



Programación Avanzada
Grado en Ingeniería Informática en Sistemas de Información - Curso 2017/2018
Prueba de evaluación 3: JavaScript y jQuery

Nombre: _____

Equipo: _____

Importante: El resultado de esta prueba consistirá en 4 archivos (2 páginas HTML y 2 archivos JavaScript) y 2 carpetas (la carpeta *img* conteniendo las fotografías y la carpeta *js* con las librerías), lo cual será todo comprimido en formato ZIP. El nombre del fichero tendrá un formato específico dictado por el nombre de cada alumno. Por ejemplo, para un alumno llamado "José María Núñez Pérez" el fichero se nombrará como NunyezPerezJM.zip. Obsérvese que las tildes son ignoradas y las eñes sustituidas. Las rutas de los ficheros empleados serán relativas, a fin de que las resoluciones a los ejercicios y problemas puedan ser examinadas en cualquier equipo. Cualquier entrega que no cumpla las reglas de nombrado, el formato de compresión del archivo o el contenido de los archivos del mismo, será penalizada con 2 puntos sobre 10 por cada incumplimiento. Pasado el límite de entrega se aceptará el envío del trabajo, con una penalización de 2 puntos sobre 10 de la calificación por cada minuto o fracción de retraso a partir del tercer minuto.

Problema Propuesto

IMPORTANTE: Recuerde que el uso del atributo `innerHTML` no se tolera y su uso invalidará el ejercicio entregado.

Partiendo de la página `calificaciones.html`, suministrada como material adicional, e incluyendo todo el código JavaScript en un archivo separado a la página, dote a ésta de las siguientes funcionalidades interactivas:

1. **[2 puntos]** Observe en la página que, en la tabla HTML que muestra las calificaciones, dentro de la primera fila se encuentran las mismas como campos de tipo numérico, y en la segunda fila de la tabla se encuentran celdas vacías con identificadores de la forma `eX_annotaciones` (donde X es un número índice para cada calificación, comenzando en 1). Será dentro de estas celdas donde se mostrarán ciertas anotaciones asociadas a cada calificación (tenga en cuenta que puede haber más de una anotación a mostrar). Nótese que el identificador de cada celda de la fila inferior es similar al de la celda de la fila superior correspondiente añadiendo el sufijo `"_annotaciones"`.

Implemente una función `anotar(annotaciones, id)` que recibe como primer argumento un vector con las cadenas de texto que representan las anotaciones y como segundo argumento el identificador del campo al que se refieren las anotaciones. Esta función localizará la celda vacía donde se han de mostrar las anotaciones relativas a dicho campo **(0.4 puntos)** e imprimirá estas anotaciones **(0.5 puntos)**. Tenga en cuenta que:

1. El contenido de la celda donde colocar las anotaciones ha de limpiarse antes de colocar nuevos contenidos (para evitar que se acumule el texto en sucesivas llamadas a la función) **(0.5 puntos)**.
2. Todas las anotaciones se separarán mediante una lista con viñetas **(0.5 puntos)**.
3. En caso de recibir un vector vacío, simplemente se limpiarán los errores mostrados **(0.1 puntos)**.

Implemente esta funcionalidad usando **JavaScript**.

2. **[1.5 puntos]** Además de las anotaciones, también se marcarán las propias calificaciones de manera especial empleando un estilo CSS. Para ello, implemente una función `marcarElementoAnotado(annotado, id)`. El primer parámetro es un valor booleano que indica si se ha de marcar el nodo cuyo identificador es suministrado como segundo parámetro. Esta función modificará el estilo CSS de los campos de las calificaciones que tengan al menos una anotación, poniendo su texto en negrita **(0.5 puntos)**. Las calificaciones que no tienen anotaciones deberán tener su texto sin negrita **(0.5 puntos)**. Implemente la funcionalidad anterior usando **jQuery**. Modifique la función `anotar()` para que llame a `marcarElementoAnotado()` de forma conveniente, y poder así marcar (poner en negrita) las calificaciones en caso de que se muestren anotaciones, o desmarcarlas (quitar la negrita) en caso contrario **(0.5 puntos)**.
3. **[1 punto]** Cada vez que se modifique cualquiera de las calificaciones, se deberán comprobar estas condiciones:
 1. Si la calificación no está entre 0 y 10 o está vacía, se generará una anotación con el texto "Inválida".
 2. Si la calificación está en el intervalo [0,5), se generará la anotación "Suspendo"; si está en [5,7), "Aprobado"; si está en [7,9), "Notable"; y si está en [9,10], "Sobresaliente".

En función de las condiciones que se cumplan, se deberán anotar, empleando la función desarrollada en el punto 1, los mensajes correspondientes, según convenga **(0.5 puntos)**. Tenga en cuenta que si, tras una modificación, alguna de las condiciones no se cumple, su anotación correspondiente debe desaparecer **(0.25 puntos)**. Si no se satisficiese ninguna de las condiciones, la celda de anotaciones aparecería completamente vacía **(0.25 puntos)**.



