

# Programación en Shell: Ciclo `while`

Moisés García Villanueva

Enero de 2018

## 1 Uso del ciclo `while` en el shell

Otra forma de implementar instrucciones repetitivas en el `shell` es utilizar la estructura `while`. Es posible convertir un ciclo `for` a un ciclo `while` y viceversa.

### 1.1 Estructura del ciclo `while`

Es necesario aprenderse la estructura o forma correcta de escribir un ciclo `while` en las instrucciones del `shell` para poder aplicarlo a los problemas que se plantean en los sistemas de cómputo.

```
while [ condición-de-fin-del-ciclo ]
do
    # Conjunto de instrucciones a repetir, incluso puede ser otro ciclo o bucle
    # instrucción para hacer que se logre cumplir con la condición de fin del ciclo
done
```

El siguiente código de ejemplo nos imprime en la salida estándar de la terminal las tablas de multiplicar 11 y 12.

```
X=1
while [ $X -le 10 ]
do
    res1='expr 11 \* $X'
    res2='expr 12 \* $X'
    echo -e "$X * 11 = $res1\t\t$X * 12 = $res2"
    X='expr $X + 1'
done
```

Escribir en un archivo de texto plano, con nombre `ejem-while.sh`, el código del ejemplo anterior y ejecutar en una terminal de comandos de Linux de la siguiente forma:

```
bash ejem-while.sh
```

### Ejercicios

1. Editar nuevamente el archivo `ejem-while.sh`, cambiar el valor 10 de la condición de fin del ciclo a 30 y ejecutar en la terminal el archivo.
2. Hacer que el incremento de la variable `X` sea de 2 en 2 o de 5 en 5.

### 1.2 Imprimir una imagen ppm que contenga un triángulo

Dibujar un triángulo rectángulo de un color de relleno azul y color de fondo verde. Ver la Figura 1 en donde se muestra una imagen cuadrada de 250 píxeles tanto de ancho como de alto, en la cual se encuentra circunscrito el triángulo.

La siguiente secuencia de actividades a realizar, permiten entender la forma en que se va construyendo la imagen, presentando los resultados parciales para un mejor entendimiento de la solución planteada.

1. Primero escribiremos en un archivo (`triangulo.sh`) el código para imprimir el encabezado del archivo ppm y un ciclo `while` que permite imprimir todos los datos del primer renglón de la imagen en el color verde.

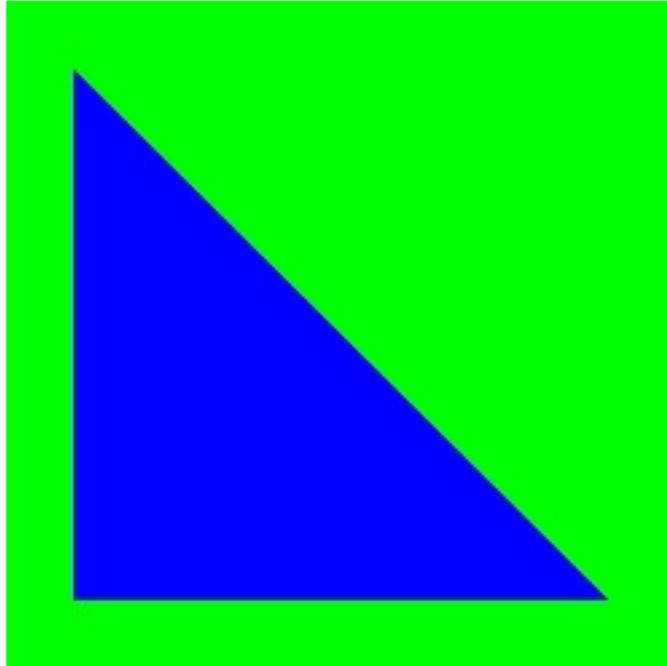


Figure 1: Triángulo dibujado en una imagen de 250 píxeles tanto de ancho como de alto.

```
# Se imprime la línea del encabezado del archivo imagen ppm
echo -e "P3 \n250 250\n255"
```

```
# Ciclo while para imprimir un renglón de color verde
p=0
while [ $p -lt 250 ]
do
    echo -n "0 255 0 "
    p='expr $p + 1'
done
```

