Grado en Ingeniería Informática del Software

Software y estándares para la Web

P7. JQUERY Y CONSUMO DE SERVICIOS WEB

Escuela de Ingeniería Informática
Universidad de Oviedo

Contenido

Objetivos

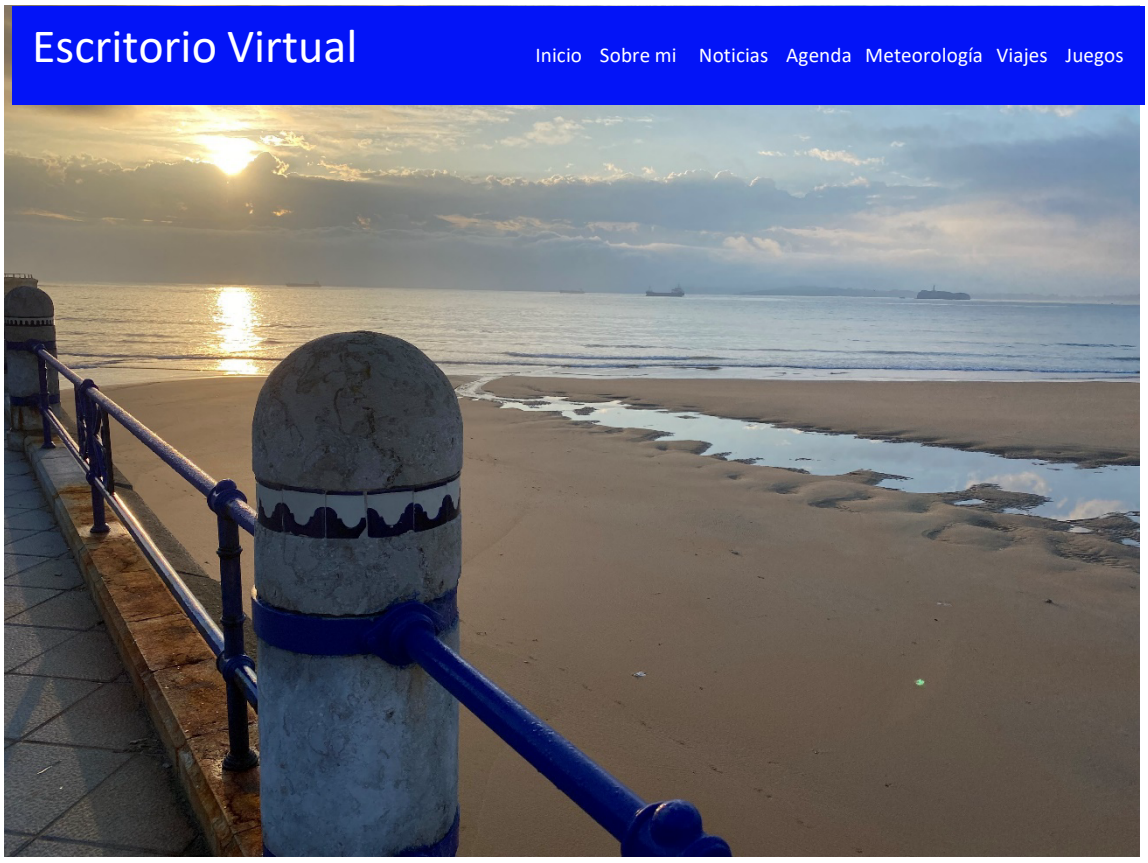
En esta práctica se va a realizar:

- La creación de un documento de script utilizando el estándar ECMAScript
- La creación de objetos para la realización de diferentes tareas en combinación con HTML.
- La generación de código HTML desde el documento de script utilizando la biblioteca jQuery y su posterior inserción en el documento HTML original.
- El consumo de servicios web a través de llamadas AJAX utilizando la librería jQuery y la inserción de la información obtenida del servicio en los documentos HTML utilizando la librería jQuery.
- La validación del código HTML estático y el código HTML generado

IMPORTANTE: Recuerda las pautas de trabajo establecidas en la primera sesión de prácticas (P0. Pautas de trabajo): valida todos los documentos HTML, valida todas las hojas de estilo CSS, comprueba la adaptabilidad y la accesibilidad con las herramientas proporcionadas.

Temática del proyecto: Escritorio virtual (Virtual Desktop)

El proyecto Escritorio Virtual (Virtual Desktop) es evolutivo y será creado, completado y modificado en las diferentes prácticas de la asignatura.



Ejercicio 1: Introducción a AJAX utilizando jQuery

jQuery es una librería escrita en JavaScript que está destinada, entre otras cuestiones, a facilitar la creación y manipulación del árbol DOM de un documento HTML.

