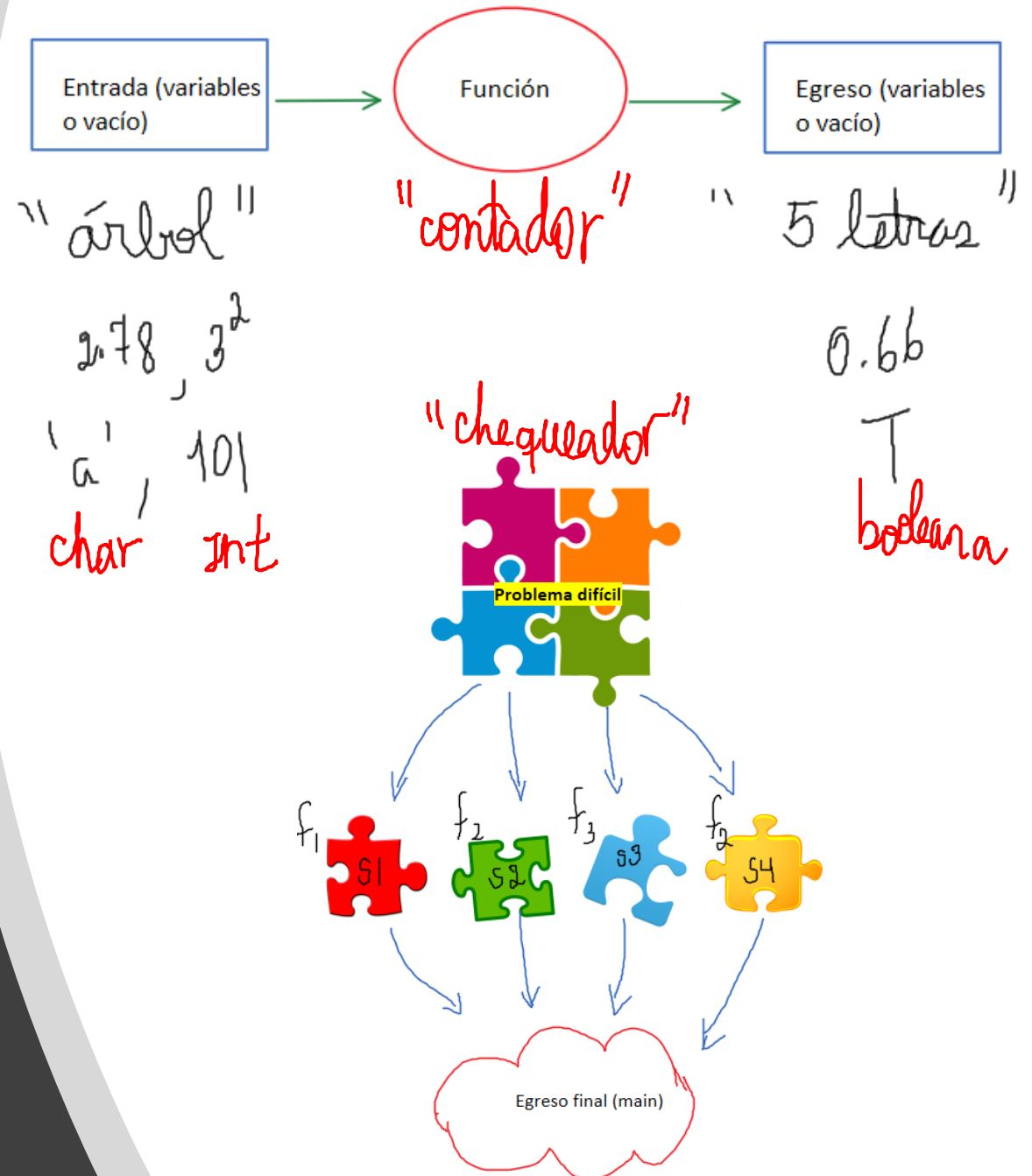


Capítulo II.I - Funciones

Función

- Transformación de entradas para devolverme un egreso/salida
- P.C: Saber cómo y cuándo definir funciones para descomponer un problema en sub-problemas
- Funciones pueden ser reusadas y simplifican el código



