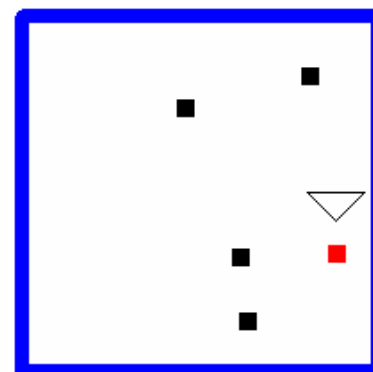


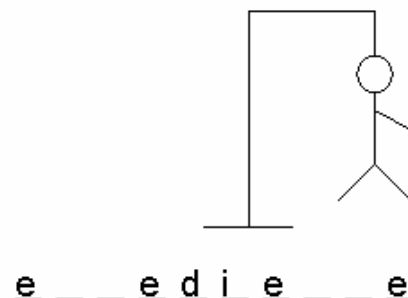
INTRODUCCIÓN

Vamos a continuar analizando nuevas instrucciones y posibilidades de MSWLogo, fijándonos como meta la realización de otros juegos:

En primer lugar, se trata de dirigir la tortuga, con el teclado, para que vaya chocando con los cuadrados rojos que irán apareciendo al azar en el marco del juego. Cada vez que choca con uno de ellos se gana un punto y el cuadrado cambia de color (negro). Si la tortuga choca con uno de los cuadrados negros o con el marco azul, el juego finaliza.



El segundo juego es el conocido con el nombre del "ahorcado": hay que adivinar una palabra, previamente introducida por un compañero, indicando las letras que la forman; cuando la letra forma parte de la palabra se coloca en el lugar adecuado, si no, se dibuja una parte de la figura del ahorcado. El juego finaliza al acertar la palabra completa o cuando el número de errores es suficiente para que la figura se haya dibujado completamente.



MÁS INSTRUCCIONES GRÁFICAS

poncolorrelleno [rojo verde azul]

poccr [rojo verde azul] Selecciona el color que será utilizado posteriormente por la instrucción **rellena**.

rellena

Rellena del color previamente fijado con la instrucción **poncolorrelleno** el contorno cerrado en cuyo interior se encuentre la tortuga.

Ejemplo: Las instrucciones siguientes dibujan un rectángulo y lo rellenan de color rojo. Observar que una vez dibujado el cuadrado hay que colocar la tortuga en su interior antes de relleno de color.

```
repite 4[av 60 gd 90]
sl gd 45 av 20
poccr 4 rellena
```

rectangulorrelleno ancho alto

Dibuja un rectángulo de las dimensiones indicadas y lo rellena del color que previamente hayamos seleccionado con **poncolorrelleno**. El rectángulo se dibuja a partir de la posición de la tortuga (esquina inferior izquierda).

Ejemplo: Las instrucciones siguientes hacen el mismo efecto que el ejemplo anterior, dibujan un cuadrado de lado 60 y lo rellenan de color rojo.

```
poccr 4 rectangulorrelleno 60 60
```

circulo radio

Dibuja un círculo del radio indicado tomando como centro la posición de la tortuga.

rotula "palabra

rotula "[frase] Escribe la palabra o frase indicada en la pantalla gráfica, iniciándola en la posición de la tortuga y con su orientación actual. Una vez finalizada su ejecución la tortuga mantiene la posición y orientación inicial.

Ejemplo:

```
rotula "|Hola Mundo|
gd 90 rotula "|Hola Mundo|
```

Si escribimos una palabra en una determinada posición de la pantalla gráfica y, posteriormente, queremos escribir otra palabra en la misma posición, tenemos que borrar la palabra inicial, volviéndola a escribir con color blanco o borrando toda la pantalla gráfica.

ACTIVIDADES:

1. Realizar un programa que cuente, mostrando el número en la pantalla gráfica, las veces que accionamos el teclado. Hacerlo de dos maneras, utilizando la instrucción bp y sin usarla.

PALABRAS Y LISTAS

En el lenguaje LOGO se denomina palabra a un grupo de caracteres que no contengan espacios en blanco y algunos signos especiales como las comillas o los de puntuación. Las palabras deben ir precedidas de comillas, salvo si sólo están formadas por caracteres numéricos.

Son palabras: "HOLA 23 "23 d48c
 No son palabras: "hola mundo "hola"mundo

Las listas son grupos de palabras, o de otras listas, que empiezan y terminan con un corchete y van separadas por espacios en blanco. Las palabras que forman la lista no necesitan empezar con las comillas, pues LOGO sabe que lo que hay dentro de los corchetes son palabras. Las listas se utilizan para almacenar datos y su manejo es fundamental para sacar el máximo rendimiento al lenguaje LOGO.