En este ejercicio se utilizará la librería jQuery para realizar una llamada AJAX a un servicio web de imágenes para obtener una imagen que se aplique como imagen de fondo al fichero index.html del proyecto de escritorio virtual.

El servicio web de imágenes recomendado para la resolución de este ejercicio es Flickr. https://www.flickr.com/

NOTA: Al utilizar jQuery para establecer la imagen de fondo se creará en el documento index.html un atributo style que establece el estilo correspondiente para situar la imagen como fondo de pantalla de dicho documento html. Esto se corresponde con el funcionamiento normal de jQuery y está permitido para este ejercicio.

Tarea 1. Creación de un nuevo documento JavaScript

Dentro de la carpeta js del directorio del proyecto Escritorio Virtual crea un nuevo fichero llamado fondo.js.

Añade una referencia a este nuevo fichero en el documento index.html.

Utiliza el siguiente enlace para saber cómo es la referencia que debes incluir en el fichero index.html para utilizar la opción minificada de jQuery Core 3.x: https://releases.jquery.com/

Tarea 2. Creación de la clase Fondo

Dentro del fichero creado en la tarea anterior crea la clase Fondo.

Añade a esta clase un método constructor que reciba como parámetro el nombre del país, el nombre de la capital y las coordenadas de la capital utilizados en la clase Pais.

Almacena estos valores en atributos dentro de la clase Fondo.

Tarea 3. Creación del método de consulta a la API de Flickr

Crea un método que realice una consulta AJAX a la API de Flickr para obtener una imagen del país.

Realiza la llamada a la API de tal forma que esta conteste en formato JSON.

Guía para resolver la tarea 3

Consulta los diferentes métodos existentes en la API de Flickr a través de este enlace:

https://www.flickr.com/services/api/

Identifica el método que permita obtener las fotos de una localización concreta y realiza la llamada de tal manera que devuelva una imagen asociada a la capital del país cuya información está contenida en la clase fondo.

Consulta el ejemplo incluido a continuación para conocer la forma en la que debe realizarse una llamada AJAX a través de jQuery:

http://di002.edv.uniovi.es/~cueva/JavaScript/44jQueryJSON.html

Tarea 4. Estableciendo la imagen de fondo

Utiliza una de las imágenes devueltas por el servicio de imágenes para establecerla como imagen de fondo del documento index.html

Guía para resolver la tarea 4

Consultar el siguiente enlace para conocer la forma en la que se puede acceder a los diferentes elementos del html utilizando la librería jQuery:

https://learn.jquery.com/using-jquery-core/selecting-elements/

Utiliza las fórmulas de acceso para encontrar el elemento que debe tener la imagen de fondo y establece la imagen de fondo en dicho elemento. Consulta la sección de css de jQuery para obtener información sobre como establecer la imagen de fondo:

https://api.jquery.com/css/

Resultado del ejercicio 1

Una vez completado el ejercicio deberían haberse añadido los siguientes ficheros al proyecto del Escritorio Virtual:

- Fichero fondo.js en el directorio js del proyecto

Además, se debe haber modificado el fichero index.html para incluir la referencia al fichero JavaScript mencionado anteriormente y para incluir la llamada que permita establecer la imagen de fondo en dicho documento.

Ejercicio 2: Consulta de servicio de meteorología con jQuery

En este ejercicio se utilizará la librería jQuery para realizar una consulta a un servicio web de meteorología que devuelva la información del tiempo en un lugar.

El servicio web de meteorología recomendado para la resolución de este ejercicio es OpenWeatherMap. http://openweathermap.org/

Tarea 1. Modificación del documento pais.js

Modifica el documento país.js para incluir en él el código necesario para realizar una llamada AJAX a través de jQuery al servicio web de meteorología y obtener la previsión del tiempo en la capital del país para los próximos 5 días.

Haz que el servicio web devuelva la información del tiempo en formato JSON y con unidades de medida del sistema métrico.

Guía para resolver la tarea 1

Consulta los diferentes métodos existentes en la API de OpenWeatherMap a través del siguiente enlace, prestando especial atención a aquellos que pronostican el tiempo a varios días vista (forecast) y a los parámetros que reciben para su invocación:

https://openweathermap.org/api

Crea una cuenta gratuita en la web de OpenWeatherMap para obtener una API Key que te permita realizar peticiones a los métodos de la API existentes en el enlace anterior.

Consulta el ejemplo incluido a continuación para conocer la forma en la que debe realizarse una llamada AJAX a través de jQuery para la API de OpenWeatherMap:

http://di002.edv.uniovi.es/~cueva/JavaScript/86-jQuery-JSON-meteo.html

Tarea 2. Modificación del documento meteorologia.html

Utiliza la información devuelta por el servicio web de meteorología en la tarea anterior para escribir en el documento meteorología.html el pronóstico del tiempo para los siguientes 5 días.

