

# Ejercicios sobre Script

## Ejercicio SCRIPT1

Modificar el primer script para que:

1. Todo el código JavaScript se encuentre en un archivo externo llamado `codigo.js` y el script siga funcionando de la misma manera.
2. Después del primer mensaje, se debe mostrar otro mensaje que diga *"Soy el primer script"*
3. Añadir algunos comentarios que expliquen el funcionamiento del código
4. Añadir en la página XHTML un mensaje de aviso para los navegadores que no tengan activado el soporte de JavaScript

[Descargar página HTML](#)

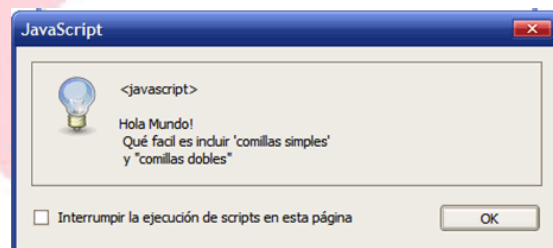
[Ver solución JS](#)

[Ver solución HTML](#)

## Ejercicio SCRIPT2

Modificar el primer script del capítulo anterior para que:

1. El mensaje que se muestra al usuario se almacene en una variable llamada `mensaje` y el funcionamiento del script sea el mismo.
2. El mensaje mostrado sea el de la siguiente imagen:



**Figura** Nuevo mensaje que debe mostrar el script

[Descargar página HTML](#)

[Ver solución](#)

## Ejercicio SCRIPT3

Crear un array llamado `meses` y que almacene el nombre de los doce meses del año. Mostrar por pantalla los doce nombres utilizando la función `alert()`.

[Ver solución](#)

**Ejercicio SCRIPT4**

A partir del siguiente array que se proporciona: `var valores = [true, 5, false, "hola", "adios", 2];`

1. Determinar cual de los dos elementos de texto es mayor
2. Utilizando exclusivamente los dos valores booleanos del array, determinar los operadores necesarios para obtener un resultado `true` y otro resultado `false`
3. Determinar el resultado de las cinco operaciones matemáticas realizadas con los dos elementos numéricos

[Ver solución](#)

**Ejercicio SCRIPT5**

Completar las condiciones de los `if` del siguiente script para que los mensajes de los `alert()` se muestren siempre de forma correcta:

```
var numero1 = 5;
var numero2 = 8;

if(...) {
    alert("numero1 no es mayor que numero2");
}
if(...) {
    alert("numero2 es positivo");
}
if(...) {
    alert("numero1 es negativo o distinto de cero");
}
if(...) {
    alert("Incrementar en 1 unidad el valor de numero1 no lo hace mayor o igual que numero2");
}
```

[Ver solución](#)

Prof. Miguel Ángel Gómez

**Ejercicio SCRIPT6**

El cálculo de la letra del Documento Nacional de Identidad (DNI) es un proceso matemático sencillo que se basa en obtener el resto de la división entera del número de DNI y el número 23. A partir del resto de la división, se obtiene la letra seleccionándola dentro de un array de letras.

El array de letras es:

```
var letras = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E', 'I'];
```

Por tanto si el resto de la división es 0, la letra del DNI es la T y si el resto es 3 la letra es la A. Con estos datos, elaborar un pequeño script que:

1. Almacene en una variable el número de DNI indicado por el usuario y en otra variable la letra del DNI que se ha indicado. (Pista: si se quiere pedir directamente al usuario que indique su número y su letra, se puede utilizar la función `prompt()`)
2. En primer lugar (y en una sola instrucción) se debe comprobar si el número es menor que 0 o mayor que 99999999. Si ese es el caso, se muestra un mensaje al usuario indicando que el número proporcionado no es válido y el programa no muestra más mensajes.
3. Si el número es válido, se calcula la letra que le corresponde según el método explicado anteriormente.
4. Una vez calculada la letra, se debe comparar con la letra indicada por el usuario. Si no coinciden, se muestra un mensaje al usuario diciéndole que la letra que ha indicado no es correcta. En otro caso, se muestra un mensaje indicando que el número y la letra de DNI son correctos.

[Ver solución](#)

#### Ejercicio SCRIPT7

El factorial de un número entero  $n$  es una operación matemática que consiste en multiplicar todos los factores  $n \times (n-1) \times (n-2) \times \dots \times 1$ . Así, el factorial de 5 (escrito como  $5!$ ) es igual a:  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

Utilizando la estructura `for`, crear un script que calcule el factorial de un número entero.

