` es una función obsoleta y solo se mantiene por motivos de compatibilidad. En el caso de ser necesaria la comprobación, debería consultarse la compatibilidad del navegador.

```
let java = navigator.javaEnabled()? "SÍ" : "NO";
```

El `iframe` no carga la url www.google.com por que la petición es rechazada por el servidor



Ejecución de la primera parte del ejercicio

Y ahora fuera del código de la función que siga haciendo lo siguiente:

La ejecución del código se sigue desarrollando en *aplicacion.js* fuera de la función creada para la primera parte del ejercicio.

```
const iniciarApp = () => {  
  // Cerramos la ventana si ya tenemos una instancia  
  if (app != null) {  
    app.close();  
  }  
  
  app = window.open("", "Aplicación", `width=${WIDTH},height=${HEIGHT}`);  
  app.onresize = () => app.resizeTo(WIDTH, HEIGHT);  
  
  app.document.write(crearContenido());  
  
  // -- GESTIÓN DE LOS DATOS QUE SE MOSTRARÁN EN LA VENTANA PRINCIPAL ---  
  // A partir de aquí se desarrolla la segunda parte del ejercicio
```

La forma en la que vamos a escribir el código en la ventana principal es agregando elementos, ya que el método *write()* sobre escribe el código de *document*.

Para escribir el elemento *h1* se crea un elemento “h1” y se añade un *TextNode* con el texto que se va a mostrar. Finalmente, el elemento *h1* se añade al elemento *body* de la ventana principal justo antes de la botonera.

```
// Referenciamos la botonera para poder insertar los elementos correctamente  
let botonera = document.getElementById("botonera");  
  
// Establecemos el h1  
// Crear el elemento que se va a insertar  
let h1 = document.createElement('h1');  
  
// Crear el texto del elemento  
let h1Text = document.createTextNode('TAREA DWEC03');  
  
// Insertar el texto en el elemento  
h1.appendChild(h1Text);  
  
// Insertar el elemento en el cuerpo de la ventana principal  
// antes de la botonera  
document.body.insertBefore(h1, botonera);
```

Una vez que hemos escrito el encabezado, seguimos pidiendo los datos al usuario.

Los datos se declaran y procesan antes de ser mostrados.

Para poder mostrarla edad se ha declarado un método *getEdad(fecha)* que devuelve la edad calculando la diferencia, en milisegundos, entre la fecha dada por el usuario y el momento actual. Para que se pueda realizar la operación, se genera una variable de tipo *Date* a partir de la fecha dada.

```
// Solicitud de datos
let nombre_apellidos = app.prompt("Escriba su nombre y apellidos");
let dia = parseInt(app.prompt("Día de nacimiento"));
let mes = parseInt(app.prompt("Mes de nacimiento (número)"));
let anno = parseInt(app.prompt("Año de nacimiento"));
let fecha = new Date(anno, mes, dia);
```

Ahora toca mostrar los datos, para lo cual se crea una lista con los requisitos de la tarea.

En lugar de usar *TextNode*, en esta ocasión se crea la lista desordenada *ul* y se añade el texto que se va a mostrar mediante *ul.innerHTML*.

El texto con las líneas se va a cargar en un solo bloque.

```
// Insertamos una lista como nodo para la ventana principal
// y ajustamos los estilos.
let lista = document.createElement('ul');
lista.style.listStyle = 'none';
lista.style.padding = 0;

// Creamos los elementos de la lista
lista.innerHTML = `<li>Buenos días ${nombre_apellidos}</li>
<li>Tu nombre tiene ${nombre_apellidos.length} caracteres, incluidos espacios.</li>
<li>La primera letra A de tu nombre está en la posición ${nombre_apellidos.indexOf("a") + 1}</li>
<li>Tu nombre menos las 3 primeras letras es: ${nombre_apellidos.substring(3)}</li>
<li>Tu nombre todo en mayúsculas es: ${nombre_apellidos.toUpperCase()}</li>
<li>Tu edad es: ${isNaN(getEdad(fecha)) ? " -- " : getEdad(fecha)} años.</li>
<li>Naciste un feliz ${
  typeof semana[fecha.getDay()] === "undefined"
    ? " -- "
    : semana[fecha.getDay()]
} del año ${isNaN(anno) ? " ---- " : anno}</li>
<li>El coseno de 180 es: ${Math.cos(180)}</li>
<li>El número mayor de 6(34,7,23,75,35,19) es: ${Math.max(34,7,23,75,35,19)}</li>
<li>Ejemplo de número al azar: ${parseInt(Math.random() * 1000)}</li>`;
```

Finalmente, se inserta antes de la botonera, al igual que el encabezado.

```
// Inserta la lista antes de la botonera
document.body.insertBefore(lista, botonera);
```

Para obtener el año se ha creado una función que devuelve el año a partir de una cantidad de milisegundos que son el resultado de restar dos fechas: la de nacimiento de la de el momento actual.