Por cada día de pronóstico se debe incluir al menos la siguiente información: temperatura máxima, temperatura mínima, porcentaje de humedad, un icono que represente el tiempo que va a hacer y la cantidad de lluvia del día.

Utiliza el siguiente enlace para saber cómo es la referencia que debes incluir en el fichero meteorologia.html para utilizar la opción minificada de jQuery Core 3.x: https://releases.jquery.com/

IMPORTANTE: El formato de presentación se deja libre al estudiante, valorándose la calidad de dicho formato y el uso de los diferentes conceptos de CSS.

Guía para resolver la tarea 2

Consulta el siguiente enlace para conocer los métodos existentes en jQuery que permiten crear los diferentes elementos en el documento html para representar la información.

https://api.jquery.com/add/#post-8

Consulta los siguientes ejemplos para ver formas en las que añadir, modificar y eliminar elementos y propiedades de elementos:

http://di002.edv.uniovi.es/~cueva/JavaScript/80-jQuery-DOM-set-val-text-html.html

http://di002.edv.uniovi.es/~cueva/JavaScript/81-jQuery-DOM-set-attr.html

http://di002.edv.uniovi.es/~cueva/JavaScript/82-jQuery-DOM-add-elementos.html

Modifica la hoja de estilos estilo.css para añadir en ella los estilos que sean necesarios para que los datos del tiempo se visualicen correctamente.

Resultado del ejercicio 2

Una vez completado el ejercicio deberían haberse modificado los siguientes ficheros: el fichero país.js para incluir el código de llamada al servicio de meteorología, el fichero meteorología.html para incluir en el la información del pronóstico del tiempo y el fichero estilo.css para incluir los estilos necesarios para presentar el pronóstico del tiempo.

Ejercicio 3: Agenda de Fórmula 1

En este ejercicio se utilizará la librería jQuery para realizar una consulta a un servicio web de información relacionado con la Fórmula 1. Será necesario realizar una consulta que devuelva la información relacionada con las carreras de esta temporada 2023.

El servicio web de consulta de información es la API Ergast http://ergast.com/mrd/

NOTA: Esta API no permite que se hagan más de 4 peticiones por segundo o 200 peticiones a la hora, por lo que es recomendable (y necesario) almacenar la información que devuelve en una variable dentro del código y solamente realizar una llamada real a la API en un intervalo grande de tiempo (por ejemplo, 1 llamada cada 10 minutos).

Tarea 1. Creación de un nuevo documento JavaScript

Dentro de la carpeta js del directorio del proyecto Escritorio Virtual crea un nuevo fichero llamado agenda.js.

Añade una referencia a este nuevo fichero en el documento agenda.html

Tarea 2. Creación de la clase Agenda

Dentro del fichero creado en la tarea anterior crea la clase Agenda.

Añade a esta clase un método constructor que inicialice en el objeto un atributo con la URL de la dirección que permitirá consultar la información de las carreras de la temporada en curso.

Además deben existir también otros dos atributos:

- Un atributo llamado *last_api_call* que almacene el momento temporal de la última petición a la API, inicializado a *null*.
- Un atributo llamado *last_api_result* que almacene la última respuesta que ha dado la API a la consulta en cuestión, inicializado a *null*.

La información que devuelva la API debe estar en formato XML.

Guía para resolver la tarea 2

Accede a la siguiente dirección para consultar la información general de la API y conocer la forma para configurar los tipos de respuesta de la misma.

http://ergast.com/mrd/

Consulta en la dirección anterior el apartado *Race Schedule* dentro del menú *API Documentation* para conocer la forma en la que se debe invocar al servicio para que devuelva las carreras de la temporada en curso.

Tarea 3. Llamada al servicio web

Modifica el código del fichero agenda.js para añadir un método que permita realizar la consulta de la información sobre las carreras de la presente temporada.

Una vez obtenida la información de las carreras, el método debe recorrer dicha información y crear una estructura de elementos en el html que permita visualizar la información de cada una de las carreras de la temporada. Como mínimo se debe mostrar:

- Nombre de la carrera
- Nombre del circuito donde se celebra
- Coordenadas del circuito
- Fecha y hora de la carrera

Guía para resolver la tarea 3

Consulta el ejemplo incluido a continuación para conocer la forma en la que debe realizarse una llamada AJAX a través de jQuery y el posterior tratamiento de la respuesta cuando la API responde en formato XML:

http://di002.edv.uniovi.es/~cueva/JavaScript/87-jQuery-AJAX-XML-meteo.html

Recuerda añadir una comprobación antes del código que realice la llamada de AJAX para saber si han pasado los minutos del intervalo establecido entre consultas. Utiliza en esta comprobación el atributo creado en la tarea 2 para esta cuestión.