[Ver solución](#)

#### Ejercicio SCRIPT8

Escribir el código de una función a la que se pasa como parámetro un número entero y devuelve como resultado una cadena de texto que indica si el número es par o impar. Mostrar por pantalla el resultado devuelto por la función.

[Ver solución](#)

#### Ejercicio SCRIPT9

Definir una función que muestre información sobre una cadena de texto que se le pasa como argumento. A partir de la cadena que se le pasa, la función determina si esa cadena está formada sólo por mayúsculas, sólo por minúsculas o por una mezcla de ambas.

[Ver solución](#)

#### Ejercicio SCRIPT10

Definir una función que determine si la cadena de texto que se le pasa como parámetro es un palíndromo, es decir, si se lee de la misma forma desde la izquierda y desde la derecha. Ejemplo de palíndromo complejo: "La ruta nos aporoto otro paso natural".

[Ver solución](#)

#### Ejercicio SCRIPT11

Generar un Script que utilizando matrices en dos dimensiones, primero rellene una matriz 2D y después genere una salida por pantalla como la que se muestra en pantalla, usando la función alert().

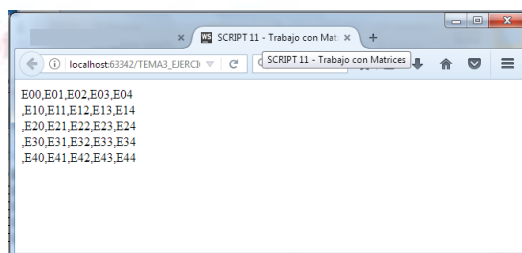
```
E00,E01,E02,E03,E04  
.E10,E11,E12,E13,E14  
.E20,E21,E22,E23,E24  
.E30,E31,E32,E33,E34  
.E40,E41,E42,E43,E44
```

Aceptar

[Ver solución](#)

#### Ejercicio SCRIPT12

Ampliar el ejercicio anterior y generar un Script que utilizando matrices en dos dimensiones, primero rellene una matriz 2D y después genere una salida por pantalla como la que se muestra en pantalla



[Ver solución](#)

# Ejercicios sobre Expresiones Regulares

## Ejercicio ER1

Genera un script que valide una dirección de email obtenida de un cuadro de texto, teniendo en cuenta que una dirección de email se compone de: nombre\_usuario + @ + servidor + dominio

### AYUDA

La expresión regular delimita su inicio y fin con `/^ expr regular $/`

#### Nombre y servidor

Debe empezar por letra o número. Al menos tiene una letra o número. Para asegurarnos de que la letra o número aparece al menos una vez utilizaremos el modificador `+`.

Puede contener puntos y guiones además de las letras y números. Esta combinación podrá aparecer, es por ello que se utilizará el modificador `*` (cero o varias veces).

Resultado: `\w+([\.-]?\w+)*`

#### dominio

Irá al final, detrás de un punto. Podrá tener dos (.es, .fr, .it,...) o tres letras (.com, .net, .org,...)

Además puede tener varios dominios seguidos (.com.ar,.com.uk.), es por ello que se deberá usar el modificador `+` ya que el dominio podrá aparecer varias veces.

Resultado: `(\.\w{2,3})+`

#### Nombre y servidor + dominio

Resultado: `/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/`

[Descargar página HTML](#)

[Ver solución HTML](#)

## Ejercicio ER2

Crear una expresión regular para campos de teléfono que compruebe que se han introducido sólo números, espacios o el carácter '-' en el campo correspondiente. Para validar los campos deberemos utilizar el método test de la expresión regular correspondiente. Este método compara la cadena que se le pasa como argumento con el patrón de la expresión regular.

[Ver solución HTML](#)

## Ejercicio ER3

Realizar una página Web que tenga un botón y un párrafo, y en ese párrafo un texto como se muestra en la imagen:

**Expresiones Regulares - Metodo exec.**

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lentejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda.

Buscar la palabra "hidalgo"

Metodo exec.

Se desea crear un Script que se ejecute al pulsar el botón localice la palabra “galgo”, indique en qué posición se encuentra y el texto donde se ha encontrado, como se muestra en la imagen

**Expresiones Regulares - Metodo exec.**

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lentejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda.

