

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier  
mirror_ob.select = 0  
= bpy.context.selected_obj  
data.objects[one.name].se  
print("please select exactly
```

```
--- OPERATOR CLASSES ---
```

```
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"
```

```
context):  
context.active_object is not
```

Variables tipos de datos operadores y pseudocódigo

ING. CARLOS H. RUEDA C.

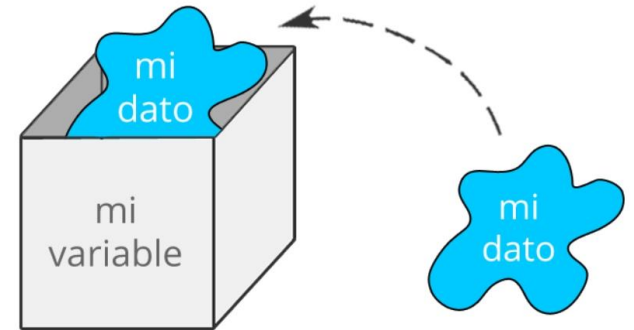


CONTENIDO

1. Variables y asignación de valores
2. Tipos de datos
3. Operadores
4. Pseudocódigo

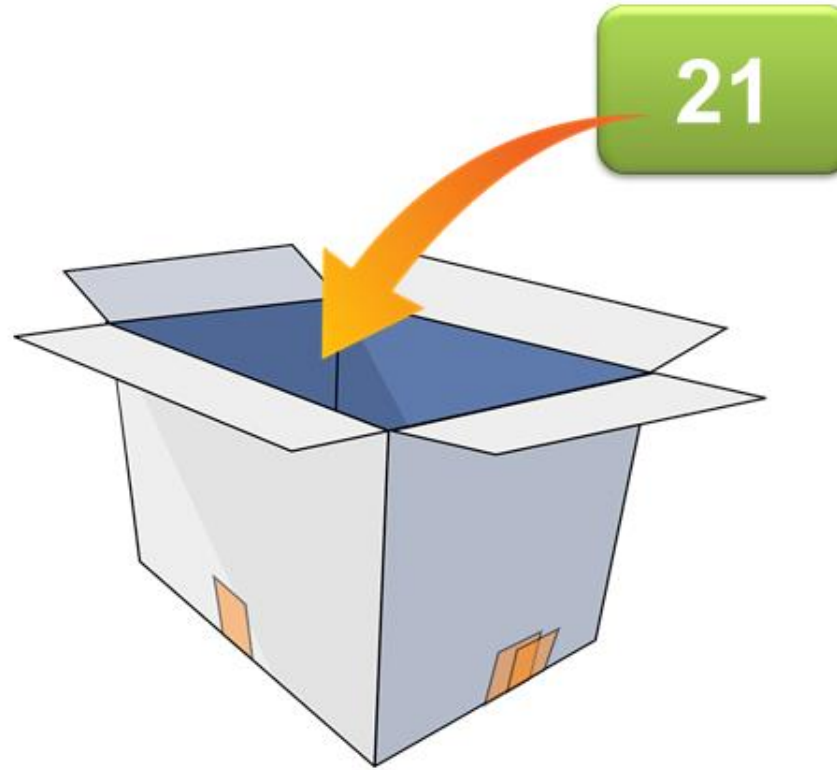
1. Variables y asignación de valores

Constantemente se necesita disponer de algún dato o alguna información para realizar una operación o un proceso, para esto sirven las variables.



1. Variables y asignación de valores

Son contenedores donde se almacenan diferentes tipos de datos que pueden cambiar de valor, de ahí sale su nombre.