> **NOTA:** En el caso de que no se haya cumplido el intervalo de tiempo establecido desde la última llamada, hacer que el método de consulta a la API no ejecute la llamada AJAX y devuelva la información contenida en la variable *last_api_result*.

Actualiza el valor del atributo *last_api_call* con el momento de la petición actual en el caso de que se haya realizado dicha petición.

Tarea 4. Modificación del documento agenda.html

Modifica el código del fichero agenda.html para añadir un botón que al ser pulsado solicite la información de las carreras de esta temporada a la clase Agenda y muestre esa información por pantalla.

Utiliza el siguiente enlace para saber cómo es la referencia que debes incluir en el fichero index.html para utilizar la opción minificada de jQuery Core 3.x: https://releases.jquery.com/

IMPORTANTE: El formato de presentación se deja libre al estudiante, valorándose la calidad de dicho formato y el uso de los diferentes conceptos de CSS.

Guía para resolver la tarea 4

Consulta el ejemplo incluido a continuación para conocer la forma en la que se utilizan los botones para que ejecuten un código tras su pulsación:

http://di002.edv.uniovi.es/~cueva/JavaScript/114-evento-botones.html

Añade el código html necesario para visualizar la información de cada carrera utilizando los métodos existentes en jQuery para modificar el documento html. Se recomienda consultar el siguiente enlace:

http://di002.edv.uniovi.es/~cueva/JavaScript/80-jQuery-DOM-set-val-text-html.html

http://di002.edv.uniovi.es/~cueva/JavaScript/81-jQuery-DOM-set-attr.html

http://di002.edv.uniovi.es/~cueva/JavaScript/82-jQuery-DOM-add-elementos.html

Resultado del ejercicio 3

Una vez completado el ejercicio deberían haberse añadido los siguientes ficheros al proyecto del Escritorio Virtual:

- Fichero agenda.js en el directorio js del proyecto
- Fichero agenda.css en el directorio estilo del proyecto

Además, se debe haber modificado el fichero agenda.html para incluir el enlace de referencia al fichero agenda.js y el resto de código necesario para completar el ejercicio.

Ejercicio 4: Crucigrama Matemático

En este ejercicio se creará un juego de crucigrama matemático que utilizará jQuery para recorrer el árbol DOM del documento html y para crear y manipular diferentes elementos del html desde el documento de script. Estas llamadas de jQuery harán las mismas funciones que las llamadas realizadas a los métodos *document.write* y *document.querySelector* en los ejercicios anteriores.

Un crucigrama matemático es la analogía, con números y operaciones, al típico pasatiempo del crucigrama de palabras. Existirá una cuadrícula con casillas deshabilitadas (en color negro) y con otras casillas habilitadas en las que habrá que escribir números y/u operadores matemáticos para completar las ecuaciones que forman parte del crucigrama.

A diferencia de los otros juegos realizados hasta el momento, el crucigrama matemático debe ser capaz de calcular el tiempo que ha tardado el usuario en completar el crucigrama en horas, minutos y segundos. Para ello será necesario utilizar dos atributos del tipo predefinido Date que almacenen el momento de inicio y fin del juego respectivamente y posteriormente calcular la diferencia entre ambas para obtener el tiempo empleado en completar el crucigrama.

NOTA: La interfaz y el funcionamiento de este ejercicio se reutilizará en las prácticas de PHP, donde será necesario implementar la funcionalidad que permita guardar los récords de resolución de los crucigramas de los diferentes usuarios.

Tarea 1. Creación del documento crucigrama.html

Crea el fichero crucigrama.html en el directorio raíz del proyecto de Escritorio Virtual.

Añade al menú de juegos creado en el laboratorio de la semana anterior un enlace que permita el acceso al fichero crucigrama.html

Tarea 2. Creación de un nuevo documento JavaScript

Dentro de la carpeta js del directorio del proyecto Escritorio Virtual crea un nuevo fichero llamado crucigrama.js.

Añade una referencia a este nuevo fichero en el documento crucigrama.html

Tarea 3. Creación de la clase Crucigrama

Dentro del fichero creado en la tarea anterior crea la clase Crucigrama.

Añade a la declaración de la clase los siguientes atributos:

- Una cadena llamada *board* que represente el tablero del juego
- Un variable con un entero que represente el número de columnas del tablero (9)
- Un variable con un entero que represente el número de filas del tablero (11)
- Una variable *init_time* que represente el momento en el que se inicia el juego, sin inicializar
- Una variable *end_time* que represente el momento en el que se termina el juego, sin inicializar
- Una variable de tipo array que contendrá la representación del tablero, sin inicializar.