Ejemplo

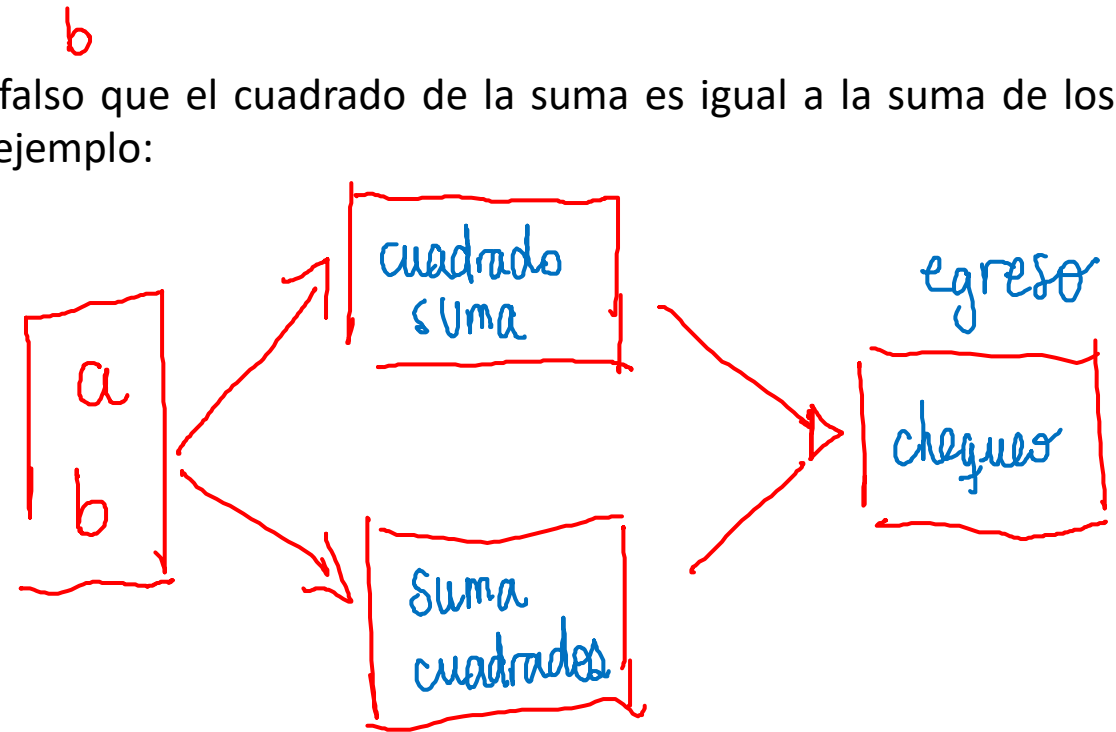
- Chequeo de igualdad

- Supongamos que queremos chequear si es verdad o falso que el cuadrado de la suma es igual a la suma de los cuadrados con respecto a un par de enteros a y b . Por ejemplo:

- $a = 18, b = 19$;

- $a = 2, b = 2$;

- $a = 3, b = 0$.



¿qué tal si queremos saber si el cubo de la suma es igual a la suma de los cubos para los enteros mencionados?

`constructor` `modificador` `tipo_var_egresado` `nombre_funcion` (`tipo_var ingresos`) {
 líneas de código;
 return *egreso* }

`public` `static` `double` `miFuncion` (`tipo_var nombre_var, ...`) {
 líneas de código;
 return *egreso*;
 }



Si tienes la curiosidad de explorar cuál es la diferencia entre funciones estáticas y no-estáticas puedes visitar: <https://www.geeksforgeeks.org/difference-between-static-and-non-static-method-in-java/>

`def` `nombre_fun` (`ingresos`):
 líneas de código
 return *egreso*

`def` `miFuncion` (`ingreso`):
 líneas de código;
 return *egreso*;



