



vocabulum



Vocabulum es un proyecto final para el curso de JavaScript de CoderHouse.

Se basa en ser una herramienta de apoyo en el estudio de una lengua extranjera (italiano en este caso pero puede ser (y será) ampliado a otros idiomas).

El main del sitio está dividido en 4 secciones principales.

Primera sección

La primera de ellas tiene un formulario, los listados junto con el botón para la restauración de la información y el botón de muestra (que está a los efectos de que el profesor no tenga que ingresar muchas palabras en el formulario para poder probar las funcionalidades del trabajo final).

Formulario

El formulario que tiene como objetivo que el usuario ingrese las palabras o frases nuevas que va aprendiendo, se carga en `arrayPalabra` y `arrayCategoria` con la función `cargarDatos`, esta función contiene la función `ordenar` que ordena los significantes en orden alfabético y las categorías en los arreglos correspondientes.

Al clicar el botón agregar hace una validación de datos (función `validarFormulario`) para que no se ingrese un objeto vacío al array. El significativo, la traducción y la categoría son obligatorias, la definición y la clase no lo son. En caso de que la validación sea negativa aparecerán los mensajes de error que desaparecerán una vez que cargue los datos correctamente y seleccione agregar.

Listado o planillas

Por otro lado en el `arrayCategoria` se irán cargando solo las categorías que ya se usaron pudiendo quedar repetidas (puede servir en un futuro para filtrar cuantas veces se usó dicha categoría aun si la palabra fue eliminada dado que no se elimina la categoría del array que queda almacenado en el almacenamiento local). De este arreglo se crea mediante la función `crearPlanillaCategoria` el listado de las categorías usadas depurado, es decir, sin que aparezcan repetidas. El usuario al pasar el ratón sobre el texto “ver categorías ya usadas” podrá visualizarlas. La visualización se realiza con una aparición que toma 1 segundo luego 5 segundos de visualización estática y una desaparición en 1 segundo con el uso de JQuery y el método `fadeIn`, `delay` y `fadeOut`. Si todavía no cargó ninguna categoría verá un mensaje que dice “Aún no hay ninguna.”

- El listado no aparecerá hasta que el usuario cargue una palabra o restaure aquello que ya haya cargado con anterioridad que queda en el almacenamiento local. Al hacer alguna de estas dos acciones, se ejecuta la función `cargarDatos` o `restaurarPlanilla` y que mediante JQuery y el método `show` muestra el menú en 2 segundos que se encuentra `<header>`.
Contiene tres botones que al clicarlos ejecutan la función creadora de la planilla

correspondiente.

Esta creación parte del `arrayPalabra`, de manera que al eliminar una palabra existente se elimina de las tres planillas dado que se elimina de dicho array.

- Planilla completa: contiene el listado que incluye todos los elementos del `arrayPalabra` (significante, clase, traducción, definición, categoría). Se crea mediante la función `crearPlanillaInicio` (Jquery FadeIn 3 segundos).
- Desde el italiano (lengua de partida): incluye los elementos significante, clase, traducción y categoría en dicho orden. Se crea mediante la función `crearPlanillaLenguaPartida` (Jquery FadeIn 3 segundos).
- Desde el español (lengua meta): incluye los elementos traducción, clase, significante y categoría en dicho orden. Se crea mediante la función `crearPlanillaLenguaMeta` (Jquery FadeIn 3 segundos).

Que la información cargada sea desde el formulario o desde la restauración no aparezca automáticamente en la pantalla es porque el objetivo es que el usuario se concentre en la carga de la información y si desea verla tiene que seleccionar la planilla.

Para eliminar una palabra al clicar el botón de eliminar se ejecuta la función `eliminarPalabra` que valida el `data-id` del botón, este atributo toma como referencia significante, clase y traducción, dado que puede haber una mismo significante que tenga traducciones o clases diferentes.

Restauración de la información.

Al cargar los datos del formulario (función `cargarDatos`) se toman el `arrayPalabra` y el `arrayCategoría` y se convierten en archivos JSON para ser almacenados en el almacenamiento local bajo las claves `listado` y `categorías` respectivamente.

Al clicar el botón “restaurar última planilla” se ejecuta la función `restaurarPlanilla`, la cual carga en ambos array mencionados la información que estaba en el almacenamiento local, por lo que se vuelve parte del `arrayPalabra` y `arrayCategoría`.

Luego de restaurar la planilla es necesario hacer clic en la planilla que se desea ver.

Muestra de la información.

Al clicar el botón “muestra” se ejecuta la función `cargarMuestra`, ésta llama el archivo JSON con el método `getJSON` usando Jquery con el mismo nombre y mediante un `forEach` se agregan al `arrayPalabra` y al `arrayCategoría` los elementos del archivo. Por lo que también quedarán en el almacenamiento local y servirán para probar la restauración de la información.

Segunda sección

Es un `<section>` que contiene un `<article>` con la explicación de la ejercitación que aparece a la derecha en los dos `<div>`.

Ejercitación

En los `<div>` aparecen un botón en cada uno indicando “Mostrar palabra” al cliquearlos aparece una palabra aleatoria tomada del `arrayPalabra` mediante la función `mostrarPalabraAleatoriaLP` (lengua de partida, “Desde el italiano”) y `mostrarPalabraAleatoriaLM`, (lengua de meta, “Desde el español”).