Para poder restar las fechas en milisegundos, estas deben ser objetos del tipo Date.

```
/**
 * @param Date fecha es la fecha en tipo Date
 * @return int el año
 */
function getEdad(fecha){

  // Obtener la edad en milisegundos
  // Date.now() obtiene el momento actual en milisegundos
  // edadEnMilisegundos es un objeto Date con el poder trabajar con fechas.
  let edadEnMilisegundos = new Date( Date.now() - fecha );

  // Devolvemos los años entre la edadEnMilisegundos y 1970.
  // El 1/1/1970 se toma como en inicio de la cuenta del tiempo (Timestamp)
  return Math.abs(edadEnMilisegundos.getUTCFullYear() - 1970);
}
```

Como ya tenemos el objeto de tipo Date *fecha*, resulta sencillo obtener datos a través de sus métodos.

Ya que `getDay()` devuelve el número del día de la semana, he declarado un array con los nombres de los días, ordenados dependiendo de la configuración del primer día de la semana.

```
// Días de la semana
const semana = ["jueves", "viernes", "sábado", "domingo", "lunes", "martes", "miércoles"];
```

```
<li>Naciste un feliz ${
  typeof semana[fecha.getDay()] === "undefined"
    ? " -- "
    : semana[fecha.getDay()]
} del año ${isNaN(anno) ? " ---- " : anno}</li>
```

`Math.random()` devuelve un número entre 0 y 1, por lo que aparece un número con muchos decimales, es por ello por lo que multiplica por 1.000 y se convierte a entero.

```
Ejemplo de número al azar: ${parseInt(Math.random() * 1000)}
```

Resultado

The screenshot shows a web browser window with a single tab titled "DWEC02. Objetos en JavaScript". The address bar displays the file path "file:///E:/cfgs/daw/dwec/roc". The page content is divided into a sidebar and a main area. The sidebar, on the left, has a header "Desarrollo Web en E" and a sub-header "Desarrollo Web en Entorno". Under "Recursos generales.", there are links for "Tutoría por chat.", "Foro de novedades.", "Mensaje de bienvenida", "Glosario para DWEC.", and "Versiones imprimibles". The main area has a header "TAREA DWEC03" and a list of tasks:

- Buenos días .
- Tu nombre tiene 0 caracteres, incluidos espacios.
- La primera letra A de tu nombre está en la posición 0.
- La última letra A de tu nombre está en la posición: 0.
- Tu nombre menos las 3 primeras letras es: .
- Tu nombre todo en mayúsculas es: .
- Tu edad es: 54 años.
- Naciste un feliz domingo del año 1969.
- El coseno de 180 es: -0.5984600690578581.
- El número mayor de 6(34,7,23,75,35,19) es: 75
- Ejemplo de número al azar: 502

At the bottom of the main area, there are two buttons: "Iniciar app" and "Cerrar app".

Captura de ejecución de la segunda parte de la tarea

Código

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>DWEC02. Objetos en JavaScript</title>
    <script src="js/aplicacion.js"></script>
  </head>
  <body onload="inicializar()">

    <div id="botonera" style="margin-top: 20px">
      <button id="iniciar-app">Iniciar app</button>
      <button id="cerrar-app">Cerrar app</button>
    </div>

  </body>
</html>
```

aplicacion.js

```
// Declaración de la ventana
let app;

// Dimensiones de la ventana
const WIDTH = 900;
const HEIGHT = 900;

/**
 * Constante que recibe la ejecución de una función que inicializa
 * los métodos onclick de los botones.
 */
const inicializar = () => {
  document.getElementById("iniciar-app").onclick = iniciarApp;
  document.getElementById("cerrar-app").onclick = cerrarApp;
};

/**
 * Se inicia la aplicación abriendo una nueva ventana
 * @see https://developer.mozilla.org/en-US/docs/Web/API/Window/open
 */
```

```
const iniciarApp = () => {
  // Cerramos la ventana si ya tenemos una instancia
  if (app !== null) {
    app.close();
  }

  // -- APERTURA Y REDIMENSIÓN DE LA VENTANA -----
  // open(destino, título, opciones)
  // window no siempre tiene el control sobre el tamaño de la ventana.
  // No se indica el parámetro resizable=no para que se ejecute el evento onresize
  app = window.open("", "Aplicación", `width=${WIDTH},height=${HEIGHT}`);

  // La ventana no debería redimensionarse por defecto, pero no funciona.
  // Para ello uso el método onresize para capturar el evento cuando se
  // intenta cambiar el tamaño de la ventana.
  // resizeTo() establece en nuevo valor de la ventana.
  app.onresize = () => app.resizeTo(WIDTH, HEIGHT);

  // -- CONTENIDO PARA LA APLICACIÓN -----
  // Llamar a la función que creará el contenido
  app.document.write(crearContenido());

  // -- GESTIÓN DE LOS DATOS QUE SE MOSTRARÁN EN LA VENTANA PRINCIPAL ---
  // A partir de aquí se desarrolla la segunda parte del ejercicio