Buscar la palabra "galgo"

Metodo exec.

palabra buscada: galgo
posición: 164
Texto en que se encuentra: En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lentejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda.

[Ver solución HTML](#)

**Ejercicio ER4**

Ampliar la página web y el Script, usando el objeto RegExp de tal forma que aparezca 2 nuevos cuadros de texto más y un botón y que cuando se pulse el botón y usando la expresión regular, el objeto RegExp dado y el texto suministrado en uno de los nuevos cuadros de texto se reemplace en el objeto RegExp según la expresión regular dejando el resultado en el segundo cuadro de texto.

[Descargar página HTML](#)

[Ver solución HTML](#)

**Ejercicio ER5**

Crear una página web como la que aparece en la imagen y que cuando se haga click sobre los párrafos se ejecute sobre el texto superior lo que indica en cada párrafo.

**Expresiones Regulares****Texto de ejemplo**

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lentejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda.

Restablecer texto a su estado original

Resaltar letra "e"

Remplazar artículos determinados por guión bajo.

Remplazar espacios por guión

**Propiedades del objeto regExp:**

Cadena explorada: **input**

Última coincidencia encontrada: **lastMatch**

Indica si la cadena tiene saltos de línea: **multiline**

desde el principio hasta la coincidencia: **leftContext**

desde la coincidencia hasta el final: **rightContext**

- Restablecer texto a su estado original
- Resaltar letra "e"
- Remplazar artículos determinados por guión bajo.
- Remplazar espacios por guión

Propiedades del objeto regExp:

- Cadena explorada: **input**
- Última coincidencia encontrada: **lastMatch**
- Indica si la cadena tiene saltos de línea: **multiline**
- desde el principio hasta la coincidencia: **leftContext**
- desde la coincidencia hasta el final: **rightContext**

[Ver solución HTML](#)

#### Ejercicio ER6

Genera todo el código necesario para crear una página WEB que disponga de un campo que permita seleccionar un fichero de cualquier parte de mi ordenador y que dicho fichero una vez seleccionado sea validado como imagen. Para esta validación deberemos usar expresiones regulares comprobando que dicho archivo es del tipo **PNG, JPG, GIF, png, jpg, gif, Png, Jpg, Gif, pNg, jPg, gIf, ...** (dicha validación deberá ser **CASE SENSITIVE**).

[Ver solución HTML](#)

[Ver solución HTML a](#)

[Ver solución HTML b](#)

DESARROLLO WEB EN ENTORNO CLIENTE

Prof. Miguel Ángel Gómez



# Ejercicios sobre DOM

## Ejercicio DOM1

A partir de la página web proporcionada y utilizando las funciones DOM, mostrar por pantalla la siguiente información:

1. Número de enlaces de la página
2. Dirección a la que enlaza el penúltimo enlace
3. Numero de enlaces que enlazan a `http://prueba`
4. Número de enlaces del tercer párrafo

[Descargar página HTML](#)

[Ver solución](#)

## Ejercicio DOM2

Completar el código JavaScript proporcionado para que cuando se pinche sobre el enlace se muestre completo el contenido de texto. Además, el enlace debe dejar de mostrarse después de pulsarlo por primera vez. La acción de pinchar sobre un enlace forma parte de los "Eventos" de JavaScript que se ven en el siguiente capítulo. En este ejercicio, sólo se debe saber que al pinchar sobre el enlace, se ejecuta la función llamada `muestra()`.

[Descargar página HTML](#)

[Ver solución](#)

## Ejercicio DOM3

Completar el código JavaScript proporcionado para que se añadan nuevos elementos a la lista cada vez que se pulsa sobre el botón. Utilizar las funciones DOM para crear nuevos nodos y añadirlos a la lista existente. Al igual que sucede en el ejercicio anterior, la acción de pinchar sobre un botón forma parte de los "Eventos" de JavaScript que se ven en el siguiente capítulo. En este ejercicio, sólo se debe saber que al pinchar sobre el botón, se ejecuta la función llamada `anade()`.