Primeramente se almacenan los significantes y las traducciones del arreglo en dos nuevos arreglos `arraySignificante` y `arrayTraduccion` con el uso del método `map`. luego con el método se busca una posición aleatoria del `arrayPalabra` en base a su longitud mediante el objeto `Math` y sus métodos `floor` y `random` dicha posición se almacena en `aleatorio`.

En el caso de la lengua de partida (Italiano), se va a mostrar primero lo que en el arreglo llamamos de significativo y luego el de traducción (Para el usuario el significativo siempre será el primero que aparezca y la traducción el segundo) y viceversa para el caso de lengua meta (español).

Con el uso de JQuery se muestran dichos elementos con animación, los contenedores `contenedorSignificanteLP` y `contenedorTraduccionLM` creados por las funciones aparecen con una transición de 1 segundo y con un retraso de 8 segundos aparecen con una transición de 1 segundo `contenedorTraduccionLP` y `contenedorSignificanteLM`.

En el plazo de los 8 segundos el usuario debería haber adivinado, recordado o pensado la traducción correspondiente.

Las funciones `mostrarPalabraAleatoriaXX` al mismo tiempo muestran dos nuevos botones mediante el uso de JQuery y esconden el botón visible. El primer botón indica “Mostrar nueva palabra” el cual vuelve a ejecutar dicha función y el segundo botón “Terminar repaso” ejecuta la función `terminarRepasoLM` o `terminarRepasoLP`, deja el contenedor con las mismas características que poseía antes de empezar, esconde los botones “Mostrar nueva palabra” y “Terminar repaso” y visibiliza “Mostrar palabra”.

Tercera sección

Es un `<section>` que contiene un `<article>` con la explicación de la ejercitación que aparece a la derecha en los dos `<div>`.

Ejercitación

El usuario debe hacer clic en el botón que dice “Mostrar tarjetas”, así se ejecuta la función `crearArrayCategoriaAleatoria`, ésta mapea el `arrayPalabra` (y no el `arrayCategoria` dado que éste puede contener categorías que no están en uso) en busca de las categorías, guardándolo en `categorizacion` luego elige una posición aleatoria en base al largo (`length`) de `categorizacion` luego busca la categoría que está en esa posición y la guarda en `categoriaSorteada`, mediante un `forEach` se buscan los elementos bajo su categoría dentro del `arrayPalabra` y si coinciden con la `categoriaSorteada`, se agregan los significantes de estos elementos elegidos en `arrayElegidos`, aquellos que no coinciden se agregan en `arrayDesubicado`. Por último, de este arreglo se busca una posición aleatoria (`posicionAleatoria`) basada en su

longitud (`length`) y se agrega al `arrayElegidos` el elemento del `arrayDesubicado` que está en la posición. Se ordena alfabéticamente el arreglo y se llama la función `crearGrilla`.

`crearGrilla` recorre con un `forEach` del `arrayElegidos` y mediante condicional `if` los aquellos elementos que están incluidos en el `arrayDesubicado` se insertan en el HTML con la clase `tarjetaCorrecto` dado que el objetivo es que seleccione el que no tiene relación, mientras que los que no pasan esa condición (`else`) se insertan en el HTML con la clase `tarjetaIncorrecto`. El array es ordenado alfabéticamente mediante el método `sort` para que no quede siempre en primer lugar la tarjeta correcta.

Las tarjetas son `<button>`. Basándose en las clases `tarjetaCorrecto` y `tarjetaIncorrecto` se cargan en las variables `bienSeleccionado` y `malSeleccionado` respectivamente y mediante un bucle de `for` a cada `<button>` se le asigna la clase `verde` para los correctos y para `rojo` los incorrectos, estas modifican el color de fondo de las tarjetas mediante CSS.

Esta sección posee la misma lógica respecto al uso de los botones “Mostrar tarjetas”, “Renovar tarjetas” y “Terminar repaso” con el uso de JQuery dentro de las funciones `crearGrilla`, `renovarTarjetas` y `terminarRepaso`.

Cuarta sección

Es un `<section>` que contiene un `<article>` con botones que llaman a los archivos JSON almacenados localmente mediante `getJSON` y que aparecerán a la derecha en un `<div>` creando mediante un `forEach` una planilla. A diferencia de las planillas de la primera selección, ésta no posee la opción de eliminar dado que no corresponde porque es vocabulario nuevo para que el usuario aprenda. Por otro lado se pensó pero no se implementó la opción de que el usuario pueda desde esa misma planilla cargar los elementos en el `arrayPalabra` dado que si el usuario los ingresa manualmente en la primera sección se genera mayor ejercitación.

Al seleccionar los botones del `<section>` cada uno ejecuta una función diferente almacenadas como constantes (`JSONElClima`, `JSONElDormitorio`, `JSONLaCocina`, `JSONLaCasa`, `JSONLaSala`, `JSONLasBebidas`, `JSONLosAnimales`, `JSONLosColores`, `JSONLosDeportes`). Éstas a parte de crear la lista insertan un `<div>` con un `</button>` que tiene como función limpiar el contenedor y volverlo al estado inicial mediante la función `limpiarVocImportado`.