  // Días de la semana, ordenados según el inicio de la semana
  const semana = [
    "jueves",
    "viernes",
    "sábado",
    "domingo",
    "lunes",
    "martes",
    "miércoles",
  ];

  // Referenciamos la botonera para poder insertar los elementos correctamente
  let botonera = document.getElementById("botonera");

  // Establecemos el h1
  // Crear el elemento que se va a insertar
  let h1 = document.createElement('h1');

  // Crear el texto del elemento
  let h1Text = document.createTextNode('TAREA DWEC03');

  // Insertar el texto en el elemento
  h1.appendChild(h1Text);

  // Insertar el elemento en el cuerpo de la ventana principal
  // antes de la botonera
  document.body.insertBefore(h1, botonera);
```



```

// Solicitud de datos
let nombre_apellidos = app.prompt("Escriba su nombre y apellidos");
let dia = parseInt(app.prompt("Día de nacimiento"));
let mes = parseInt(app.prompt("Mes de nacimiento (número)"));
let anno = parseInt(app.prompt("Año de nacimiento"));
let fecha = new Date(anno, mes, dia);

// Insertamos una lista como nodo para la ventana principal
// y ajustamos los estilos.
let lista = document.createElement('ul');
lista.style.listStyle = 'none';
lista.style.padding = 0;

// Creamos los elementos de la lista
lista.innerHTML = `<li>Buenos días ${nombre_apellidos}</li>
<li>Tu nombre tiene ${nombre_apellidos.length} caracteres, incluidos espacios.</li>
<li>La primera letra A de tu nombre está en la posición ${nombre_apellidos.indexOf("a") +
1}</li>
<li>Tu nombre menos las 3 primeras letras es: ${nombre_apellidos.substring(3)}</li>
<li>Tu nombre todo en mayúsculas es: ${nombre_apellidos.toUpperCase()}</li>
<li>Tu edad es: ${isNaN(getEdad(fecha)) ? " -- " : getEdad(fecha)} años.</li>
<li>Naciste un feliz ${
    typeof semana[fecha.getDay()] === "undefined"
    ? " -- "
    : semana[fecha.getDay()]
} del año ${isNaN(anno) ? " ---- " : anno}</li>
<li>El coseno de 180 es: ${Math.cos(180)}</li>
<li>El número mayor de 6(34,7,23,75,35,19) es: ${Math.max(34,7,23,75,35,19)}</li>
<li>Ejemplo de número al azar: ${parseInt(Math.random() * 1000)}</li>`;

// Inserta la lista antes de la botonera
document.body.insertBefore(lista, botonera);

};

/**
 * Cierre de la aplicación.
 */
const cerrarApp = () => {
  if (app) {
    app.close();
    app = null;
  }
};

/**
 * Función para dar contenido a la ventana abierta
 */
function crearContenido() {
  // Obtener los valores de los objetos.

```

```

// Algunas de estas propiedades, y algún método, están obsoletas (noviembre de 2023).
let documento = app.document;
let url = location.href;
let protocolo = location.protocol;

// Para obtener la información del navegador se usa userAgent,
// que contiene la información que da appCodeName() (obsoleta).
let navegador = navigator.userAgent;

// javaEnabled() en un método obsoleto.
// No se recomienda su uso y la alternativa es consultar
// la compatibilidad del navegador.
let java = navigator.javaEnabled() ? "Sí" : "NO";

// Header con el título.
return `

### Ejemplo de nueva ventana</h3> <dl> <dt>URL:</dt><dd>${url}</dd> <dt>Protocolo:</dt><dd>${protocolo}</dd> <dt>Navegador:</dt><dd>${navegador}</dd> <dt>Java habilitado:</dt><dd>Java ${java} disponible en esta ventana</dd> </dl> <iframe id="iframe-goole" title="Google" width="800" height="600" src="www.google.es"></iframe>`; } /** * Devuelve los años resultantes de obtener la diferencia * entre dos fechas. * @param {Date} fecha la fecha de nacimiento * @returns la edad */ function getEdad(fecha) { // Obtener la edad en milisegundos // Date.now() obtiene el momento actual en milisegundos // edadEnMilisegundos es un objeto Date con el poder trabajar con fechas. let edadEnMilisegundos = new Date(Date.now() - fecha); // Devolvemos los años entre la edadEnMilisegundos y 1970. // El 1/1/1970 se toma como en inicio de la cuenta del tiempo (Timestamp) return Math.abs(edadEnMilisegundos.getUTCFullYear() - 1970); }


```

Recursos

Documentar JavaScript (jsdoc)

<https://www.gradiweb.com/es/marketing-digital/documentacion-de-codigo-para-javascript-con-jsdoc/>

Manual JavaScript

<https://www.w3schools.com/js/default.asp>

Referencias sobre JavaScript

<https://developer.mozilla.org/es/>