Ejercicios de Javascript. Bloque III - Uso de bucles.

1. **Bucles WHILE / DO-WHILE (JS 53).**

Realizar un programa que pida al usuario dos números y presente los números del primer número al segundo que introdujo el usuario.

2. **Bucles WHILE / DO-WHILE (JS 54).**

Realizar un programa que imprima por pantalla tantos asteriscos como diga el usuario. Al ejecutarse debe preguntar "Cuantos asteriscos desea imprimir?", leer el número que introduce el usuario e imprimir los asteriscos en HTML.

3. **Mostrar calificación WHILE / DO-WHILE con IF (JS 55).**

Escribir un programa que pida por teclado una nota de evaluación. La nota podrá ser decimal, no inferior a 0 ni superior a 10, redondeada a dos decimales. Si no cumple los requisitos el programa seguirá pidiendo una nota. Cuando se introduzca un valor entre 0 y 10 se mostrará la calificación según la nota:

0 - 2,99: Muy deficiente	**5 - 5,99:** Suficiente	**8 - 8,99:** Notable
3 - 4,99: Insuficiente	**6 - 7'99:** Bien	**9 - 10:** Sobresaliente

4. **Control de Contraseñas WHILE / DO-WHILE con IF (JS 56).**

Realizar un programa que pida por pantalla un nombre de usuario. A continuación pedirá una contraseña y solicitará repetir la contraseña. Se establecerán las siguientes condiciones:

- Si el nombre del usuario está vacío el programa seguirá pidiendo un nuevo nombre de usuario.
- Si las contraseñas están vacías el programa seguirá pidiendo unas nuevas contraseñas.
- Si la primera contraseña es distinta de la segunda el programa mostrará un mensaje de error y pedirá unas nuevas contraseñas.
- Cuando el nombre del usuario y las contraseñas estén correctas el programa finalizará mostrando el nombre del usuario y una de las contraseñas.

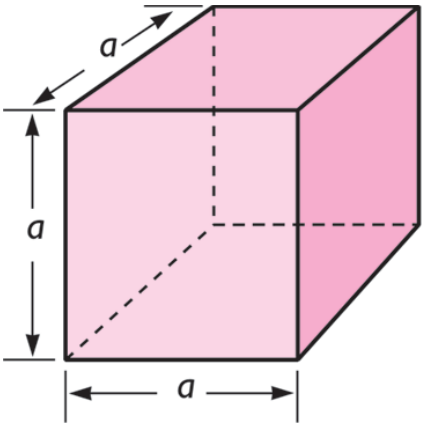
5. **Volumen de un cubo (JS 57).**

Escribir un programa que pida por teclado la arista de un cubo mediante un número. Si el número es menor de cero el programa seguirá pidiendo un número positivo. En el caso de que la arista sea mayor que cero, mostrará por pantalla el mensaje:

"El volumen del cubo de arista <arista> es: <volumen>".

El **volumen de un cubo = arista^3.** Al final del programa se le preguntará al usuario si quiere repetir el proceso o finalizar el programa.

La función para calcular una potencia es **Math.pow(base, exponente).**



$$V_c = a^3$$

6. **Listado de números pares o impares (JS 58).**

Realizar un programa que pida al usuario dos números y una letra: “i” ó “p”. El programa presentará los números pares (si se pulsó la “p”) ó impares (si se pulsó la “i”) que hay desde el primer número al segundo que introdujo el usuario. Si se pulsa alguna tecla distinta de “p” ó “i”, el programa no imprime ningún número.

7. **Contador de números (JS 59).**

Escribir un programa que nos permita introducir números hasta que metamos un 0. Calcular la cantidad de números introducidos hasta ese momento y mostrarlo por pantalla con el mensaje **“Has introducido <cantidad> números”.**

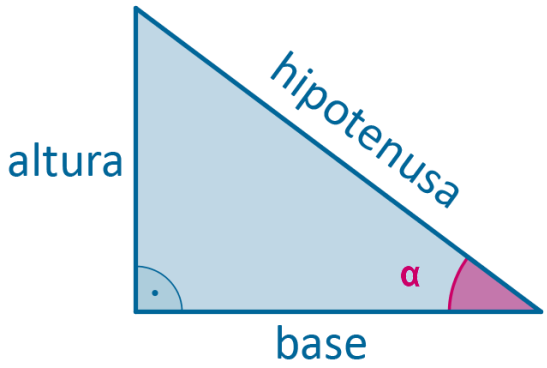
Al final del programa se le preguntará al usuario si quiere repetir el proceso o finalizar el programa.