Ejercicios

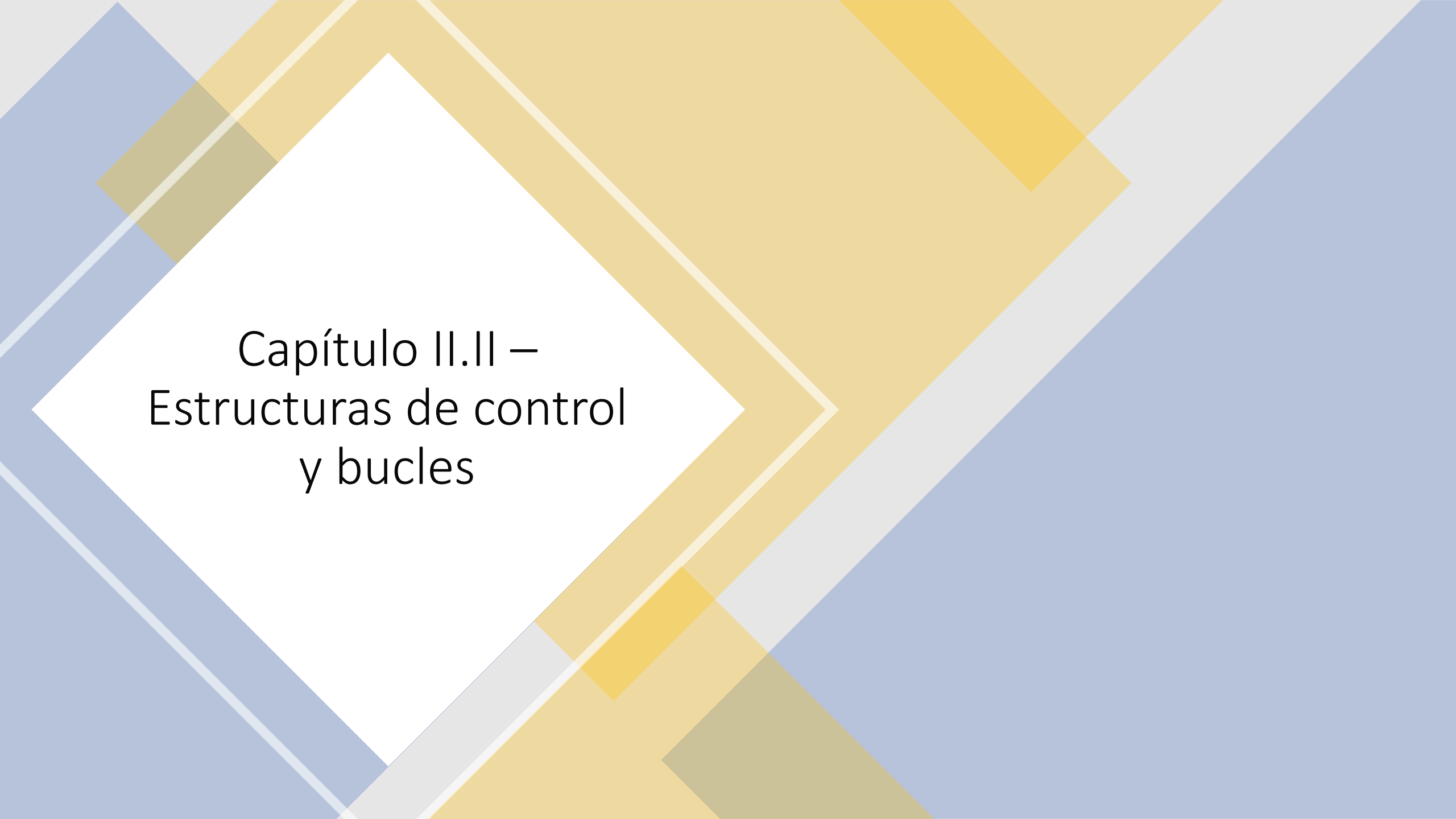
1. Define una función que chequea si un entero es divisible para 2 (usa módulo %)
2. Define una función que dado un decimal que equivale a la temperatura en Farenheit, esta lo convierte en Celsius.
3. Define una función que en primer lugar imprime una introducción al programa (tú decides qué mensaje imprimir). Este programa calcula el promedio de 5 decimales.
4. Define una función que toma una cadena como entrada e imprime los versos de la canción de cumpleaños con el nombre indicado por la cadena.
5. Define una función que toma un entero n y obtiene el término en la posición n de una sucesión Fibonacci.
6. Define una función que dado un entero no negativo m imprime la siguiente estructura piramidal:
1
22
333
4444
....
mmmmm..mm
7. Dado una lista de números $a = [0, 1, 2, \dots, 10]$ y $b = [0, 1, 2, \dots, 10]$, define una función que me dice qué pares de números (a, b) respetan la siguiente igualdad $a = 2b + 1$

Resumen:

- Una función es parte integral del arte programar: saber programar es saber definir funciones
- Una función generaliza procesos y simplifica código
- Varias funciones ayudan a resolver subproblemas de un problema general.