Añade a la clase Crucigrama un método constructor que inicialice el tablero a un array bidimensional del tamaño que le corresponde.

Guía para resolver la tarea 3

Inicializa el array bidimensional a partir de las variables con el número de filas y el número de columnas del crucigrama.

La cadena que representa el tablero del juego se puede inicializar, dentro o fuera del constructor, con uno de los siguientes valores (incluyendo las comillas):

- **Nivel fácil:**
  "4,*,.,=,12,#,#,#,5,#,#,*,#,/,#,#,#,*,4,-
  ,.,=,.,#,15,#,.,*,#,=,#,=,#,/,#,=,.,#,3,#,4,*,.,=,20,=,#,#,#,#,#,=,#,#,8,#,9,-,.,=,3,#,.,#,#,-
  ,#,+,#,#,#,*,6,/,.,=,.,#,#,#,.,#,#,=,#,=,#,#,#,=,#,#,6,#,8,*,.,=,16”
- **Nivel medio:**
  “12,*,.,=,36,#,#,#,15,#,#,*,#,/,#,#,#,*,.,-
  ,.,=,.,#,55,#,.,*,#,=,#,=,#,/,#,=,.,#,15,#,9,*,.,=,45,=,#,#,#,#,#,=,#,#,72,#,20,-,.,=,11,#,.,#,#,-
  ,#,+,#,#,#,*,56,/,.,=,.,#,#,#,.,#,#,=,#,=,#,#,#,=,#,#,12,#,16,*,.,=,32"
- **Nivel difícil:**
  "4,.,.,=,36,#,#,#,25,#,#,*,#,.,#,#,#,.,.,-
  ,.,=,.,#,15,#,.,*,#,=,#,=,#,.,#,=,.,#,18,#,6,*,.,=,30,=,#,#,#,#,#,=,#,#,56,#,9,-
  ,.,=,3,#,.,#,#,*,#,+,#,#,#,*,20,.,.,=,18,#,#,#,.,#,#,=,#,=,#,#,#,=,#,#,18,#,24,.,.,=,72"

Los valores numéricos de las cadenas anteriores representan celdas que ya se dan rellenas al comenzar el juego mientras que los valores “.” representan celdas que estarán vacías y que el usuario deberá rellenar para completar satisfactoriamente el juego. Las celdas con valor “#” son las que no se van a utilizar en el juego (celdas deshabilitadas).

Tarea 4. Inicializando el array del crucigrama

Crea un método auxiliar de nombre *start* dentro de la clase Crucigrama que ponga valores dentro de las celdas del array bidimensional que se ha inicializado dentro del constructor.

Guía para resolver la tarea 4

El valor de cada posición del array se debe tomar de la cadena que representa el tablero, que se debe ir recorriendo carácter a carácter:

- Si el valor en dicha cadena es numérico se vuelca el mismo número al array
- Si el valor en la cadena es un “.” se debe introducir el valor 0 (cero) en dicha posición del array.
- Si el valor en la cadena es un “#” se debe introducir el valor -1 (menos uno) en dicha posición del array.

Tarea 5. Creando la estructura del crucigrama

Crea un método auxiliar de nombre *paintMathword* dentro de la clase Crucigrama.

Este método será el encargado de crear en el documento HTML, a través de jQuery, los párrafos que representarán las celdas del crucigrama. Una vez creados los párrafos, este método inicializa la variable *init_time* de la clase Crucigrama al valor de la fecha actual.

Guía para resolver la tarea 5

Los párrafos se deben crear dentro del elemento main del documento HTML.

Según el contenido del array que representa el estado del crucigrama, los párrafos creados deberán tener el siguiente contenido:

- Si en el array hay un 0, el párrafo no tendrá contenido y se añadirá al mismo en manejador del evento click para que se pueda seleccionar con el ratón.
  - **NOTA:** Al hacer click en una celda se debe modificar el valor del atributo *data-state* al valor *clicked*.
- Si en el array hay un valor distinto de cero:
  - Si es un valor -1, el párrafo se pinta vacío (sin contenido) y se debe modificar el valor del atributo *data-state* a *empty*.
  - Si el valor es un número positivo, el párrafo se pinta con dicho número en su interior y se debe modificar el valor del atributo *data-state* a *blocked*.

En el documento crucigrama.html añade una etiqueta script fuera del body y antes del cierre de la etiqueta html e incluye una llamada al método *paintMathword*.