Lista de dos palabras: [hola mundo]
 Lista de tres palabras: [hoy es martes]
 Lista de 4 elementos, 3 palabras y otra lista: [hoy es lunes [9 de agosto]]

Las instrucciones y los procedimientos no son palabras y, por tanto, no necesitan identificador. Cuando escribimos algo, que no sea un número, sin identificador (" en las palabras, [en las listas o : cuando nos referimos al valor de una variable) LOGO los interpreta como una instrucción, como un procedimiento o no la entiende y, entonces, emite un mensaje de error.

DIBUJAR CUADRADOS EN POSICIONES ALEATORIAS

Vamos a ver, en primer lugar, como MSWLogo puede generar números aleatorios a través de la instrucción azar:

azar número1

Genera un número entero positivo aleatorio, menor que el argumento que debe ser un número entero.

Ejemplo: azar 50 34
 azar 50 7
 (100 - azar 200) -37 así se pueden generar números positivos o negativos.
 (100 - azar 200) 74

Veamos, ahora, como podemos dibujar cuadrados en posiciones aleatorias:

Por ejemplo, para dibujar un cuadrado (relleno de color rojo) de lado 10 en la posición [30 50] podemos utilizar las instrucciones siguientes:

```
ponpos [30 50]
poccr 4 rectangulorrelleno 10 10
```

Si queremos que la posición en la que se dibuje sea aleatoria tenemos que generar las coordenadas de ponpos utilizando la instrucción azar, por ejemplo:

```
ponpos [azar 100 azar 100]
```

De esta manera, las coordenadas x e y serían sendos números, al azar, comprendidos entre 0 y 100.

Ahora bien, si ejecutamos la instrucción anterior obtendríamos un mensaje de error:

```
ponpos no reconoce [azar 100 azar 100]como entrada
```

ya que, dentro de los corchetes de ponpos, azar es tomado como una palabra y no como una instrucción que hay que ejecutar. Para que funcione correctamente tenemos que utilizar una nueva instrucción:

```
(lista cosa1 cosa2 cosa3 ...)
```

lista cosa1 cosa2 Devuelve una lista cuyos miembros son las entradas, que pueden ser cualquier objeto de Logo (palabra, variable, instrucción, otra lista,...).

```
Ejemplo: haz "x azar 100
          haz "y azar 100
          ponpos lista :x :y

          o tambien:
          ponpos lista azar 100 azar 100
```

ACTIVIDADES:

- Realizar un procedimiento, denominado `dibuja_marco`, que dibuje en el centro de la pantalla un cuadrado de 200 puntos de lado, grosor 3 y color azul. La tortuga terminará colocada en el centro del cuadrado, el color se volverá a colocar en negro y el grosor en 1.
- Hacer un procedimiento, denominado `cuadrado_al_azar`, que dibuje (dentro del marco generado por el procedimiento anterior) un cuadrado relleno de color rojo en posiciones obtenidas al azar.

TRABAJAR CON MÚLTIPLES TORTUGAS

MSWLogo puede trabajar con varias tortugas a la vez, manteniendo cada tortuga su propia dirección, posición y control del lápiz, aunque no el color del lápiz. Para seleccionar una determinada tortuga se utiliza la instrucción activa:

```
activa número de la tortuga
```

Selecciona y activa la tortuga indicada para poder controlarla. Las tortugas comienzan con el número 0, que es la tortuga por defecto.

```
Ejemplo: para multiples
          activa 0 gd 1 av 1 espera 1
          activa 1 gi 1 av 1 espera 1
          multiples
          fin
```



ACTIVIDADES:

- Realizar Un programa para que cuatro tortugas se vayan alejando del centro de la pantalla (se abran en diagonal).

UNA PRIMERA VERSIÓN DEL JUEGO

El programa que se muestra a continuación permite mover una tortuga, desde el teclado, dentro del marco del juego. Cuando choca con un cuadrado rojo este cambia de color (negro) y se dibuja, en una posición al azar, un nuevo cuadrado rojo. Si la tortuga choca con el borde azul o con uno de los cuadrados negros del juego finaliza.

```
para cabezaserpiente
  bp bt
  dibuja_marco
  ponfoco [Pantalla de MSWlogo]
  ponteclado [control_movimiento]
  cuadrado_al_azar
  movimiento
  fin
```

```

para dibuja_marco
pon grosor [8 8]
pon cl [0 0 255]
sl ponpos [-100 -100] bl repite 4 [av 200 gd 90]
sl ponpos [0 0]
bl poncl [0 0 0] pongrosor [1 1] mt
fin

para control_movimiento
haz "tecla car lc
si :tecla = "o [ponrumbos 0] ;arriba
si :tecla = "l [ponrumbos 180] ;abajo
si :tecla = "k [ponrumbos 270] ;izquierda
si :tecla = "ñ [ponrumbos 90] ;derecha
fin

para cuadrado_al_azar
activa 1 ot sl
haz "x (80 - azar 160) haz "y (80 - azar 160) ;coordenadas al azar
ponpos lista :x :y
bl poccr [255 0 0] rectangulorrelleno 10 10
fin

para movimiento
activa 0 ot sl av 4
;si rojo cambia cuadrado y dibuja otro
si pixel = [255 0 0] [av 11 cambia_cuadrado cuadrado_al_azar activa 0]
si pixel = [0 0 255] [alto] ;si azul choque con el marco, fin
si pixel = [0 0 0] [alto] ;si negro choque con un cuadrado negro, fin
re 4 mt
av 1 repite 12000[]
movimiento
fin

para cambia_cuadrado
activa 1 ot bl
poccr [0 0 0] rectangulorrelleno 10 10
fin

