

Sentencias de Repetición

Fundamentos de Programación

Agenda

- Sentencias de Repetición
 - Tipos de Repeticiones
 - Por cantidad de veces
 - Todos los elementos de una serie
 - Condicionales
 - Contadores y Acumuladores
- Resolución de Problemas en Clase

Sentencias de Repetición

- También conocidas como sentencias iterativas, sirven para ejecutar un mismo bloque de código varias veces seguidas.
 - En Logo utilizabamos el comando repeat y la misma combinación de comandos ocurría tantas veces como el argumento que indicáramos.

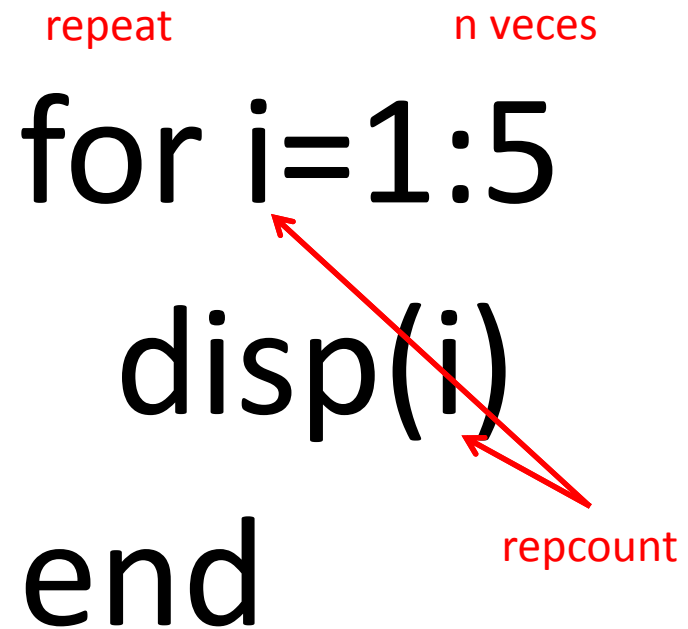
for, while

TIPOS DE REPETICIONES

Repetición por Cantidad de Veces

- La sentencia de control **for** nos ayuda a repetir un mismo bloque de código N veces.
 - Encontraremos inicialmente los mismos elementos que tenía el repeat en Logo:
 - Un valor límite para la iteración.
 - Un **contador** (número que va avanzando acorde a la cantidad de veces iterada).

```
repeat          n veces
for i=1:5
  disp(i)
end
repcount
```



Acumuladores

- Los acumuladores son variables que deseamos se modifiquen durante la ejecución de todas las iteraciones de un lazo.
- Usualmente:
 - Se inicializan a un valor antes de entrar al lazo.
 - Se verifican al finalizar todas las iteraciones.

```
sum=0
```

```
for i=1:100
```

```
    sum = sum + i
```

```
end
```

```
disp(sum)
```

(Una) Aplicación en Matemáticas

$$\sum_{i=1}^{100} i$$

$$\sum_{i=1}^n i^2 + 5$$

$$\sum_{i=1}^{100} \sum_{j=1}^i j * i^2$$

for anidados

Los valores, condiciones, Incrementos o decremento, entre otros variarán dependiendo del ejercicio que se intente solucionar.

La variable índice del for más externo cambia más lento.

for (**i = 1 : 1 : n**)

for (**j = 1 : 1 : m**)

La variable índice del for más interno cambia más rápido.