1. Variables y asignación de valores

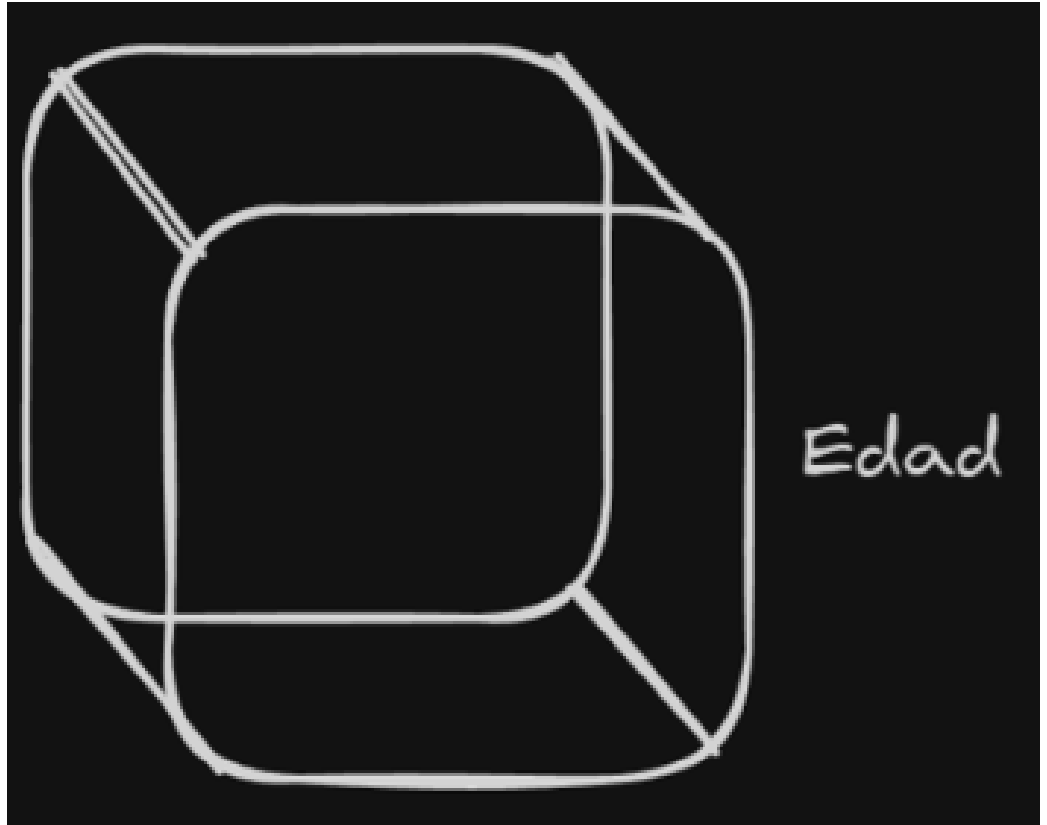


Se recomienda dar nombres semánticos, con semántica se refiere a que con solo leer la variable ya se sepa que hay en esa variable

```
name = 'brayan'  
age = 23  
ocupation = 'dev'
```

1. Variables y asignación de valores

Creamos una
cajita la cual
llamamos edad





1. Variables y asignación de valores

A esa cajita

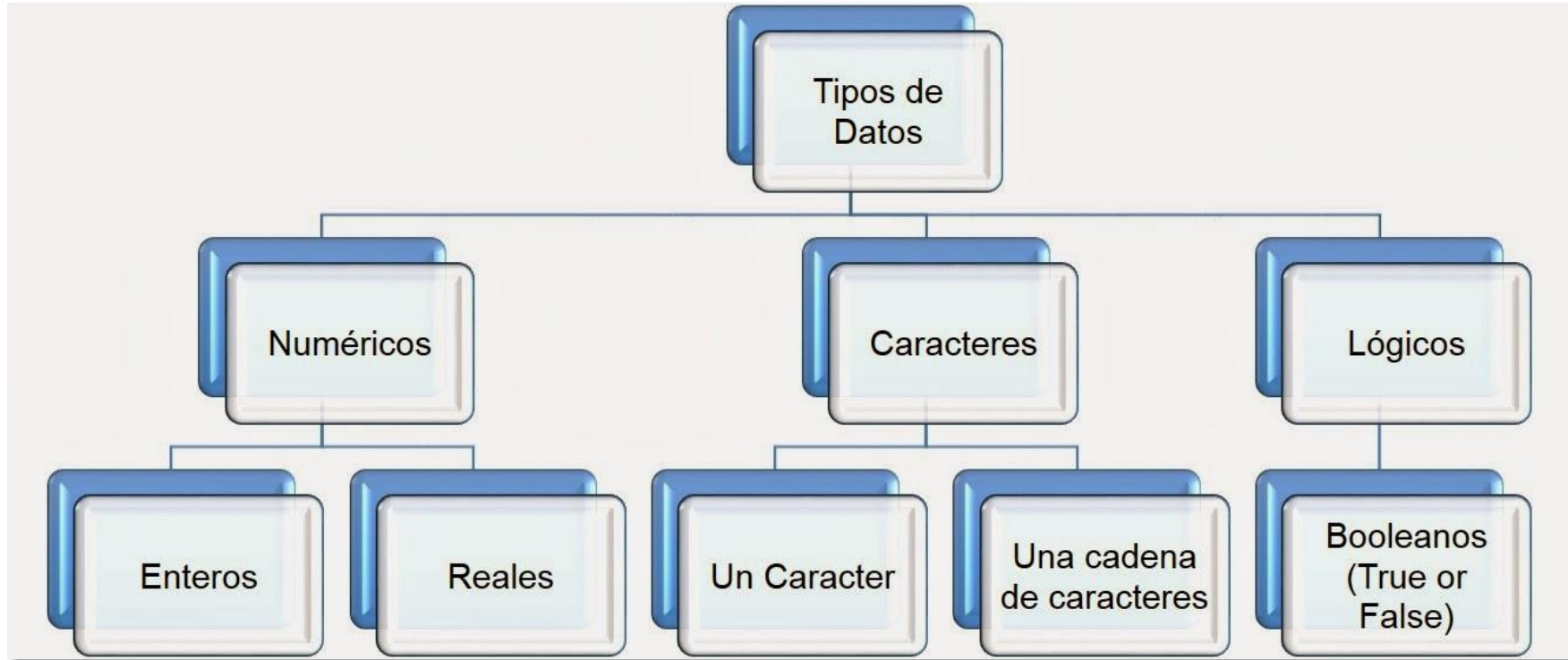
le vamos a
asignar el
valor de 36.