```

ACTIVIDADES:

- Ampliar el programa anterior para que muestre en la pantalla gráfica (hay que usar la instrucción rotula) los puntos que va obteniendo el jugador (puede resultar interesante activar otra tortuga para este menester).
- Modificar el programa para que la tortuga, cada vez que choque con un cuadrado rojo, aumente ligeramente su velocidad (habrá que crear una variable que disminuya de valor cada vez que se produce un choque y utilizarla en la instrucción repite que marca el retraso entre los avances de la tortuga).

EL AHORCADO. PROCEDIMIENTOS CON PARÁMETROS.

Vamos a iniciar el juego del ahorcado construyendo un procedimiento que haga los dibujos de la horca y las distintas partes de la figura humana (cabeza, cuerpo, brazos y piernas). La idea es la siguiente: realizar un procedimiento en el que, junto al nombre, indiquemos el número de errores cometidos, para que dibuje sólo la parte que corresponda en ese momento. Es decir, le vamos a pasar al procedimiento una variable o parámetro, que puede ser usado por las instrucciones que lo forman.

```

para fallo :n
si :n=1 [horca] ;si escribimos fallo 1 se dibujará la horca
si :n=2 [cuerda] ;si escribimos fallo 2 se dibujará la cuerda
si :n=3 [cabeza] ;y así sucesivamente
si :n=4 [cuerpo]
si :n=5 [piernaderecha]

```

```

si :n=6 [piernaizquierda]
si :n=7 [brazoderecho]
si :n=8 [brazoizquierdo]
fin

```

ACTIVIDADES:

7. Realizar los procedimientos horca, cuerda, cabeza, etc. Se puede tomar como modelo el dibujo de la página 1 y comenzar la base de la horca en la posición [105 20].
8. Realizar un procedimiento para dibujar un rectángulo (rectángulo :n :m) de cualquier tamaño; el largo y el ancho se pasan en el nombre del procedimiento como parámetros.

ALGUNAS INSTRUCCIONES PARA TRABAJAR CON PALABRAS Y LISTAS

primero lista, palabra o número

pri lista, palabra o número Devuelve el primer elemento, carácter o dígito de la lista, palabra o número indicado.

Ejemplo: es pri [cara cruz] cara
 es pri "Antonio A
 es pri 234 2

último lista, palabra o número

ul lista, palabra o número Devuelve el último elemento de la lista, palabra o número indicado.

Ejemplo: es ul [cara cruz] cruz

menosprimero lista, palabra o número

mp lista, palabra o número Devuelve la lista, palabra o número indicado sin su primer elemento.

Ejemplo: es mp [a b c d] b c d

menosultimo lista, palabra o número

mu lista, palabra o número Devuelve la lista, palabra o número sin su último elemento.

Ejemplo: es mu [a b c d] a b c
 es mu [a b [c d]] a b

ponprimero objeto [lista]

pp objeto [lista] Devuelve la lista que resulta de añadir el objeto indicado como primer elemento de la misma.

Ejemplo: es pp "Marte [Júpiter Saturno] Marte Júpiter Saturno

ponultimo objeto [lista]

pu objeto [lista] Devuelve la lista que resulta de añadir el objeto indicado como último elemento de la misma.

Ejemplo: es pu "Marte [Júpiter Saturno] Júpiter Saturno Marte

elemento índice lista o palabra

Devuelve el carácter o palabra número índice de la palabra o lista indicada.

Ejemplo: es elemento 3 [a b c d] c
 es elemento 2 "Juan u

cuenta lista o palabra

Devuelve el número de elementos o caracteres de la lista o palabra indicada.

Ejemplo: es cuenta [a b c d] 4

miembro? objeto1 objeto2

Devuelve verdadero si el primer objeto es miembro del segundo, en caso contrario devuelve falso.

Ejemplo: es miembro? b [a b c d] verdadero
 es miembro? h [a b c d] falso

vacio? objeto

Devuelve verdadero si la entrada es una palabra o lista vacía, falso en caso contrario.