Ejecutar en la terminal el archivo `triangulo.sh` de la siguiente forma: `bash triangulo.sh > tri.ppm; eog tri.ppm`

2. Para hacer que el triángulo quede centrado en la imagen, se requiere imprimir una cantidad de renglones del mismo color (verde en este ejemplo), por lo tanto se hace necesario repetir 25 veces el código que imprime el primer renglón, para ello se colocará el ciclo del código anterior dentro de otro ciclo que tenga una repetición de 25 veces; se imprime un salto de línea cada vez que se termina de imprimir un renglón de datos. Para lograrlo se edita y actualiza el código del archivo `triangulo.sh` de la siguiente forma:

```
# Imprimir el encabezado del archivo
echo -e "P3\n250 250\n255"
```

```
# Ciclo while para imprimir los primeros 25 renglones de color verde de la imagen
r=0
while [ $r -lt 25 ]
do
    # Ciclo while para imprimir un renglón de color verde
    p=0
    while [ $p -lt 250 ]
    do
        echo -n "0 255 0 "
        p='expr $p + 1'
    done
    echo ""
done
```

```

    r='expr $r + 1'
done

```

Para observar el resultado del script, ejecutar en la terminal de comandos de la siguiente forma:  
**bash triangulo.sh > tri.ppm; eog tri.ppm**  
 se observa en pantalla la imagen de la Figura 2.

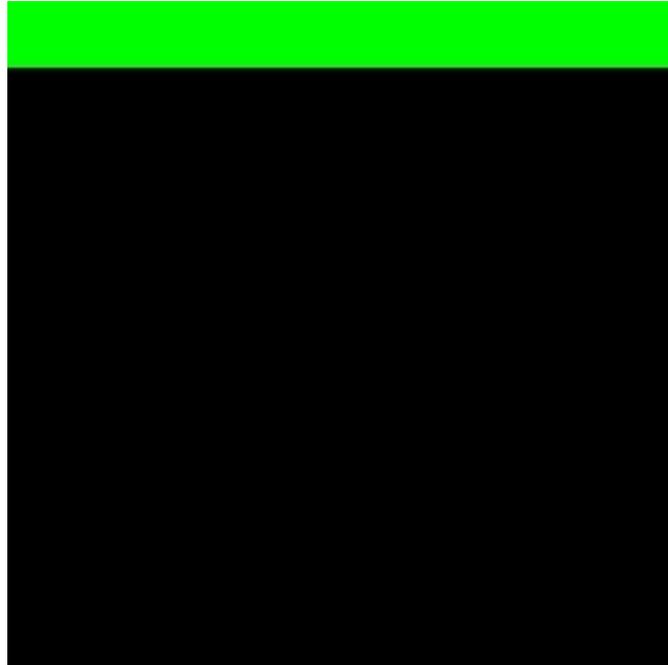


Figure 2: La imagen muestra los 25 primeros renglones de la imagen ppm cuyo fondo de color será verde, los cuales se generaron con los dos ciclos anidados que se presentan en el código del archivo `triangulo.sh`

- Para crear el triángulo rectángulo de la Figura 1, cada renglón en donde existe color azul debe iniciar en el píxel de la columna 25 (primer ciclo del renglón); Por cada renglón que se complete se debe ir incrementando en uno la cantidad de píxeles azules que se van coloreando en la imagen (segundo ciclo del renglón). Finalmente el resto de píxeles en el renglón deben pintarse del color de fondo de la imagen (tercer ciclo del renglón). Incluir el siguiente código al final del archivo `triangulo.sh` y ejecutarlo en la terminal de comandos para observar el resultado parcial de la imagen generada:

```
bash triangulo.sh > tri.ppm; eog tri.ppm
```

```

r=1 # Esta variable nos ayudará a incrementar la cantidad de píxeles azules
#   que se imprimen por cada renglón de datos
# Para imprimir un renglón de los que dibujan el triángulo, se utilizan
#   los siguientes tres ciclos while
i=0
while [ $i -lt 25 ]
do
    echo -n "0 255 0 "
    i='expr $i + 1'
done
colsTriangulo=0
while [ $colsTriangulo -lt $r ]
do
    echo -n "0 0 255 "
    colsTriangulo='expr $colsTriangulo + 1'
done
while [ $colsTriangulo -lt 225 ]
do
    echo -n "0 255 0 "