$$\sum_{i=1}^{100} \sum_{j=1}^i j * i^2$$

Repetición por Elementos

- El argumento del **for** es en realidad un vector.

```
Startup execution:
  loading initial environment

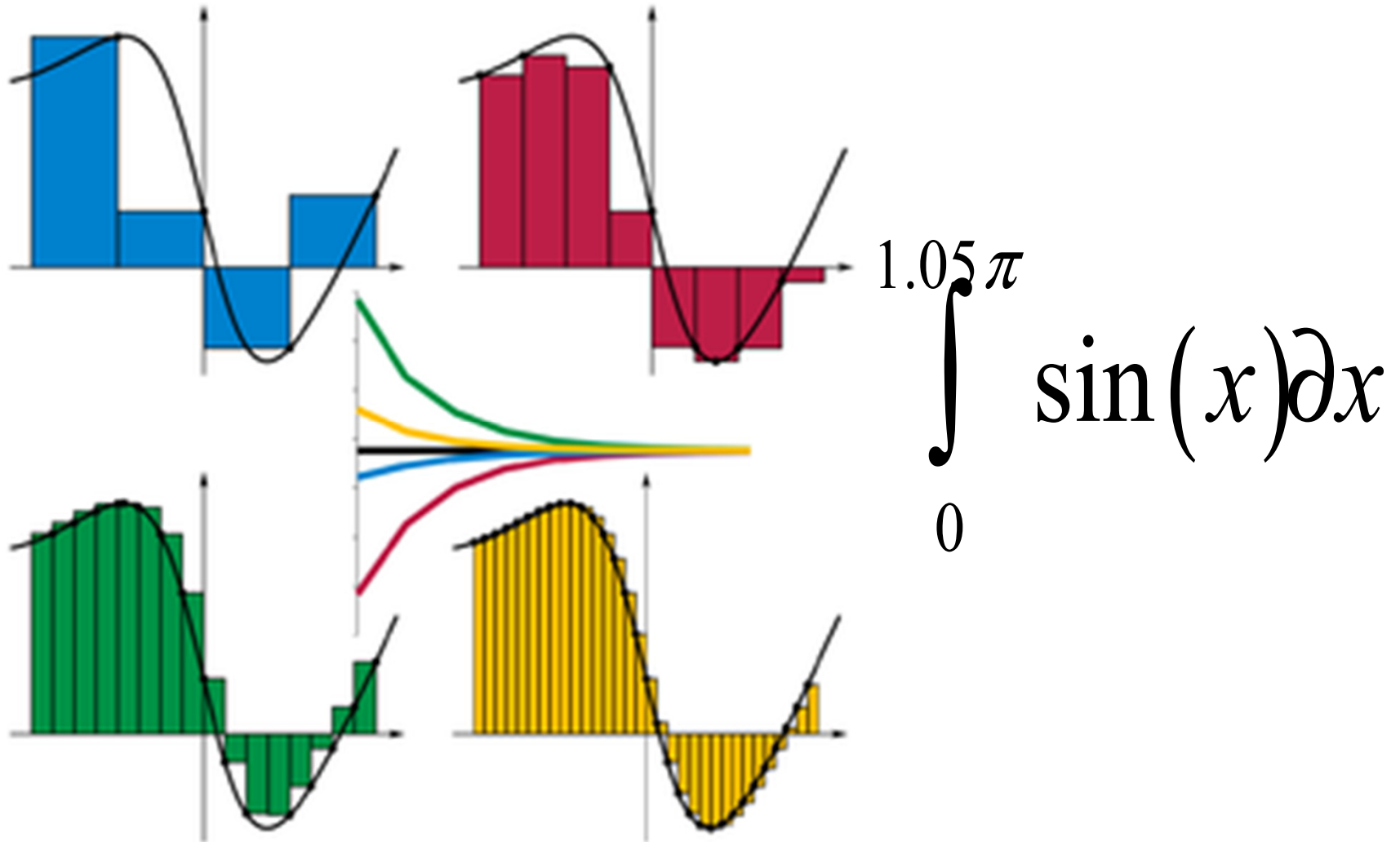
-->i=1:5
i =

     1.     2.     3.     4.     5.

-->
```

- La repetición mediante **for** es entonces una forma de iterar sobre todos los elementos de un vector.