[Descargar página HTML](#)

[Ver solución](#)



# Ejercicios sobre Eventos

## Ejercicio EVENTOS1

A partir de la página web proporcionada, completar el código JavaScript para que:

1. Cuando se pinche sobre el primer enlace, se oculte su sección relacionada
2. Cuando se vuelva a pinchar sobre el mismo enlace, se muestre otra vez esa sección de contenidos
3. Completar el resto de enlaces de la página para que su comportamiento sea idéntico al del primer enlace
4. Cuando una sección se oculte, debe cambiar el mensaje del enlace asociado (pista: propiedad `innerHTML`)

[Descargar página HTML](#)

[Ver solución](#)

## Ejercicio EVENTOS2

Completar el código JavaScript proporcionado para que:

1. Al mover el ratón en cualquier punto de la ventana del navegador, se muestre la posición del puntero respecto del navegador y respecto de la página:

**Figura 6.1** Información que se muestra para los eventos del ratón

### Ratón

Navegador [326, 432]

Página [326, 88]

Para mostrar los mensajes, utilizar la función `muestraInformacion()` deduciendo su funcionamiento a partir de su código fuente.

2. Al pulsar cualquier tecla, el mensaje mostrado debe cambiar para indicar el nuevo evento y su información asociada:

**Figura 6.2** Información que se muestra para los eventos del teclado

### Teclado

Carácter [a]

Código [97]

3. Añadir la siguiente característica al script: cuando se pulsa un botón del ratón, el color de fondo del cuadro de mensaje debe ser amarillo (`#FFFFCC`) y cuando se pulsa una tecla, el color de fondo debe ser azul (`#CCE6FF`). Al volver a mover el ratón, el color de fondo vuelve a ser blanco.

### Ratón

Navegador [606, 457]

Página [606, 113]

### Teclado

Carácter [c]

Código [99]

**Figura 6.3** El color de fondo del cuadro de información cambia en función del tipo de evento

[Descargar página HTML](#)

[Ver solución](#)

## Ejercicio EVENTOS3

Crear un script que informe al usuario en que zona de la pantalla ha pulsado el ratón. Las zonas definidas son las siguientes: izquierda arriba, izquierda abajo, derecha arriba y derecha abajo. Para determinar el tamaño de la ventana del navegador, utilizar la función *tamanoVentanaNavegador()* proporcionada.

[Descargar página HTML](#)

[Ver solución](#)

## Ejercicios sobre Calendarios

## Ejercicio CALENDARIO1

Mejorar el calendario creado añadiendo las opciones necesarias para que muestre el siguiente aspecto:



**Figura 8.6** Aspecto mejorado del calendario JavaScript

1. Que no se muestre el número de la semana en el calendario (pista: `weekNumbers`)
2. Modificar el formato en el que se muestra la fecha seleccionada. El formato original es 21 / 08 / 2007 (indicado como `%d / %m / %Y`). El formato deseado es Martes, 21 de Agosto de 2007 (pistas: `%A, %B`)
3. El nuevo formato de fecha es mucho más agradable para los usuarios, pero más incómodo para los scripts que tienen que manejarlo. Afortunadamente, el calendario dispone de la posibilidad de guardar dos valores: un valor para mostrar a los usuarios y otro valor para que lo procesen los scripts. Cuando el usuario seleccione una fecha, la fecha con el formato original se debe almacenar en un campo oculto de formulario y el otro formato más largo debe mostrar al usuario en un elemento de tipo `<span>` (pistas: `displayArea, daFormat`)

[Ver solución](#)

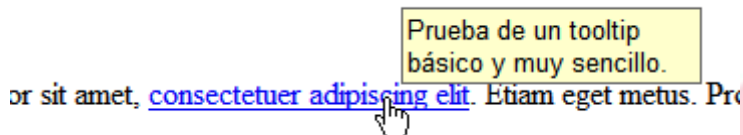
# Ejercicios sobre ToolTip

## Ejercicio TOOLTIP1

Mejorar el *tooltip* propuesto añadiendo las siguientes características:

1. Que el *tooltip* no se muestre instantáneamente, sino que transcurra un cuarto de segundo hasta que se muestre (pista: `DELAY`)
2. Que exista una separación horizontal de 15 píxel entre el puntero del ratón y el *tooltip* (pista: `OFFSETX`)
3. Que el *tooltip* se muestre en la parte superior del puntero del ratón y no en la parte inferior (pista: `ABOVE`)

El nuevo aspecto del tooltip se muestra en la siguiente imagen:



**Figura** Nuevo aspecto del tooltip

[Ver solución](#)

# Ejercicios sobre Reloj pantalla

## Ejercicio RELOJ1

Genera una página web que muestre un reloj en formato digital donde se visualice la fecha + hora + minutos + segundos en formato decimal y en tiempo real, es decir que la hora vaya avanzando según avance el tiempo.

