

# Logo

Fundamentos de Programación

# Agenda

- Repeticiones que Varían
- Comandos Personalizados
- Parámetros
- Aleatoriedad



# Contar Repeticiones

- El comando **repcount** nos indica cuantas repeticiones se encuentran realizando dentro de un comando **repeat**.
- Esto nos ayuda a realizar ligeros ajustes dentro del *lazo* de repetición.

```
repeat 100 [ fd repcount * 2 rt 90 ]
```

¿Qué se Dibuja?

# Comandos Propios/Personalizados

- A pesar de que no hemos visto un listado de comandos completos, podemos intuir que:
  - El diccionario de comandos existentes debe de ser limitado.
  - Nos gustaría que la tortuga aprenda a hacer dibujos más complejos, sin tener que indicarle siempre todos los pasos.
- Se pueden definir nuevos comandos.
- En la jerga tradicional de programación, se suele llamar a estos comandos propios **funciones**.

# Funciones en Logo

- Las palabras claves son **to** y **end**.
  - Entre estos comandos ubicaremos aquellos que deseamos se ordenen cuando *llamemos* a la función.

```
to cuadrado
  repeat 4 [ fd 100 rt 90 ]
end

to triangulo.equilatero
  rt 30
  repeat 3 [ fd 100 rt 120 ]
  lt 30
end
```

```
to casa
  cuadrado
  fd 100
  triangulo.equilatero
  bk 100
end
```

# Parámetros

- ¿Qué tienen de limitante nuestras funciones?
  - Sería bueno poder hacer casas de distintos tamaños...
- Los parámetros son datos que cambian el comportamiento de la función.
  - forward **100**
  - right **90**
- ¿Por qué no,
  - casa 100?
  - triangulo 90?

# Definiendo Parámetros en Logo

- El nombre del parámetro lo decidimos nosotros pero debe de comenzar con un :
- El parámetro se debe ubicar justo después del nombre de la función (pueden haber más de un parámetro por función):
  - to cuadrado :lado
  - to rectangulo :base :altura
- Dentro de la función se los llama por su nombre completo:

```
to cuadrado :lado
  repeat 4 [ fd :lado rt 90 ]
end
```

# Aleatoriedad

- La palabra clave es **random**.
- **random** acepta un parámetro que dictamina el rango de números que pueden salir como resultado de su operación.
  - `random 10`, genera un número entero aleatorio entre 0 y 9.
  - `random 100`, genera un número entero aleatorio entre 0 y 99.
- ¿Qué podemos hacer con esto?



# ... Fractales