Ejemplo: es vacio? [34 f] falso

(palabra palabra1 palabra2 palabra3 ...)

palabra palabra1 palabra2 Devuelve una palabra formada por la unión de las entradas.

Ejemplo: es palabra "camion "es camiones

Ejemplo: El siguiente programa sirve para formar el plural de una palabra, previamente introducida a través del teclado.
 para plural
 haz "nombre lp
 haz "ultimaletra ul :nombre
 si miembro? :ultimaletra [a e i o u] [
 haz "nombre palabra :nombre "s es :nombre alto]
 haz "nombre palabra :nombre "es es :nombre
 fin

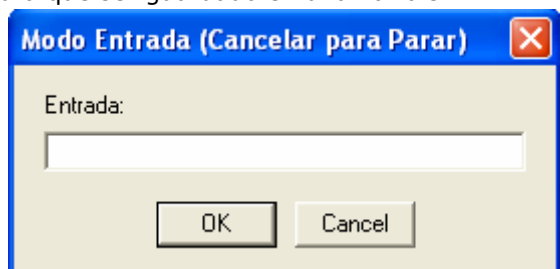
(lista cosa1 cosa2 cosa3 ...)

lista cosa1 cosa2 Devuelve una lista cuyos miembros son las entradas, que pueden ser cualquier objeto de Logo (palabra, variable, instrucción, otra lista,...).

leepalabra

lp Abre una ventana de la figura para que podamos introducir desde el teclado cualquier número o palabra que ,normalmente, tendrá que ser guardado en una variable.

Ejemplo: haz "solucion lp
 La palabra que escribamos en la ventana adjunta queda guardada en la variable "solucion.



leelista

ll Es similar a la instrucción leepalabra, que nos permite introducir datos desde el teclado. En este caso, lo que se lee es una lista, en la que sus elementos deben ir separados por espacios en blanco.

EL PROCEDIMIENTO INICIAL DEL AHORCADO.

```
para ahorcado
bp ot
haz "solucion lp ;se introduce la palabra y se almacena en "solucion
haz "numero (cuenta :solucion) ;obtenemos el número de letras de la palabra
ponpos [0 -20] gd 90
repite :numero [bl av 15 sl av 8] ;dibujamos tantas líneas como letras
ponpos [0 0]
rotula pri :solucion ;escribimos la primera letra sobre la primera línea
haz "f 0 ;la variable "f guardará el número de errores cometidos
ponfoco [Pantalla de MSWLogo] ponteclado[juega] ;cada vez que pulsemos una tecla
fin ;se ejecuta el procedimiento juega
```

Ya sólo nos queda realizar el procedimiento juega, que se encargará de comprobar si la tecla pulsada se corresponde con una letra de la palabra, si es así la escribirá encima de la línea correspondiente, si no irá dibujando la parte del ahorcado que se corresponda con el número de errores cometidos. Para ello, vamos a necesitar dos instrucciones nuevas:

sisino condición [instrucciones si][instrucciones no]

Si la condición es verdadera se ejecutan las instrucciones del primer corchete, si es falsa se ejecutan las instrucciones del segundo corchete.

Las condiciones pueden ser muy diversas:

```
:nota > 5
:nombre = "Antonio
pixel = [0 0 0]
.....
```

Ejemplo: El siguiente procedimiento nos pide la nota numérica de un alumno y nos responde con aprobado o suspenso, según que dicha nota sea mayor o menor que 5.

```
para calificación
bt es "|introduce la nota|
haz "nota lp
sisino :nota > 5 [es "suspenso][es "aprobado]
fin
```

cuentarepite

Sólo puede usarse dentro de un repite. Devuelve el número de repeticiones finalizadas, incluyendo la actual. Esto es, devuelve la primera vez 1; 2, la segunda vez, y así sucesivamente.

EL PROCEDIMIENTO JUEGA

```
para juega
haz "letra car lc ;guardamos en "letra la tecla pulsada
sisino miembro? :letra :solucion [acertamos][fallamos] ;si la letra es miembro de la
fin ;palabra que tenemos que adivinar acertamos, si no fallamos

para fallamos
haz "f :f+1 ;contamos el número de fallos
fallo :f ;dibujamos la parte correspondiente
fin
```

```
para acertamos
```

```
;------  
;en este procedimiento vamos comparando cada una de las letras de la palabra con la letra que hemos introducido  
;cuando hay coincidencia llevamos la tortuga sobre la línea que corresponde a su posición y la escribimos  
;------
```

```
repite (:numero - 1) [  
    si (elemento (cuentarepite +1) :solucion) = :letra [  
        ponpos lista (23*cuentarepite) 0  
        rotula :letra]  
    ]  
fin
```