Bibliografía:

- Textos:
 - Capítulo 6.2 en adelante de Python Programming por John M. Zelle en <http://ce.sharif.edu/courses/97-98/1/ce153-12/resources/root/Text%20Books/An%20Introduction%20to%20Computer%20Science.pdf>
 - Capítulo 2 de Introduction to Programming Using Java en <http://math.hws.edu/javanotes/index.html>
- Música de fondo:

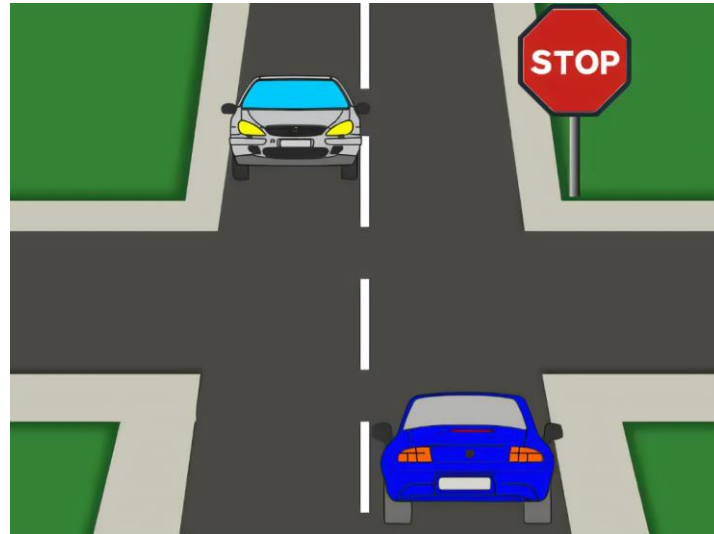


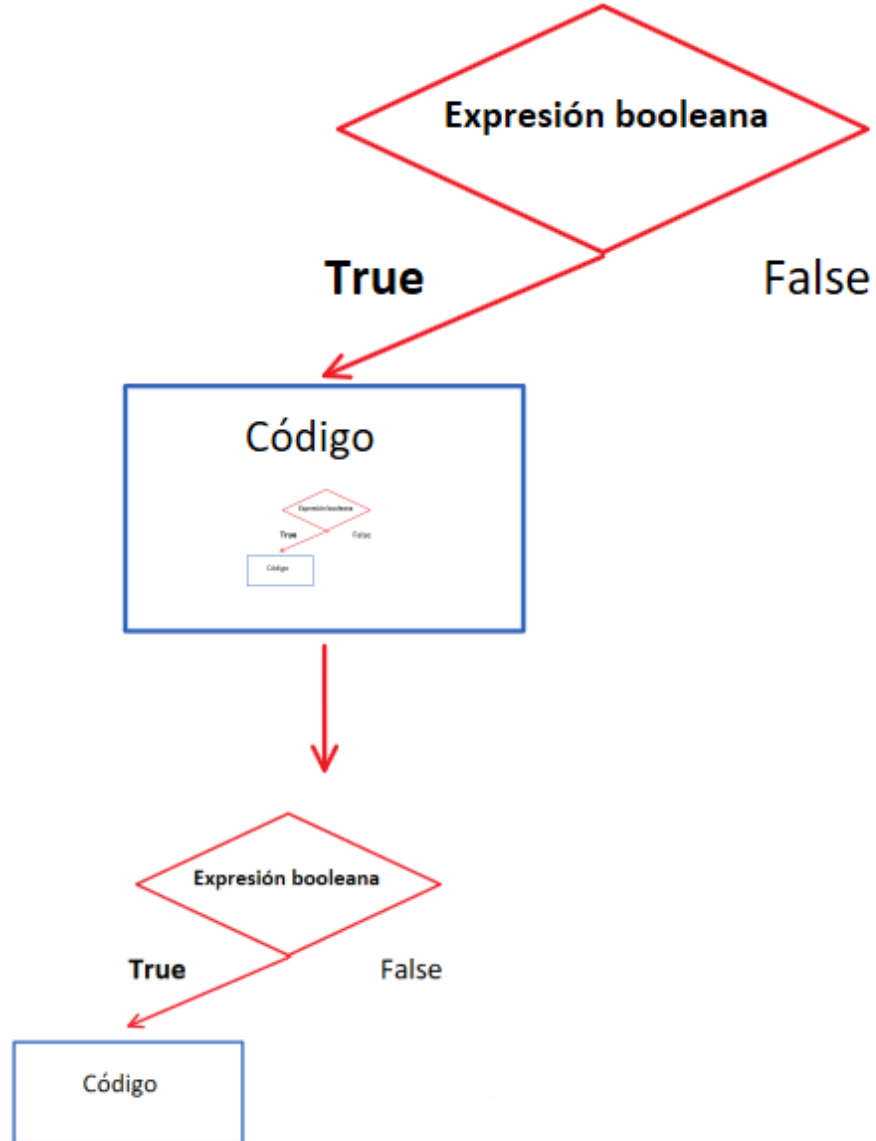
Capítulo II.II – Estructuras de control y bucles