2. Tipos de datos

a variable, se le asocia un identificador, es decir, un nombre, con un tipo de almacenamiento cuya forma determina la visibilidad y existencia de la variable.



2. Tipos de datos



2. Tipos de datos

Numéricos enteros

1, 2, 3, 4, 5

2. Tipos de datos

Numéricos flotantes (reales)

1.2, 1.567, 5.34, -23.21

2. Tipos de datos

Cadena de caracteres o strings

"1234", "hola como estan todos",
"!@#\$%^&^\$%h hola \$#@\$%@"

2. Tipos de datos

Booleanos

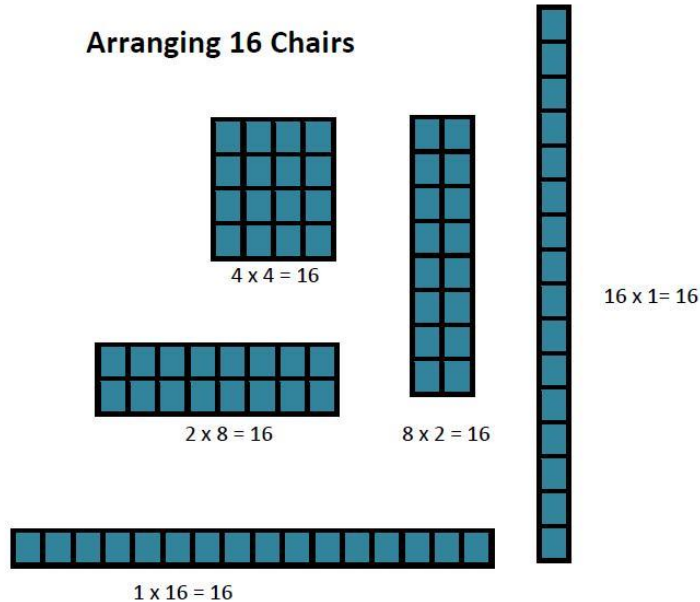
"true, false"

2. Tipos de datos

ARREGLOS

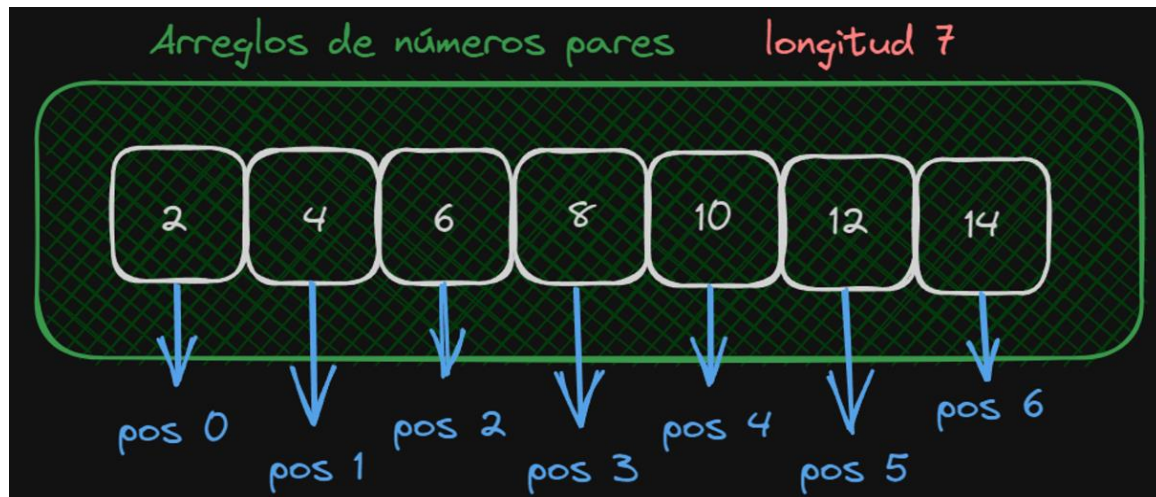
Los arreglos son estructuras de datos que permiten almacenar múltiples valores en una sola variable.