4. **[0.75 puntos]** Desarrolle una nueva funcionalidad de forma que si en el desplegable junto al texto “Crear nuevas calificaciones” se selecciona la opción “Personalizado”, se mostrarán el campo y el botón que permanecían ocultos (busque en el código la capa `div longitudNuevasCalificaciones`) usando una transición lenta (**0.5 puntos**). Al elegir cualquier otra opción distinta de “Personalizado” en el desplegable, haga que se oculten de nuevo con una transición lenta (**0.25 puntos**). Implemente esta funcionalidad usando **jQuery**.
5. **[1.25 puntos]** Impida que en el campo `longitud` se puedan escribir caracteres distintos a números (**0.5 puntos**) y tampoco números superiores a 100 (**0.5 puntos**). Si se detectara que no fuera un número (o que éste supera 100), se borraría sólo el último carácter escrito (**0.25 puntos**). El rango unicode para los números es [48,57].
6. **[1.75 puntos]** Tanto si el usuario pulsa el botón “Crear”, como si elige una opción numérica del desplegable (5, 10 ó 30), se deberá crear una nueva lista de calificaciones en la tabla HTML con tantos elementos como el usuario hubiera especificado o bien en el desplegable, o bien en el campo `longitud` (**0.7 puntos**), eliminando el vector ya existente (**0.15 puntos**). Las calificaciones que se crean estarán vacías por defecto (**0.1 puntos**). Debe respetar la notación de los identificadores de las celdas de la tabla (`eX` y `eX_annotaciones`, donde X es un número índice de cada calificación, comenzando en 1) (**0.3 puntos**). Además, tras crear las nuevas calificaciones, se deberán comprobar automáticamente las condiciones del punto 3 y realizar las anotaciones que correspondan, reutilizando al máximo las funciones descritas en los puntos anteriores (**0.5 puntos**).
7. **[0.75 puntos]** Cree un *plugin* de jQuery que haga que, cuando se abandone el foco de uno de los campos que contienen las calificaciones, éstas cambien su color de fondo en función del valor que tengan. Si el valor es menor que 5 (suspense), el color será rojo. Si el valor es mayor o igual que 5 (aprobado), pero no es un 10, entonces color verde. Si la calificación es un 10 (sobresaliente/MH), será de color dorado (color “gold”).

Captura de ejemplo de la página mostrando las diversas anotaciones

Calificaciones - PA - Examen JS, jQuery, Plugins

4	5	7	9	10	-2
• Suspenso	• Aprobado	• Notable	• Sobresaliente	• Sobresaliente	• Inválida

Crear nuevas calificaciones: Personalizado

6

Crear

Partiendo de la página `galeria.html` suministrada como material adicional, e incluyendo todo el código JavaScript en un archivo separado a la página, dote a ésta de las siguientes funcionalidades interactivas:

8. **[0.5 puntos]** Usando el *plugin* `animateScroll` haga que, al hacer doble clic en cualquier imagen, la página se desplace a la siguiente imagen implementando en cada una de las 4 imágenes un efecto distinto (**0.3 puntos**). Al hacer doble clic en la última imagen, la página deberá desplazarse a la primera imagen (**0.2 puntos**). En el material adicional encontrará el fichero con el código del *plugin* “`animateScroll`” y una breve descripción de su uso.
9. **[0.5 puntos]** Escoja dos de las imágenes de la página, y haciendo uso del *plugin* `oriDomi`, haga que las imágenes se plieguen en 2 y 4 paneles verticales respectivamente (**0.1 puntos**). La duración del plegado de la primera será de 2 segundos mientras que la segunda será de 6.5 segundos (**0.1 puntos**). Tras el plegado de una de las imágenes, se debe mostrar un cuadro emergente con un mensaje indicativo (**0.1 puntos**). Una de las imágenes se mostrará usando un efecto tipo `curl` (con ángulo 10) (**0.1 puntos**) y la otra con un efecto de fractura (con ángulo -20) (**0.1 puntos**).

Material

Páginas HTML base para crear las páginas web interactivas, librería jQuery, *plugin* `oriDomi` y `animateScroll`, breve descripción del uso de los *plugins* y una carpeta con imágenes. **Los archivos HTML NO DEBEN SER MODIFICADOS DE NINGÚN MODO para solucionar las funcionalidades requeridas.**



Datos de la Práctica

Autor del documento: Gualberto Asencio Cortés (Enero 2018).

Revisiones del documento

1. Gualberto Asencio Cortés (Enero 2018): Consignadas puntuaciones para todos los aspectos evaluables de los ejercicios propuestos. Correcciones leves en las indicaciones del primer texto del examen para aclarar su significado. Añadida figura con captura de la página mostrando elementos el aspecto de varios de los elementos que se solicitan.
2. Carlos D. Barranco González (Enero 2018): Pequeñas correcciones en formato y texto.