

Desarrollo de Aplicaciones Telemáticas, examen práctico, 21/05/2024

Grado en Ingeniería en Tecnologías de la Telecomunicación

Grado en Ingeniería en Sistemas de la Telecomunicación

Universidad Rey Juan Carlos

Instrucciones

1. Emplea lo visto en clase, y solo lo visto en clase (que puedes consultar en las transparencias). No será válido ni JavaScript más antiguo ni más avanzado.
2. En los ficheros `~/figuras.html` y `~/figuras.js` encontrarás tu práctica 4.8. Tendrás que modificar el html para indicar que el script ya no es `tablaRandom.js` sino `figuras.js`. Encontrarás tus imágenes de la práctica 4 en `~/images`. Si tu código está bien hecho, esto no necesitará ninguna adaptación.

Ejercicio 1. 3.5 puntos

Hemos comprado diversas cantidades de un activo, que tiene precio variable. P.e. 500 unidades a 65 EUR, 300 unidades a 60 EUR y 200 unidades a 62 EUR. Queremos calcular el precio medio de compra. Para ese caso, sería

$$(500/1000) * 65 + (300/1000) * 60 + (200/1000) * 62 = 62.90$$

Escribe un programa en JavaScript en el fichero `~/precios.js` que tenga una función llamada `precioMedio` que realice este cálculo.

1. Tendrá un único parámetro de entrada: una lista cuyos elementos serán una lista, donde el primer subelemento será el número de unidades y el segundo subelemento, el precio.

Ejemplo:

```
[ [500,65], [300, 60], [200, 62] ]
```

2. Devolverá un número, el resultado del cálculo. Con todos los decimales obtenidos, sin truncar ni redondear.
3. Si la lista está vacía, si la función no recibe exactamente 1 parámetro o si este parámetro no es exactamente como aquí se indica, la función elevará las excepciones que consideres oportuno.
4. Naturalmente, la función debe funcionar para una lista de cualquier tamaño: 3 elementos como en este caso, 1 elemento o 3000 elementos.
5. La función `precioMedio` no debe tener *efectos laterales*, esto es, no puede escribir nada en la salida estándar ¹, solo puede usar variables locales, solo puede trabajar con el parámetro que recibe y no puede trabajar con ningún otro valor.
6. Incluye en fichero `precios.js` varias llamadas a la función `precioMedio` para probarla. Estas llamadas sí pueden (y deben) escribir en la salida estándar.

Ejercicio 2. 6.5 puntos

Modifica los ficheros `~/figuras.html` y `~/figuras.js` según esta especificación:

1. Hasta ahora, tu programa o bien añadía una figura en modo texto o bien añadía una figura en modo gráfico. Una figura en modo texto siempre se quedaba en modo texto, una figura en modo gráfico siempre se quedaba en modo gráfico.

Pero ahora:

¹excepto alguna traza, pero quítala luego.

- En modo gráfico, todas las figuras se verán en modo gráfico. Incluyendo las que previamente fueron lanzadas en modo texto.
 - En modo texto, todas las figuras se verán en modo texto, incluyendo las que previamente fueron lanzadas en modo gráfico.
2. Tu programa siempre trabajará en modo *sin repetición*. Borra todo lo relativo al modo *con repetición*.
 3. En tu programa se tiene que ir viendo el histórico de todas las figuras aparecidas.
 - Cuando sale una figura nueva no puede borrar la anterior.
 - Al cambiar el modo gráfico/texto, no se pueden borrar las figuras.
 - Al pulsar *nuevo mazo*, las figuras sí se pueden borrar. O no. Como prefieras.
 4. Cuando tu programa esté en *modo gráfico*, el botón de cambio de modo debe contener el texto *modo texto*, porque al pulsarlo este será el nuevo modo.
 5. Cuando tu programa esté en *modo texto*, el botón de cambio de modo debe contener el texto *modo gráfico*, porque al pulsarlo este será el nuevo modo.

Sugerencia

- Cuando añadas una figura, añade simultáneamente el texto y la imagen. Pero con clases distintas. Luego, al cambiar de modo, haz que unas u otras sean visibles o invisibles, según corresponda.