Arranging 16 Chairs



2. Tipos de datos

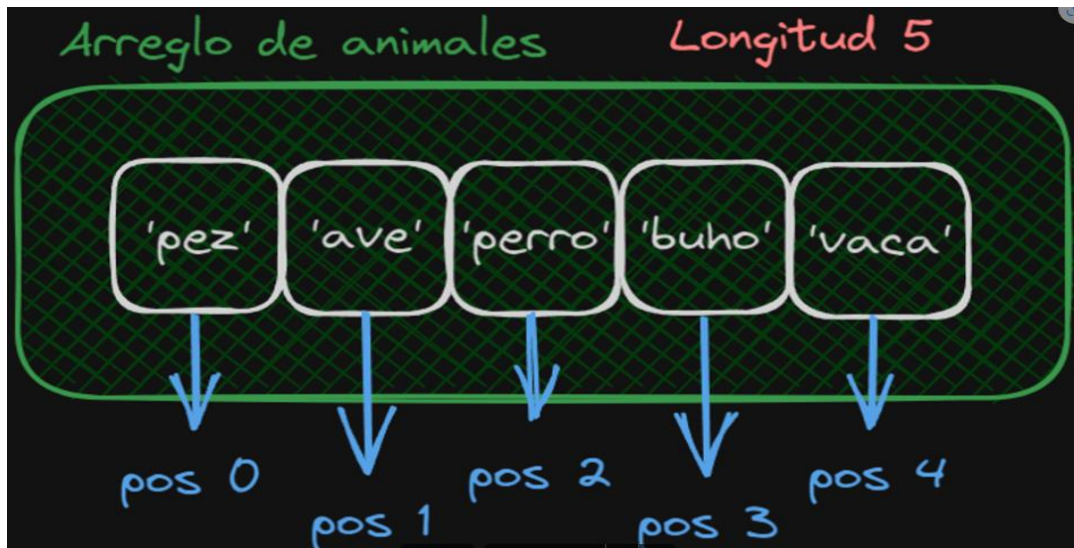
ARREGLOS



```
arr[0] // 2  
arr[6] // 14
```

2. Tipos de datos

ARREGLOS



```
arr[1] // "ave"  
arr[4] // "vaca"
```


3. Operadores



ARITMÉTICAS



RELACIONALES



LÓGICAS



Operadores

3. Operadores aritméticos

Operador	Ejemplo	Solución ejemplo
+	Suma dos valores.	Si tengo 5 manzanas y le sumo 3 manzanas, obtengo 8 manzanas.
-	Resta dos valores.	Ejemplo: Si tengo 5 manzanas y le resto 3 manzanas, obtengo 2 manzanas.
*	Multiplica dos valores.	Ejemplo: Si tengo 5 manzanas y las multiplico por 3, obtengo 15 manzanas.
/	Divide dos valores.	Ejemplo: Si tengo 10 manzanas y las divido por 2, obtengo 5 manzanas.
%	Devuelve el resto de una división.	Ejemplo: Si tengo 10 manzanas y las divido por 3, obtengo 3 manzanas y me sobran 1 .

3. Operadores de comparación

Operador	Descripción	Ejemplo
Igual (==)	Comprueba si dos valores son iguales.	Si la edad es igual a 18, entonces eres mayor de edad.
Diferente (!=)	Comprueba si dos valores no son iguales.	Si el estado civil es diferente de "casado", entonces eres soltero.
Mayor que (>)	Comprueba si el valor de la izquierda es mayor que el de la derecha.	Si la puntuación es mayor que 90, entonces obtienes una A.
Menor que (<)	Comprueba si el valor de la izquierda es menor que el de la derecha.	Si la cantidad de productos es menor que 10, entonces aplica un descuento.