```

```

    colsTriangulo='expr $colsTriangulo + 1'
done

```

La imagen resultante incluirá un nuevo renglón mayormente de color verde con un único píxel de color azul en la columna 25 del renglón 26. Entonces para generar el triángulo rectángulo se requieren generar 200 renglones de este tipo, para ello se coloca DENTRO de un nuevo ciclo `while` el código que contiene los tres ciclos en serie para generar un renglón de datos de nuestra imagen. Substituya el siguiente código en el archivo `triangulo.sh`, en donde se incluye un nuevo ciclo `while` para repetir las instrucciones de escritura de datos 200 veces.

```

# Para crear el triángulo, primero se imprime una porción de columnas de color verde,
# la parte del triángulo y después el color de fondo de la imagen nuevamente
r=1
while [ $r -lt 200 ]
do
    i=0
    while [ $i -lt 25 ]
    do
        echo -n "0 255 0 "
        i='expr $i + 1'
    done
    colsTriangulo=0
    while [ $colsTriangulo -lt $r ]
    do
        echo -n "0 0 255 "
        colsTriangulo='expr $colsTriangulo + 1'
    done
    while [ $colsTriangulo -lt 225 ]
    do
        echo -n "0 255 0 "
        colsTriangulo='expr $colsTriangulo + 1'
    done
    r='expr $r + 1'
done

```

El resultado parcial que se debe observar al ejecutar el script de la siguiente forma:

```
bash triangulo.sh > tri.ppm; eog tri.ppm
```

se encuentra en la Figura 3.

- El paso final es repetir el código que se utilizó en el paso número 2, es decir, queremos generar 25 renglones del color de fondo y que se escriban al final de nuestro archivo de imagen. Copiar el siguiente código al final del archivo `triangulo.sh` y ejecutarlo en la terminal, el resultado final será la imagen de la Figura 1.

```

r=0
while [ $r -lt 25 ]
do
    # Ciclo while para imprimir un renglón de color verde
    p=0
    while [ $p -lt 250 ]
    do
        echo -n "0 255 0 "
        p='expr $p + 1'
    done
    echo ""
    r='expr $r + 1'
done

```

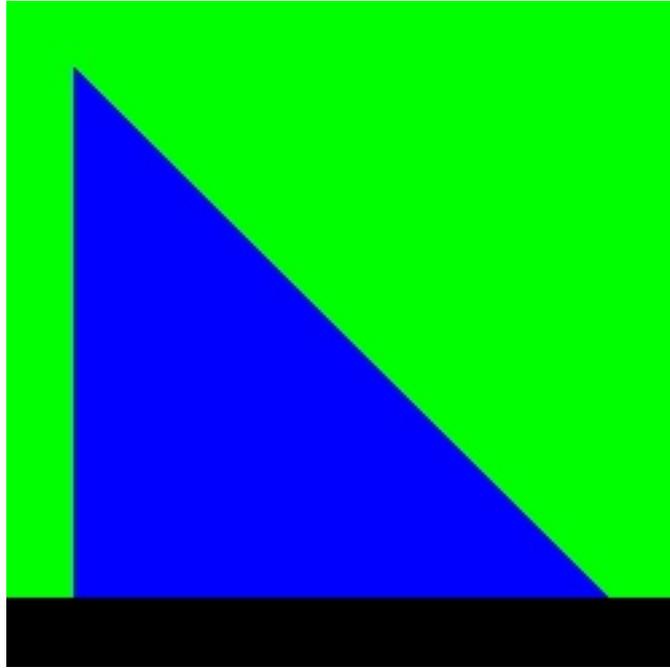


Figure 3: Resultado de generar 225 renglones de la imagen ppm, en donde se observa el triángulo rectángulo completo de color azul y con el color de fondo verde.

### 1.3 Ejercicios

1. Dividir en dos ciclos el ciclo del paso 1, de tal forma que se dibujen dos triángulos en los primeros 25 renglones, la condicional del primer ciclo debe ajustarse a la expresión:  $Renglon \times 10$ , es decir por cada renglón que se dibuja se multiplica por 10 el número de renglón. Proporcionar diferentes colores a los rectángulos, como se ve en la Figura 4.