Estructuras de control (if...else if...else)



- El código es ejecutado de manera descendiente
- Las estructuras de control (puntos de parada) deciden qué líneas de código son ejecutadas dependiendo de la situación





```
if exp_booleana :  
    # primera condición  
elif exp_booleana:  
    if exp_booleana:  
        # condición anidada  
else:  
    # tercera condición
```



```
if (exp_booleana) {  
    return ;  
}  
else if (exp_booleana) {  
    if (exp_booleana) {  
        return ;  
    }  
}  
else {  
    return ;  
}
```



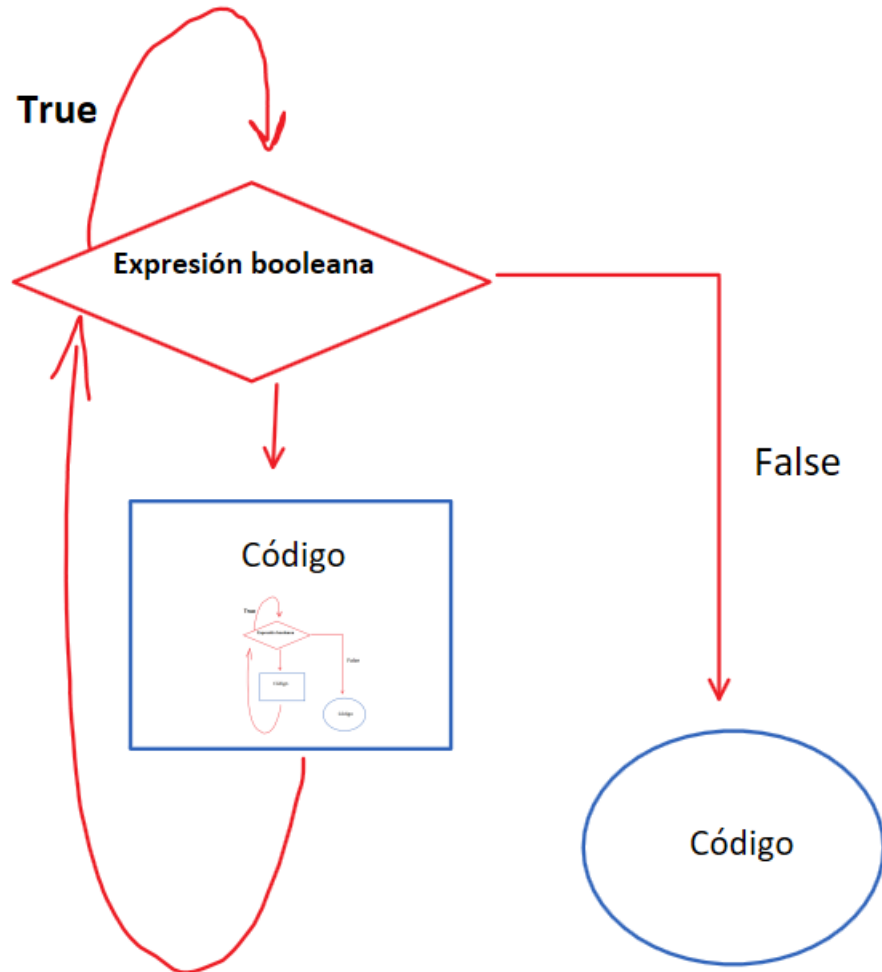
Ejemplo

- **Chequeo**
- Supongamos que queremos chequear si un número es impar, par o el único número par primo.

$$n \% 2 == 0$$

Bucles mientras (while-loops)

- Los bucles mientras son usados usualmente cuando no sabemos precisamente cuántas veces repetimos un bloque de código



```
i = 0
while i < 10:
    print("Python es lo mejor")
    i = i + 1
```



```
int contador = 0;
while (contador < 10) {
    System.out.println("Java es lo mejor");
    contador = contador + 1;
}
```

¿Qué pasaría si no escribiésemos la línea
`contador = contador + 1;`?