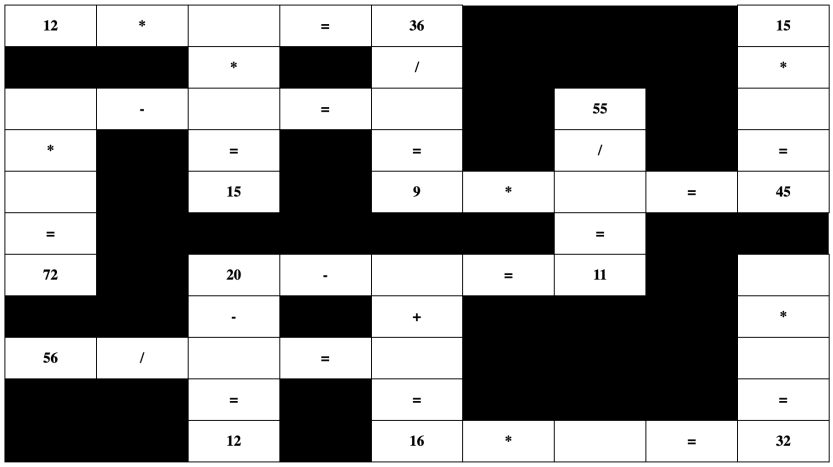
Consulta la información del objeto Date de JavaScript para saber como se crea un objeto de esta clase y como se utilizan dos objetos de tipo Date para calcular la diferencia entre dos instantes temporales.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date

Tarea 6. Pintando la estructura del crucigrama

Crea una estructura de cuadrícula que permita representar las celdas del crucigrama. Al tratarse de una estructura de 9 filas y 11 columnas, esta deberá estar dividida en 9 columnas iguales y 11 filas iguales.

El resultado final debe ser similar al que se observa en la siguiente imagen. A diferencia de lo sucedido con el sudoku en el laboratorio anterior, no es necesario que las celdas sean cuadradas.



Guía para resolver la tarea 6

Aplica GRID Layout para conseguir la disposición de las celdas del crucigrama que se observa en la imagen de ejemplo. Consulta las características del módulo [CSS-GRIDLAYOUT] a través de los siguientes enlaces:

- Referencia: https://www.w3.org/Style/CSS/current-work
- Más información: https://css-tricks.com/snippets/css/complete-guide-grid/ y https://jsfiddle.net/solisjaime/snwzmrpu/5/

Tarea 7. Creación de métodos de utilidad en la clase Crucigrama

Para que el juego funcione correctamente es necesario crear una serie de métodos de utilidad que realicen ciertas actividades: comprobar si el crucigrama ya está completo y calcular el tiempo invertido en completar el crucigrama.

Guía para resolver la tarea 7

Se deben crear los siguientes métodos:

- Método *check_win_condition*: este método comprueba si se ha completado el crucigrama, buscando elementos del array del tablero que estén al valor cero; si los encuentra, devuelve falso y en caso de no encontrarlos devuelve verdadero.
- Método *calculate_date_difference*: realiza una cuenta con los valores de las variables init_time y end_time para obtener el tiempo total invertido en resolver el crucigrama y lo devuelve como una cadena de texto.
    - **NOTA**: el formato de devolución de este método debe ser del tipo horas:minutos:segundos.

Tarea 8. Añadiendo el evento de teclado

Dentro de la etiqueta script creada en el documento crucigrama.html añade el código que permita el tratamiento del evento de teclado.

Guía para resolver la tarea 8

Consultar el evento keydown, que se produce cuando se pulsa una tecla

https://developer.mozilla.org/es/docs/Web/API/Document/keydown_event

Al tratarse de un crucigrama matemático, las únicas teclas que se deben permitir son las numéricas del 1 al 9 y las de los operadores aritméticos (+,-,*,/) por lo que el resto de teclas que pulse el usuario se pueden ignorar.

Al detectarse una pulsación de teclado se deben realizar las siguientes comprobaciones:

1. Debe estar seleccionada, con anterioridad a la pulsación de teclado, una celda del crucigrama como receptora de la tecla pulsada.
2. Si la tecla pulsada es un número o un operador aritmético y hay una celda seleccionada se debe llamar al método *introduceElement* de la clase Crucigrama, pasándole como parámetro a dicho método la tecla que se ha pulsado.

3. Si la tecla pulsada es un número o un operador aritmético pero no hay una celda del crucigrama seleccionada, se debe informar al usuario que debe seleccionar una celda antes de pulsar cualquier tecla.

Tarea 9. Introduciendo pulsaciones y comprobando su validez

Crea el método *introduceElement* en la clase Crucigrama.

