Ejercicios Parte I

1. PUM

Haga una página que genere el juego del PUM hasta 100. Es decir, que escriba los números separados por coma y cada vez que encuentre un número terminado en 7 o múltiplo de 7 escriba PUM y cambie de renglón.

2. Salarios

Desarrolla una aplicación que permita calcular los salarios mensuales de los trabajadores de una empresa a partir de los siguientes datos:

a. Número de horas trabajadas.

b. Turno realizado: Mañanas (m), Tardes (t) o Noches (n).

Para el cálculo del salario bruto se tendrá en cuenta que el turno de mañana se paga a 15€/hora, el de tarde se paga a 17 €/hora y el turno de noche se paga 20 €/hora.

Para el cálculo del salario neto se realizan determinados descuentos destinados al pago de impuestos de la siguiente forma:

-Si el salario bruto es menor de 600 € el descuento es del 8%.

-Si el salario bruto está entre 600 € y 1000 € el descuento es del 10%.

-Si el salario bruto es mayor de 10000 € el descuento es del 12%.

Se desea calcular el salario neto de cada trabajador. Para ello la aplicación irá pidiendo

uno a uno los trabajadores hasta que el usuario indique lo contrario. Para cada trabajador se mostrará su salario neto.

Antes de finalizar la aplicación mostrará el importe total de salarios abonados.

El script se escribirá usando tantas funciones como sea posible con el fin de poder

reutilizar la máxima cantidad de código en un futuro.

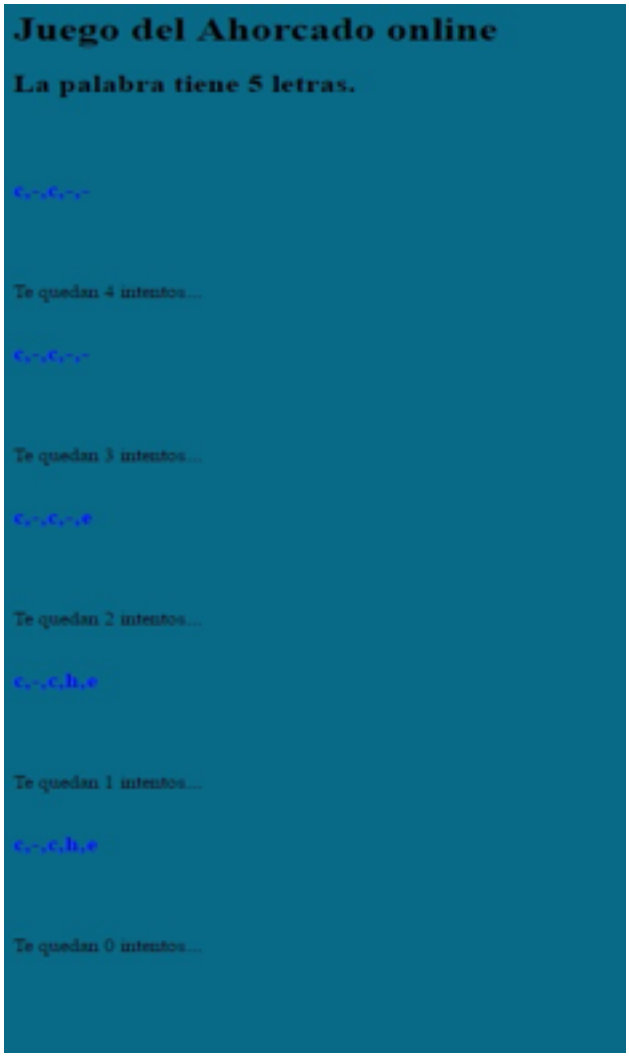
---

3. Juego del ahorcado.

Crea una web, usando JavaScript, que permita jugar al juego del ahorcado. Para desarrollar esta web sigue las siguientes reglas:

- Se debe pedir al usuario que introduzca la palabra a adivinar por teclado.
- Si la palabra a adivinar tiene mayúsculas se deben convertir a minúsculas.
- El número de intentos que tiene el jugador para adivinar la palabra es la misma que el número de letras que tiene.
- Se debe informar al jugador el número de letras que tiene la palabra a adivinar.
- Se debe comprobar que la letra que introduce el jugador no se ha probado ya antes. En caso de que sí se haya puesto antes, se informará al jugador y no influirá en el número de intentos.
- Una vez usados todos los intentos, el script debe indicar si se ha ganado o perdido y cuál era la palabra que se debía adivinar.
- En la web debe aparecer información sobre cada uno de los intentos: número de intentos que quedan, las letras que se han adivinado y su posición en la palabra a adivinar.

Ejemplo de resultado de ejecución:



4. Prototype

El objetivo de este ejercicio es que aprendan a utilizar la propiedad *prototype* y añadan un método al objeto String que permita separar caracteres recibiendo un argumento que actúe de separador.

Por ejemplo: si sobre un objeto string:

```
var cadena. = “prueba”;

console.log(cadena.separarCaracteres(“*”);
```

Mostraremos por la consola p*r*u*e*b*a

5. Math

Partiendo del código del ejemplo **ejercicioCanvas.html.**

El objetivo de esta actividad es reutilizar un código JavaScript codificado por otros y que puede aprovechar y adaptar para sus proyectos. Asimismo, otro objetivo es utilizar el objeto Math, objeto que normalmente no se utiliza en proyectos web.

La etiqueta `<canvas>`de HTML se puede utilizar para dibujar gráficos (normalmente utilizando JavaScript como lenguaje de script). En este ejercicio dibujaréis funciones trigonométricas sobre el `canvas`, y de esta manera demostraréis las posibilidades del objeto `Math`.

Utilizará el tag `<canvas>`como sábana para dibujar el gráfico.

Lo que la actividad propone es representar funciones trigonométricas sobre el `canvas`. Concretamente, pintando una encima de la otra las funciones trigonométricas *sen (x)* y *cuerpo (3x)* , la primera de color azul y la segunda de color rojo.

Extrae el código a un fichero .js independiente.

Añade otro canvas en el que pintes los ejes y muestres distintas funciones del objeto Math en distintos colores.