Parar

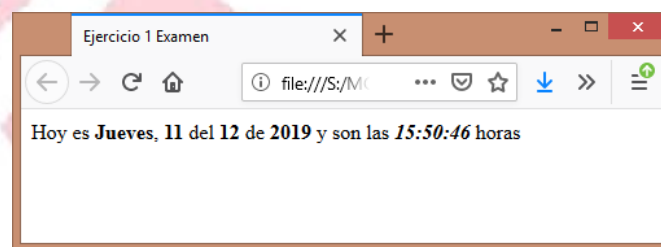
06/11/2018 - 17:49:52

[Ver solución](#)

## Ejercicio RELOJ2

Realiza un script que muestre un reloj en pantalla con fecha y hora y que se actualice cada segundo, con las siguientes premisas.

- El reloj deberá funcionar como el del vídeo e imagen adjuntos y para su creación no se podrá usar ningún tipo de Bucle.
- Algunas palabras y números deberán estar en negrita.
- Se deberá reducir al máximo el código.
- El código se ejecutará al abrir la página en el navegador.



[Ver solución](#)

# Ejercicios Ampliación

Ejercicios Ampliación1
------------------------

Ampliar todos los ejercicios anteriores añadiéndoles un control de errores, de tal manera que si por ejemplo, se produjese una entrada errónea de datos (una letra en lugar de un número) el Script respondiese con funcionamiento correcto y coherente.



# Ejercicios CANVAS

## Ejercicio CANVAS 1

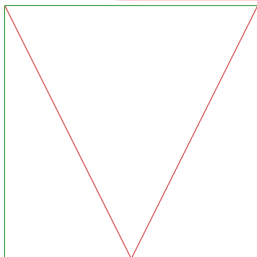
Dibujar dos rectángulos superpuestos (como los de la imagen) uno de color rojo y el otro de color azul con un índice de transparencia del 50%



[Ver solución](#)

## Ejercicio CANVAS 2

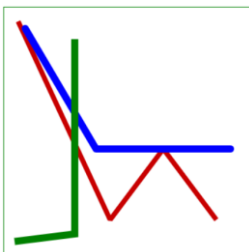
Dibujar un lienzo de dimensiones 350 x 350 colocarle un borde verde y dentro de él dibujar mediante líneas una “V” desde de los vértices superiores hasta el centro del lado inferior



[Ver solución](#)

## Ejercicio CANVAS 3

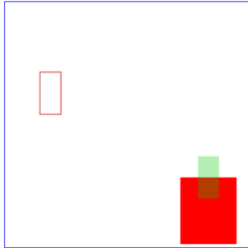
Dibujar un lienzo de dimensiones 350 x 350 colocarle un borde verde y dentro de él 3 líneas como las de la figura usando Grosor de la línea, estilo de la línea y estilos de unión de las líneas.



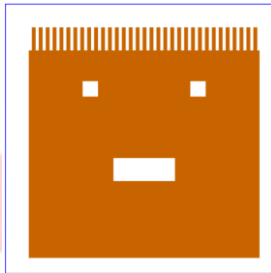
[Ver solución](#)

**Ejercicio CANVAS 4**

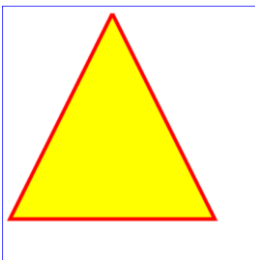
Dibujar un lienzo de dimensiones 350 x 350 colocarle un borde azul y dentro de él 3 rectángulos como los de la figura usando en el primero la función de rectángulo vacío, en el segundo rectángulo rojo relleno y en el tercero un rectángulo verde con filtro alfa y que la mitad del mismo se superponga al segundo rectángulo.

[Ver solución](#)**Ejercicio CANVAS 5**

Dibujar un lienzo de dimensiones 350 x 350 colocarle un borde azul y dentro de él un rectángulo marrón posteriormente a ese rectángulo le iremos borrando (limpiando) áreas hasta que quede como el de la imagen .