Mayor o igual que (>=)	Comprueba si el valor de la izquierda es mayor o igual que el de la derecha.	Si el tiempo de entrega es mayor o igual a 3 días, entonces el envío es estándar.
Menor o igual que (<=)	Comprueba si el valor de la izquierda es menor o igual que el de la derecha.	Si el precio es menor o igual que \$50, entonces es asequible.

3. Operadores lógicos

Operador	Descripción	Ejemplo
AND (Y)	Devuelve "verdadero" solo si ambas condiciones son verdaderas.	Si es de día Y tengo un paraguas, entonces puedo salir con el paraguas.
OR (O)	Devuelve "verdadero" si al menos una de las condiciones es verdadera.	Si está lloviendo O si tengo un paraguas, entonces puedo salir con el paraguas.
NOT (NO)	Invierte el valor de verdad de una condición.	Si NO está lloviendo, entonces no necesito un paraguas.

4. Pseudocódigo

Ventajas

1. Mayor Eficiencia
2. Facilidad de lectura
3. Mayor Flexibilidad
4. Fomenta la Colaboración
5. Gestión Local



4. Pseudocódigo

Cómo escribir pseudocódigo

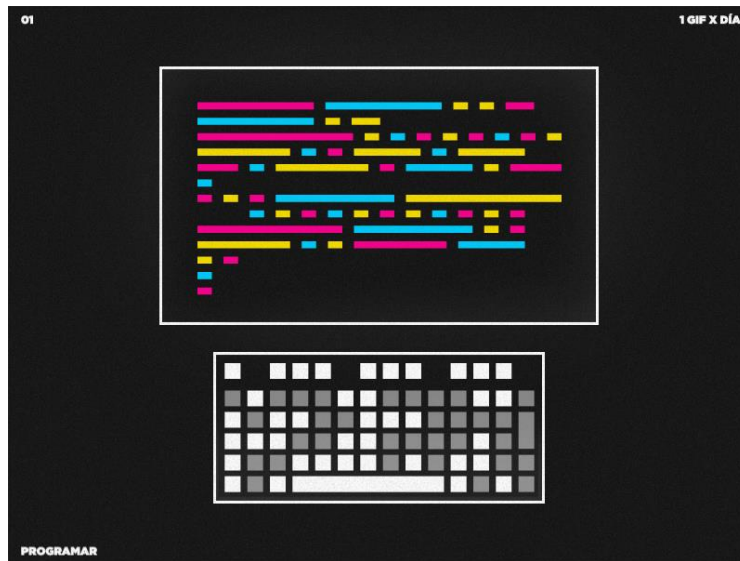
1. Abre tu editor de texto
2. Define tu objetivo
3. Divide en partes
4. Establece pasos
5. Utiliza sangrías
6. Realiza pruebas



4. Pseudocódigo

Ejemplo 1

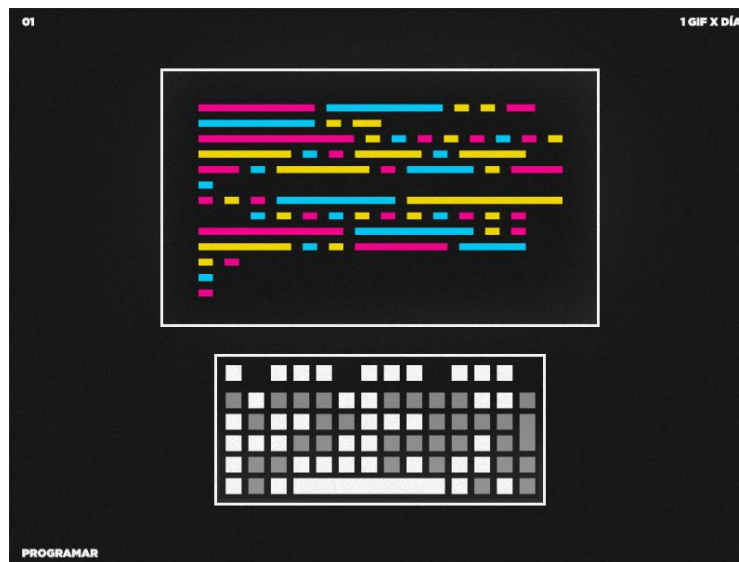
Pedir un número y mostrarlo



4. Pseudocódigo

Ejemplo 2

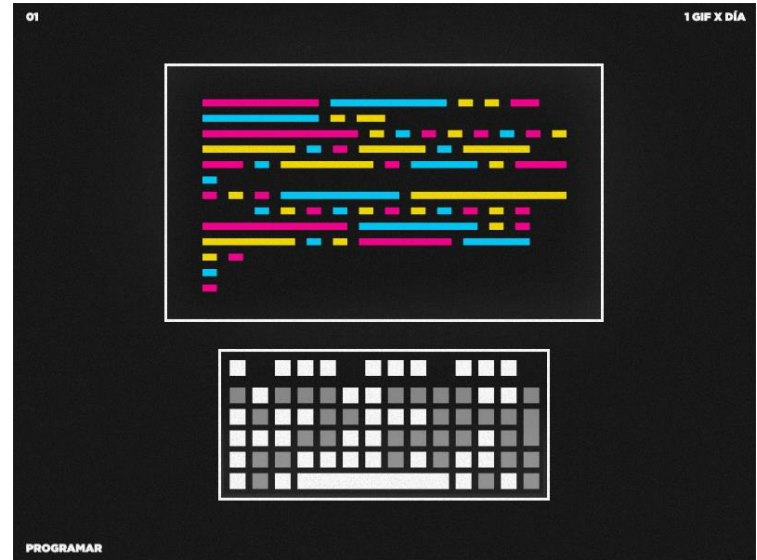
Suma de dos números



4. Pseudocódigo

Ejemplo 3

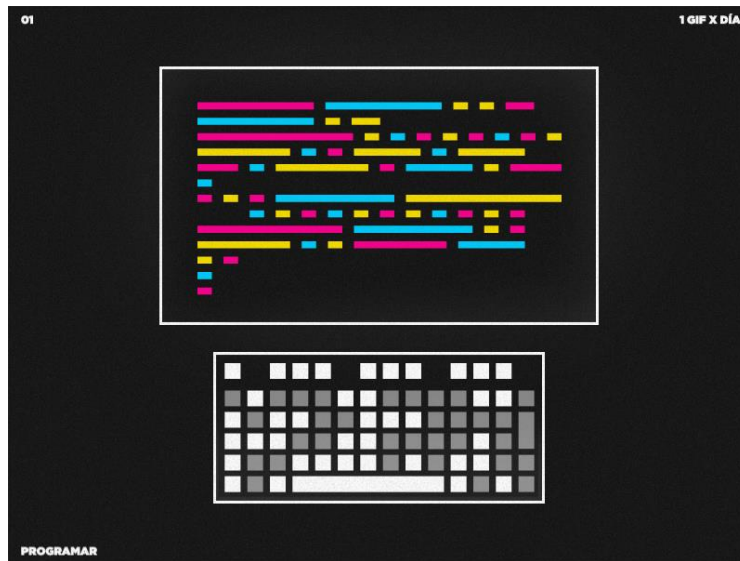
Convertir grados Celsius a Fahrenheit.



4. Pseudocódigo

Ejercicio 1

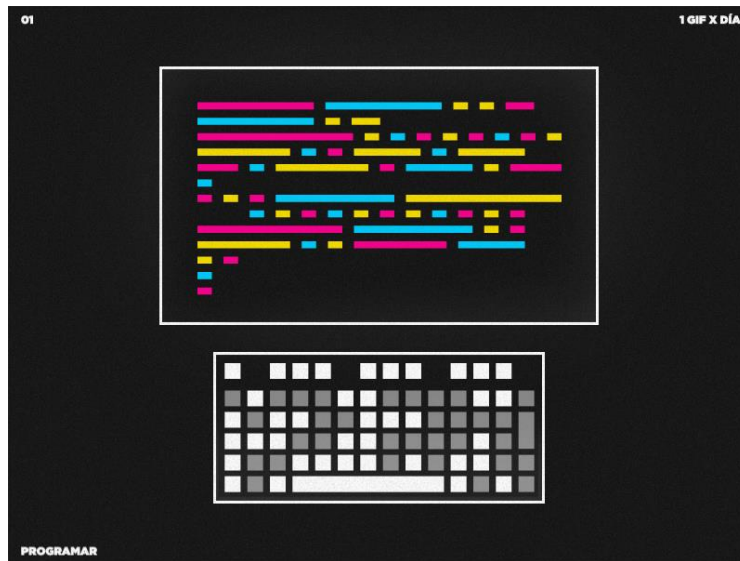
Calcular el área de un triángulo.



4. Pseudocódigo

Ejercicio 2

Calcular el área de un círculo.



4. Pseudocódigo

Ejercicio 3

Calcular la cantidad de dígitos de un número.