Dicho método recibirá como parámetro el valor que se ha pulsado en el teclado. Este método debe comprobar si el valor pulsado está permitido (es un número o un operador aritmético) y si es válido para la casilla que está seleccionada.

Guía para resolver la tarea 9

Crea dos variables *expression_row* y *expression_col* e inicializa ambas al valor true.

Identifica la posición en el array del crucigrama que se corresponde con la celda que ha pulsado el usuario y pon el valor de dicha posición del array al valor recibido como parámetro del método.

A partir de la celda seleccionada actualmente se debe comprobar si la expresión afectada por la celda que se pretende rellenar se cumple tanto a nivel vertical como a nivel horizontal.

Partiendo de la casilla inmediatamente a la derecha de la seleccionada actualmente, es necesario buscar la siguiente casilla hacia la derecha que tenga el carácter = (igual).

1. Si la casilla inmediatamente a la derecha de la actual tiene el valor -1, no hay una expresión de tipo horizontal asociada a la celda actual ya que hay una celda deshabilitada.
2. En caso contrario, se debe ir avanzando celda a celda hacia la derecha hasta encontrar el primer carácter = (igual).

Cuando se haya encontrado el primer carácter igual, se deben obtener los siguientes valores:

- El primer operando de la expresión (3 posiciones a la izquierda del carácter igual) almacenado en una variable de nombre *first_number*
- El segundo operando de la expresión (1 posición a la izquierda del carácter igual) almacenado en una variable de nombre *second_number*
- El operador aritmético de la expresión (2 posiciones a la izquierda del carácter igual) almacenado en una variable de nombre *expression*
- El resultado de la expresión (1 posición a la derecha del carácter igual) almacenado en una variable de nombre *result*.

Si los valores numéricos las variables *first_number*, *second_number*, *expression* y *result* son distintos de cero, se debe realizar una evaluación de la expresión para conocer si el elemento introducido resuelve correctamente la ecuación.

- Utilizando el método *join* de JavaScript se debe crear una expresión matemática a partir las variables *first_number*, *expression* y *second_number*.
- Utilizando el método *eval* de JavaScript se debe evaluar la expresión matemática obtenida en el paso anterior y comprobar que el resultado de la evaluación sea igual al valor de la variable *result*.

  - En caso negativo (no coinciden el valor de la variable *result* y el resultado de la evaluación de la expresión), poner la variable *expession_row* al valor *false*.

Una vez comprobada la vertiente horizontal de la expresión, se debe comprobar la vertiente vertical de la misma. Partiendo de la casilla inmediatamente inferior de la seleccionada actualmente, es necesario buscar la siguiente casilla hacia abajo que tenga el carácter = (igual).

1. Si la casilla inmediatamente hacia abajo de la actual tiene el valor -1, no hay una expresión de tipo vertical asociada a la celda actual ya que hay una celda deshabilitada.
2. En caso contrario, se debe ir avanzando celda a celda hacia abajo hasta encontrar el primer carácter = (igual).

Cuando se haya encontrado el primer carácter igual, se deben obtener los siguientes valores:

- El primer operando de la expresión (3 posiciones hacia arriba del carácter igual) almacenado en una variable de nombre *first_number*
- El segundo operando de la expresión (1 posición hacia arriba del carácter igual) almacenado en una variable de nombre *second_number*
- El operador aritmético de la expresión (2 posiciones hacia arriba del carácter igual) almacenado en una variable de nombre *expression*
- El resultado de la expresión (1 posición hacia abajo del carácter igual) almacenado en una variable de nombre *result*.

Si los valores numéricos las variables *first_number*, *second_number*, *expression* y *result* son distintos de cero, se debe realizar una evaluación de la expresión para conocer si el elemento introducido resuelve correctamente la ecuación.

- Utilizando el método *join* de JavaScript se debe crear una expresión matemática a partir las variables *first_number*, *expression* y *second_number*.
  - Más información sobre el método join: https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array/join

- Utilizando el método *eval* de JavaScript se debe evaluar la expresión matemática obtenida en el paso anterior y comprobar que el resultado de la evaluación sea igual al valor de la variable *result*.
  - Más información sobre el método eval: https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/eval

En caso negativo (no coinciden el valor de la variable *result* y el resultado de la evaluación de la expresión), poner la variable *expession_col* al valor *false*.

Si finalizadas todas las comprobaciones verticales y horizontales, las variables *expression_col* y *expression_row* tienen el valor true (ambas variables), el elemento introducido en la casilla es correcto. En ese momento se debe:

1. Modificar el valor de dicha casilla (párrafo)
2. Modificar el valor del atributo *data-state* de dicha casilla al valor *correct*

En caso contrario, la casilla se quedará vacía y se debe escribir un 0 (cero) en la posición del array correspondiente a dicha casilla. Además, se debe proceder a devolver el valor del atributo *data-*

state de la casilla a su valor inicial y se debe mostrar una alerta al usuario diciendo que el elemento introducido no es correcto para la casilla seleccionada.