[Ver solución](#)**Ejercicio CANVAS 6**

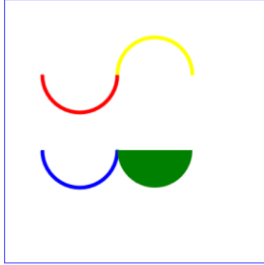
Dibujar un lienzo de dimensiones 350 x 350 colocarle un borde azul y dentro de él un triángulo con borde rojo y fondo amarillo como el de la imagen .

[Ver solución](#)

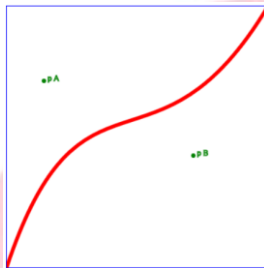


**Ejercicio CANVAS 7**

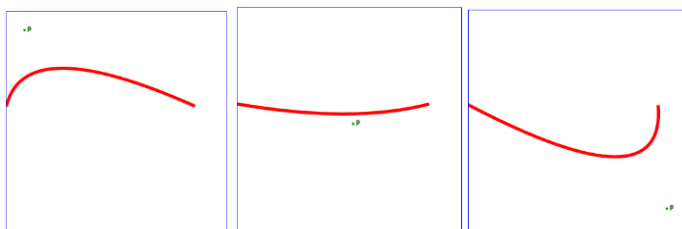
Dibujar un lienzo de dimensiones 350 x 350 4 arcos como los de la imagen con los colores rojo, amarillo, azul y el último verde relleno .

[Ver solución](#)**Ejercicio CANVAS 8**

Dibujar un lienzo de dimensiones 350 x 350 con borde azul y dentro de él dibujar la función de Bezier entre los puntos inicial (0 , 350) final (350 , 0) y punto del arco 1 (100 , 20) y del arco 2 (200 , 250), el resultado tendrá que ser algo similar al de la imagen.

[Ver solución](#)**Ejercicio CANVAS 9**

Dibujar un lienzo de dimensiones 350 x 350 con borde azul y dentro de él dibujar la función “quadraticCurveTo(cpx,cpy,x,y)” de tal manera que vayamos incrementando los 2 primeros valores en saltos de 1px (es decir en diagonal) la posición de x=300 y de y=150. La curva deberá dibujar un movimiento como el de las imágenes.

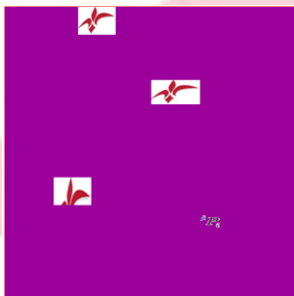
[Ver solución](#)

**Ejercicio CANVAS 10**

Dibujar un lienzo de dimensiones 350 x 450 en el escribir 3 textos como los de la imagen, el primero relleno de verde, el segundo con las letras rojas huecas y el tercero igual que el segundo pero con color degradado (amarillo, rojo, azul, rosa).

[Ver solución](#)**Ejercicio CANVAS 11**

Dibujar un lienzo de dimensiones 600 x 600 en el visualizar las imágenes como las de la figura, la primera mantiene la imagen igual que su formato original, la segunda se deforma en horizontal y la tercera se deforma y corta.

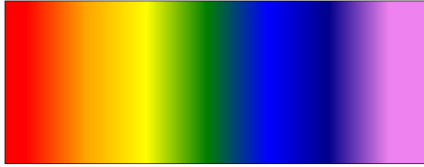
[Ver solución](#)**Ejercicio CANVAS 12**

Dibujar un lienzo de dimensiones 500 x 300 en el visualizar las imágenes con sombra como las de la figura.

[Ver solución](#)

**Ejercicio CANVAS 13**

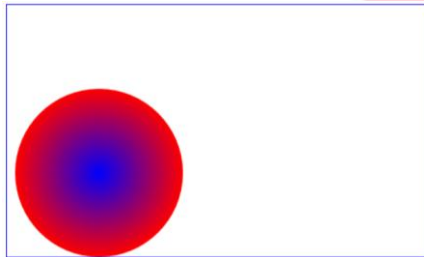
Dibujar un lienzo de dimensiones 500 x 300 en él visualizar la imagen como la de la figura con un gradiente lineal (rojo, naranja, amarillo, verde, azul, azul oscuro, violeta)..