8. Cálculo de área, perímetro e hipotenusa de un triángulo rectángulo (JS 60).

Escribir un programa que pida por teclado la *base* y la altura de un triángulo rectángulo. Si los números son menores de cero el programa seguirá pidiendo números positivos. El programa deberá calcular el *área* del triángulo, el perímetro y la hipotenusa. Se mostrará el resultado por pantalla ajustado a dos decimales.

- Área de un triángulo = (base * altura) / 2
- Perímetro de un triángulo rectángulo = base + altura + hipotenusa
- Hipotenusa = $\sqrt{base^2 + altura^2}$

La función para la raíz cuadrada es **Math.sqrt(valor a calcular).**
La función para calcular una potencia es **Math.pow(base, exponente).**



9. Bucles FOR (JS 61).

Realizar un programa **USANDO LA SENTENCIA FOR** que imprima por pantalla tantos asteriscos como diga el usuario. Al ejecutarse debe preguntar "Cuantos asteriscos desea imprimir?", leer el número que introduce el usuario e imprimir los asteriscos en HTML.

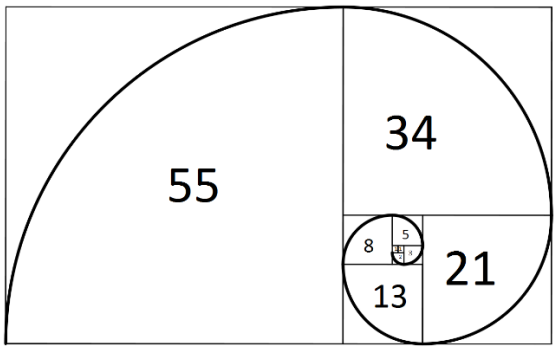
10. Juego del Oráculo y el Adivino (JS 62).

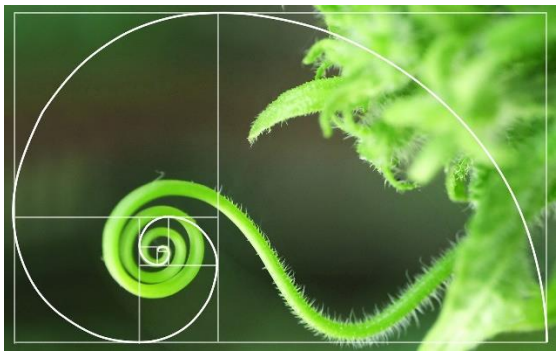
Realizar un programa en el cual un usuario Oráculo introduzca un número entero al azar entre 1 y 20. Si no cumple los requisitos el programa seguirá pidiendo un número. A continuación otro usuario Adivino deberá averiguar cuál es ese número, teniendo 5 oportunidades para acertarlo.

Si el Adivino mete un número menor el programa responderá **"El número que buscas es mayor".** Si mete un número mayor el programa responderá **"El número que buscas es menor".** Si averigua el número saldrá el mensaje **"¡Enhorabuena, acertaste, es el número X!".** Si no lo averigua después de 5 intentos saldrá el mensaje **"¡Oh, lo siento, era el número X!".**

11. Sucesión de Fibonacci usando FOR (JS 63).

La sucesión de Fibonacci es una de las secuencias de números más famosas de la historia. La llaman "el código secreto de la naturaleza" o la "secuencia divina", porque aparece una y otra vez en estructuras naturales, como los pétalos de un girasol o la cáscara de una piña.





La sucesión de Fibonacci, en ocasiones también conocida como secuencia de Fibonacci, es en sí una sucesión matemática infinita. Consta de una serie de números naturales que se suman de a 2, a partir de 0 y 1. Básicamente, la sucesión de Fibonacci se realiza sumando siempre los últimos 2 números. Todos los números presentes en la sucesión se llaman números de Fibonacci, siendo su secuencia de la siguiente manera:

0,1,1,2,3,5,8,13,21,34...

Es decir: **(0+1=1 / 1+1=2 / 1+2=3 / 2+3=5 / 3+5=8 / 5+8=13 / 8+13=21 / 13+21=34…)**
Así sucesivamente, hasta el infinito. Por regla, la sucesión de Fibonacci se escribe así:

$$X_n = X_{n-1} + X_{n-2}$$

Realizar un programa que imprima por pantalla tantos números de la serie de Fibonacci como diga el usuario. Al ejecutarse debe preguntar **"Cuantos números desea imprimir? (mínimo 2 números)",** leer el número que introduce el usuario e imprimir la serie numérica en HTML. El número deberá ser positivo y mayor que 2. Si no cumple los requisitos el programa seguirá pidiendo un número.