Más Matemáticas



mprintf

Fase de salida:

-Para imprimir un mensaje que muestre la parte entera de un dato se incluye en el mensaje **%i**, así mismo para imprimir un mensaje que dependa de una variable real se incluye en el mensaje **%f** y luego se especifica la variable que se va a imprimir

```
suma=0
n=input("Indique el numero de notas a
ingresar: ")
for nota_i=1:1:n
    mprintf("Ingrese la nota número %i:
",nota_i)
    nota=input("Nota:")
    suma=suma+nota
end
mprintf("Promedio = %f",suma/n)
```

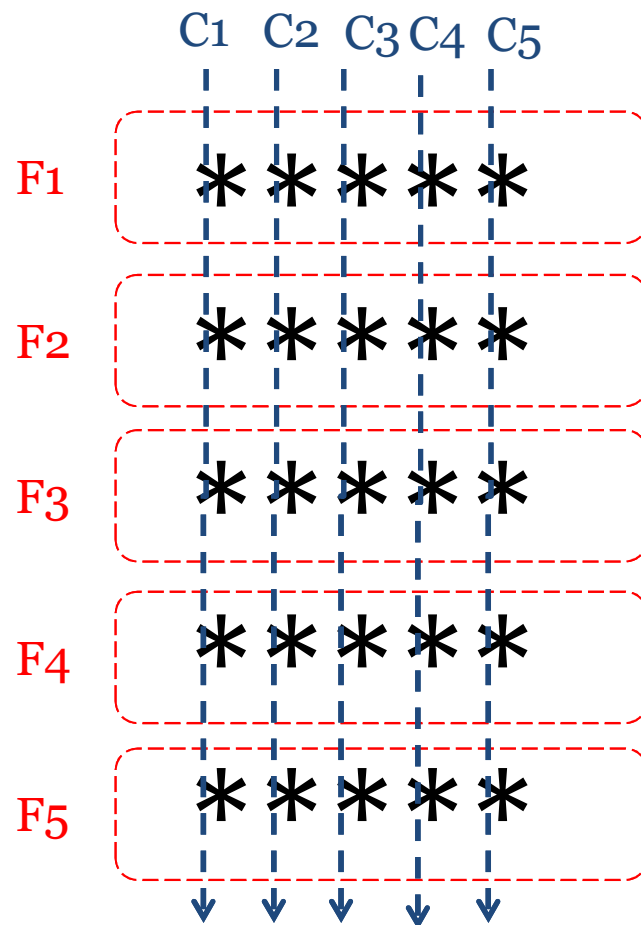
Ejercicios en Clase

Escriba un programa que muestre por pantalla:

- Un cuadrado formado por **asteriscos** de n filas y n columnas.
- Un rectángulo formado por asteriscos de n filas y m columnas.
- Un cuadrado formado por asteriscos cuya diagonal izquierda esté formada por espacios en blanco.
- El ejercicio anterior, cambiando el sentido de la diagonal.
- Los dos ejercicios anteriores juntos.
- Un triángulo formado por **asteriscos** de n filas.

NOTA: Para la realización de estos ejercicios considere que los valores de n y m son positivos y son ingresados por teclado.

Representación de caracteres



Un cuadrado representado por:

- 5 caracteres de ancho
- 5 caracteres de alto

Como podemos generarlo?

Cada fila: es una línea separada con un salto de línea “\n”

Estrategia:

Son 5 líneas y cada línea tiene 5 caracteres.

¿Cómo se puede implementar con la sentencia For?

Conceptos Básicos

Repetición Indefinida

- No se conoce el número de veces que se realizará la repetición.
- Se utiliza un dato "Centinela", la cual indica el "final de la ejecución de la repetición". *Por ejemplo:* presione 0 para finalizar el programa.

Repetición Condicionada

- Un nuevo comando, **while**, es útil para realizar repeticiones basadas en una condición lógica.

```
b=0
while(b==0)
    b=input("Ingrese un número diferente de 0: ")
end
```

mientras b siga siendo 0, continuaremos pidiendo un nuevo valor para b

- El bloque se repite siempre y cuando la condición sea **verdadera**.

Problemas

- Para una sumatoria desde **1** hasta **n**, ¿Cuál es el **n** más grande posible sin que la sumatoria sea mayor a **100**?
- Calcular el 20º término de la serie de fibonacci.

Conclusiones

- Las estructuras iterativas son útiles para repetir en secuencia un bloque de código.
- Existe dos tipos de estructuras de repetición en Scilab:
 - El for, que itera sobre los elementos de un vector.
 - El while, que repite siempre y cuando se cumpla (sea verdadera) una condición lógica determinada.
- La aplicación de las estructuras de control es sumamente importante en la programación.