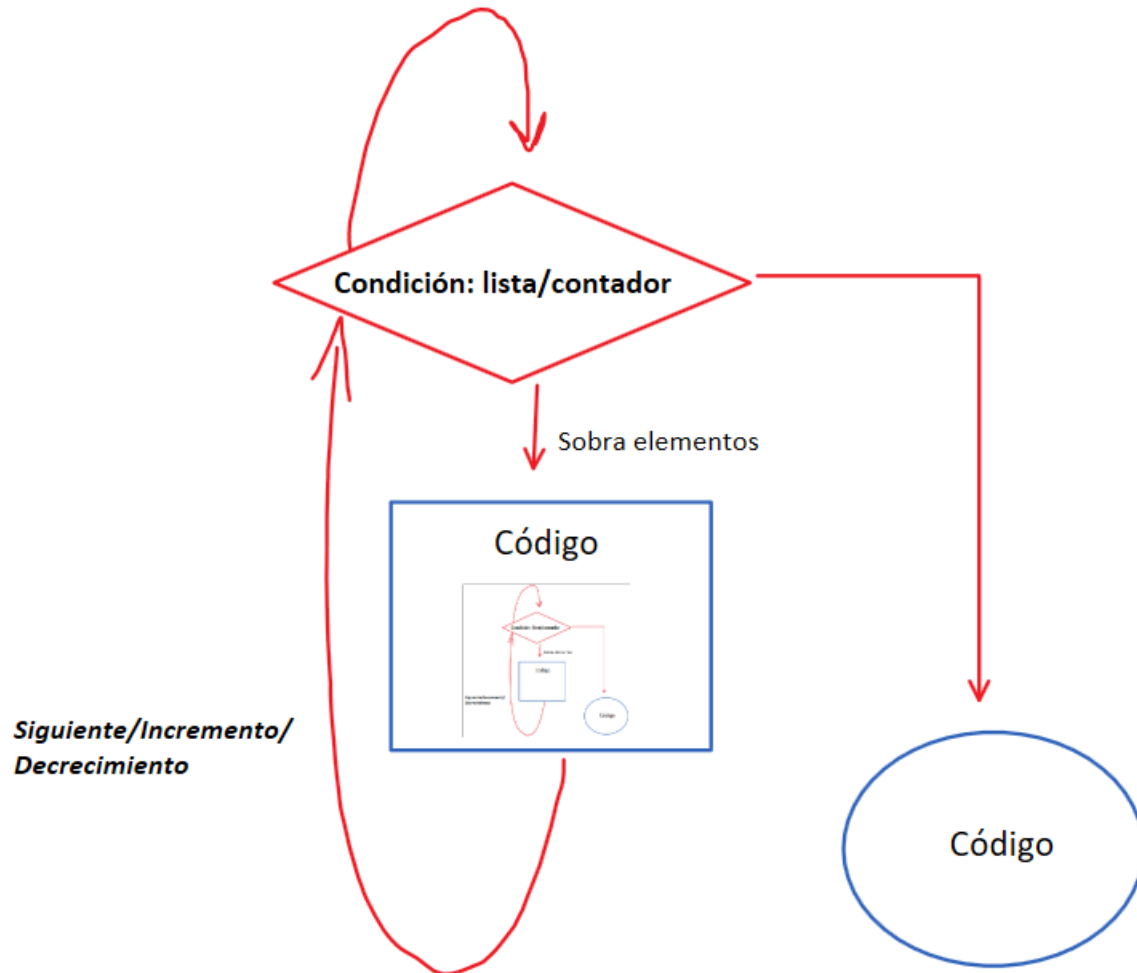


https://m.sohu.com/a/160736214_392831?_f=m-article_29_feeds_8



Bucles por (for-loops)

- Itera elementos de una lista o emula un contador/temporizador
- Debido a que las listas tienen tamaños fijos, los bucles por son usados cuando sí sabemos precisamente cuántas veces repetimos un bloque de código



```
for i in range(inicio, final, paso):  
    print(i ** 2)
```



```
int i;  
for (i = 0; i < 10; i++) {  
    System.out.println(i * i);  
}
```



El código será ejecutado
por $\frac{final - inicio}{paso}$ veces.

