

Parte práctica (¡comentar código!!!)

- 1) [3 puntos]: Implementar un sistema de elección anidado
 - En el primer nivel deberán aparecer 3 opciones con `radio`: “elija animal”, “perros”, “gatos”.
 - El segundo nivel dependerá de la opción en el primer nivel y se mostrarán `checkbox` con los siguientes textos (se usará para el campo `value` las 3 primeras letras del texto)
 - “elija animal”: no aparecerá nada
 - “perros”: chiguagua, pastor alemán, mastín
 - “gatos”: persa, angora, siamés
 - Habrá un espacio para mostrar texto (un `div`, `span`, etc.) que estará vacío cuando la primera opción sea “elija animal”, y que si se elige un animal, irá mostrando el texto de los animales seleccionados al marcar o desmarcar los `checkbox`. Cada vez que se cambie de opción en el primer nivel, este espacio se borrará.
 - Los nodos en el segundo nivel (`checkbox`, `label`, etc.) deberán ser creados/destruidos usando funciones de DOM (esto no es necesario para los atributos, eventos, etc. o para mostrar los mensajes indicando qué animales se han elegido).

- 2) [3 puntos] a) Implemente una función llamada `json2sessionStorage()` que copie/mueva datos de un `json` al `sessionStorage`. Esta función acepta un parámetro opcional, si es `false` (por defecto), copia las cookies, si es `true`, las mueve.
b) Implemente la función inversa `sessionStorage2json()`, con el mismo comportamiento descrito anteriormente.
Para las pruebas y la corrección, escriba algunos datos de prueba y compruebe que realmente las funciones operan correctamente

- 3) [4 puntos] Implemente un lanzador de dados virtual. En la web aparecerá un `input` tipo `text`, un botón al lado que indique “Lanzar”, otro “Parar” (deshabilitado por defecto) y un espacio en el que se irán mostrando los mensajes (un `div`, un `span`, etc.). Al pulsar el botón “Lanzar”, sucederá lo siguiente:
 - Se comprobará usando `javascript` que el `input` contenga un número natural entre 0 y 20, ambos incluidos, usando para ello una expresión regular (no usar atributos HTML como `pattern`, `required`, etc.). Si el `input` está vacío o no contiene un número válido, se informará al usuario (por ejemplo con un `alert`) y no se hará nada más.
 - Si el `input` tiene un valor válido, por ejemplo supongamos que es `N`:
 - i. se bloqueará el `input` para que el usuario no pueda modificar el valor
 - ii. se deshabilitará el botón “Lanzar” y se habilitará el botón “Parar”
 - iii. Si el valor es 0, ir a paso “x”
 - iv. en el espacio para emitir mensajes se escribirá “Lanzando...”
 - v. se decrementará el valor de `N` en una unidad (actualizar el valor del `input`)
 - vi. se esperará 3 segundos y se generará un número aleatorio `X` entre 1 y 6
 - vii. se cambiará el mensaje a “El valor obtenido al lanzar en el dado es `X`, quedan `N` tiradas”
 - viii. Se repetirán los pasos iii a viii mientras `N > 0` cada 10 segundos (usar función de tiempo*)
 - ix. Cuando el valor sea `N=0`, se parará la función de tiempo que realiza los lanzamientos, se escribirá el mensaje “Ya no hay más tiradas” y se hará lo descrito en el siguiente paso
 - x. Si durante cualquier momento mientras se estén tirando los dados se pulsa el botón “Parar” (o cuando `N=0`), el proceso se detendrá, el botón “Lanzar” quedará habilitado, el botón “Parar” deshabilitado, se permitirá que el usuario modifique el `input` y se borrará el mensaje de información.