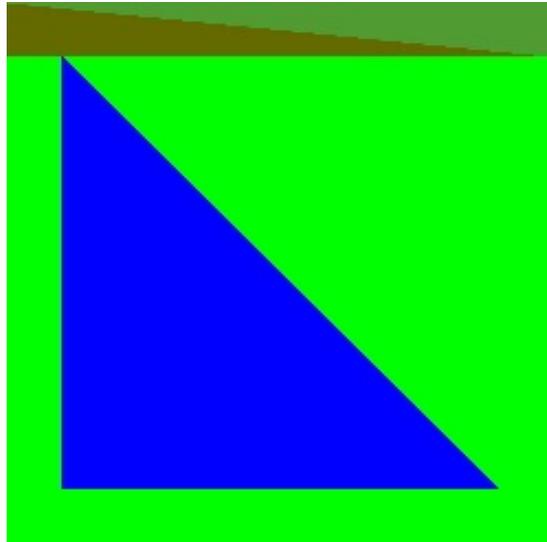


Figure 4: Para generar los dos triángulo superiores, dividir el primer ciclo en dos y modificar la condicional de fin de ciclo a que se incremente 10 veces el número de renglón que se imprime.

2. Generar la imagen de la Figura 5, con los colores a su elección. Observar que los triángulos en la parte superior e inferior se encuentran invertidos.
3. Generar por separado cada una de las imágenes en la Figura 6, con los colores a su elección.

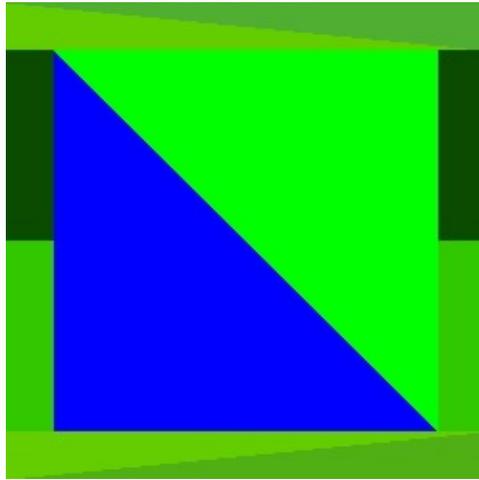


Figure 5: Construir con instrucciones repetitivas la imagen que se muestra con sus colores a elegir.

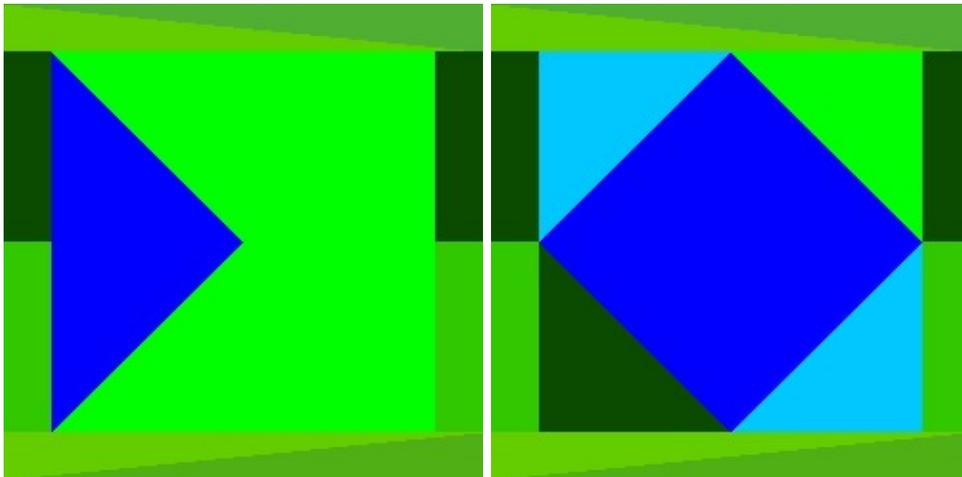


Figure 6: Construir con instrucciones repetitivas por separado, cada uno de los cuadros de la imagen que se muestra, utilizar los colores de su elección.