[Ver solución](#)

**Ejercicio CANVAS 14**

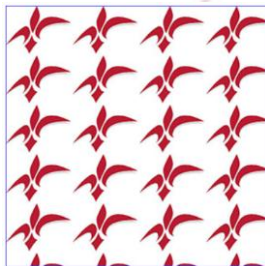
Dibujar un lienzo de dimensiones 520 x 200 en él visualizar la imagen como la de la figura con un gradiente radial (azul, rojo).



[Ver solución](#)

**Ejercicio CANVAS 15**

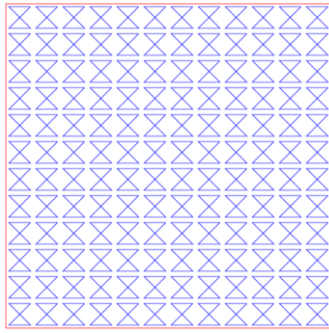
Dibujar un lienzo de dimensiones 300 x 300 en él visualizar la imagen como la de la figura con un patrón que se repita.



[Ver solución](#)

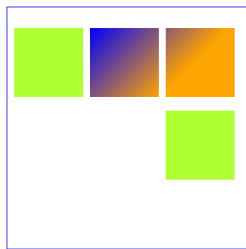
**Ejercicio CANVAS 16**

Dibujar dos lienzo de dimensiones 50 x 50 y 600 x 600 respectivamente, en él primero dibujar con líneas la figura de la imagen, posteriormente tomar el primer lienzo como patrón y dibujarlo en el segundo lienzo repitiéndolo como en la imagen.

[Ver solución](#)

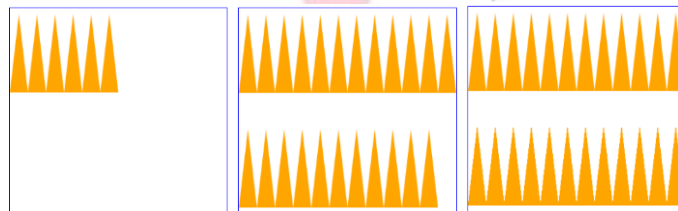
#### Ejercicio CANVAS 17

Dibujar un lienzo de dimensiones 350 x 350, en él dibujar 4 rectángulos como los de la imagen. Para ello crearemos una función que los pueda realizar automáticamente con el formato “micuadradoGradiente(lienzo,x,y,ancho,alto)”. Una vez realizada la función primero dibujaremos un rectángulo verde con “fillRect”, posteriormente usaremos el método “save()”, luego usaremos la función creada, después usaremos el método “restore()” y volveremos a dibujar otro rectángulo verde

[Ver solución](#)

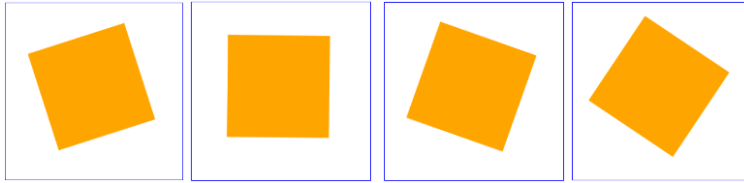
#### Ejercicio CANVAS 18

Dibujar un lienzo de dimensiones 360 x 350, en él dibujar mediante un método que crearemos nosotros llamado “dibujarTriangulo(lienzo,x,y,base,altura)” triángulos como los de la imagen. Nuestro script rellenará con un bucle todo el área del canvas repitiendo la figura que genera la función que hemos creado primero en horizontal y después en vertical,

[Ver solución](#)

### Ejercicio CANVAS 19

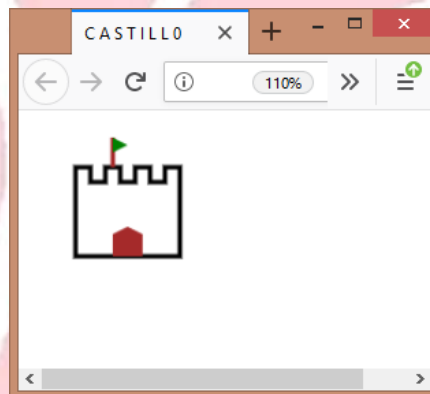
Dibujar un lienzo de dimensiones 350 x 350, en él dibujar un rectángulo que mediante el método `rotate()` gire sobre su eje,



[Ver solución](#)

### Ejercicio CANVAS 20

Crear por código Javascript una figura como la de la imagen adjunta sin utilizar imágenes de ningún tipo.



[Ver solución](#)

Prof. Miguel Ángel Gómez