El método termina con una llamada al método *check_win_condition* de la propia clase. Si dicho método devuelve verdadero, se deben realizar las siguientes acciones en este orden:

1. Inicializar la variable *init_time* al valor de la fecha actual
2. Invocar al método *calculate_date_difference* de la propia clase para conocer el tiempo que tardó el usuario en realizar el crucigrama.
3. Mostrar una alerta al usuario donde se diga que ha completado el crucigrama en el tiempo devuelto por el método *calculate_date_difference*.

Consulta la información relativa a las expresiones regulares de JavaScript en el siguiente enlace para conocer cómo se pueden utilizar a la hora de identificar si las teclas pulsadas son números u operadores matemáticos (función test).

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/RegExp

Resultado del ejercicio 4

Una vez completado el ejercicio deberían haberse añadido los siguientes ficheros al proyecto del Escritorio Virtual:

- Fichero crucigrama.js en el directorio js del proyecto
- Fichero crucigrama.html en el directorio raíz del proyecto
- Fichero crucigrama.css en el directorio estilo del proyecto

Además, se debe haber modificado la sección de menú del fichero juegos.html para incluir el enlace de acceso al juego del crucigrama matemático.

Recuerda

Se requiere el uso correcto de los elementos HTML5 establecidos en los ejercicios y se valorará positivamente el uso correcto y adecuado al contexto y funcionalidad, de elementos adicionales HTML5.

Se requiere el uso correcto de las propiedades de los módulos CSS establecidos en los ejercicios y se valorará positivamente el uso correcto y adecuado al contexto, de propiedades y módulos adicionales CSS.

Solamente se permite el paradigma de orientación de objetos y todo debe estar organizado con clases y objetos. Además, solamente se permite el uso de la biblioteca jQuery para realizar las llamadas a los servicios externos y la creación de contenido en el html, quedando prohibido el uso de cualquier otra biblioteca.

IMPORTANTE: Todas las llamadas que se realicen a la biblioteca jQuery deben estar encapsuladas en los métodos pertenecientes a las clases que se utilicen para resolver los ejercicios.

Las siguientes condiciones son de obligado cumplimiento en todo el proyecto:

- **TODOS** los documentos HTML que componen el proyecto deben ser HTML5 válidos y sin advertencias utilizando el validador de lenguajes de marcado del W3C.
    - Recuerda que el contenido de los documentos HTML puede cambiar después de la ejecución del código ECMAScript. En ese caso, se debe comprobar la validez del código HTML en todos los diferentes estados por los que pase el documento.
- No está permitido el uso indiscriminado de bloques anónimos (**div**) en los documentos HTML5, se debe priorizar el uso de contenedores semánticos HTML5.
- Se deben utilizar selectores específicos, el uso de selectores id y class deberá estar justificado a través de comentarios en las hojas de estilo que expliquen la necesidad de su utilización y para que se utilizan.
- **TODAS** las reglas de todas las hojas de estilo deben estar precedidas por un comentario donde se indique la especificidad del (o los) selectores de la regla.
- **TODAS** las hojas de estilo que se utilizan en el sitio web deben ser validas utilizando el validador CSS del W3C
- **TODAS** las hojas de estilo deben tener 0 advertencias.
    - Recuerda seleccionar en “Más opciones” el informe de “Todas las advertencias” dentro de las opciones del validador CSS del W3C.
    - Excepcionalmente se permite las advertencias referidas a la verificación de los colores (color y background-color). **OBLIGATORIAMENTE** se debe indicar mediante un comentario en la regla de la hoja de estilo afectada la herencia de colores garantizando que la advertencia ha sido comprobada, verificada y garantizando que no provoca efectos laterales no deseados.
    - Excepcionalmente se permiten las advertencias referidas a la redefinición de propiedades derivadas del uso de @media-queries. **OBLIGATORIAMENTE** se debe indicar mediante un comentario en las reglas de la hoja de estilo afectada que propiedades se están redefiniendo.
- Se debe garantizar la adaptabilidad y realizar su verificación para todos los documentos que componen el proyecto.

    - Se deben utilizar medidas relativas en las hojas de estilo.
- Se debe garantizar la accesibilidad del proyecto mediante los test de las herramientas de accesibilidad para el nivel AAA de las WCAG 2.0 con 0 errores de modo automático en todos los documentos que lo componen.

El no cumplimiento de las características anteriores derivará en la invalidación del proyecto.