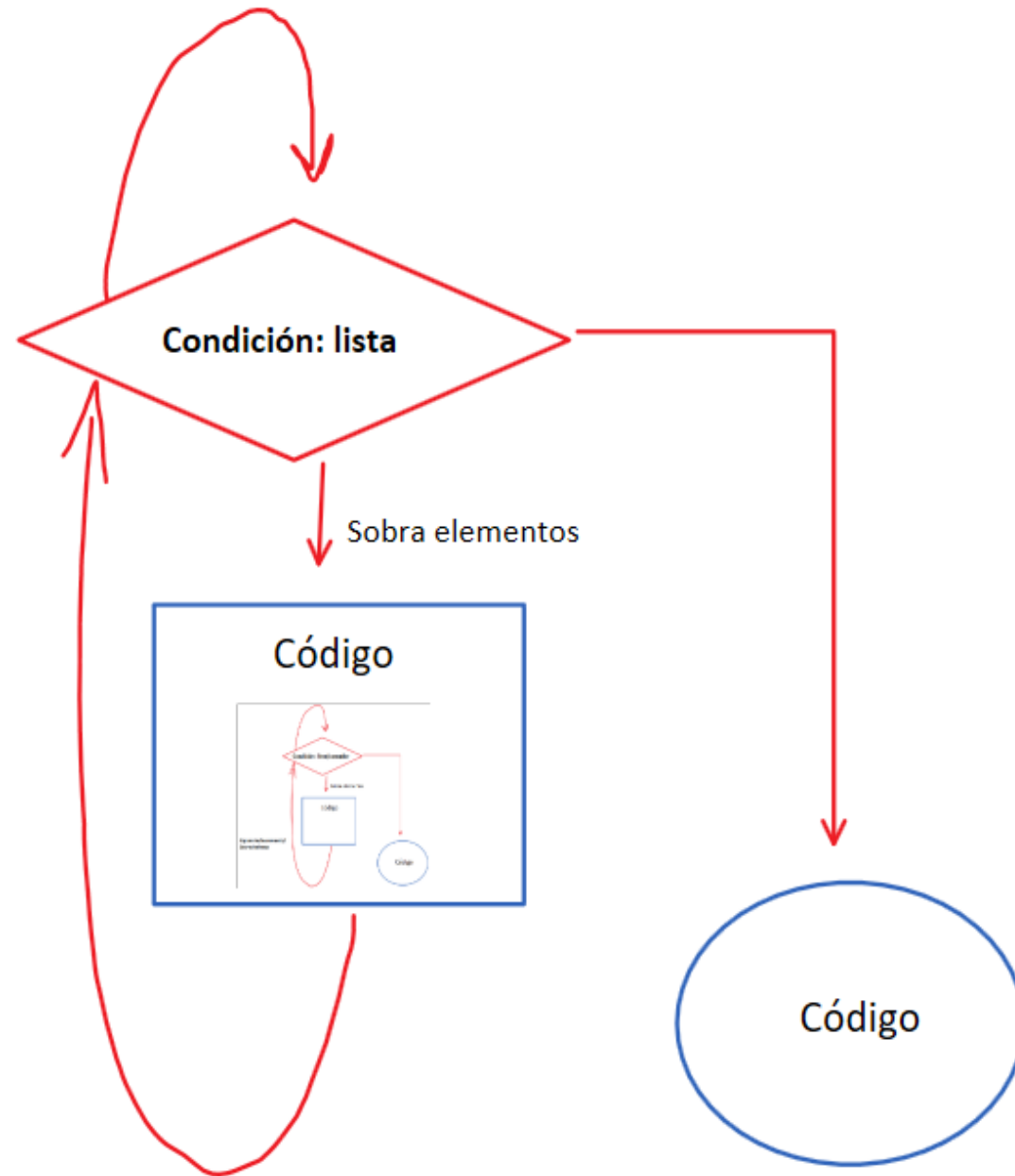
Bucles po

- En un bucle p
lista, por ende

```
miLista = [5, 1, 9,  
for num in miLista:  
    print(num ** 2)
```



Siguiente



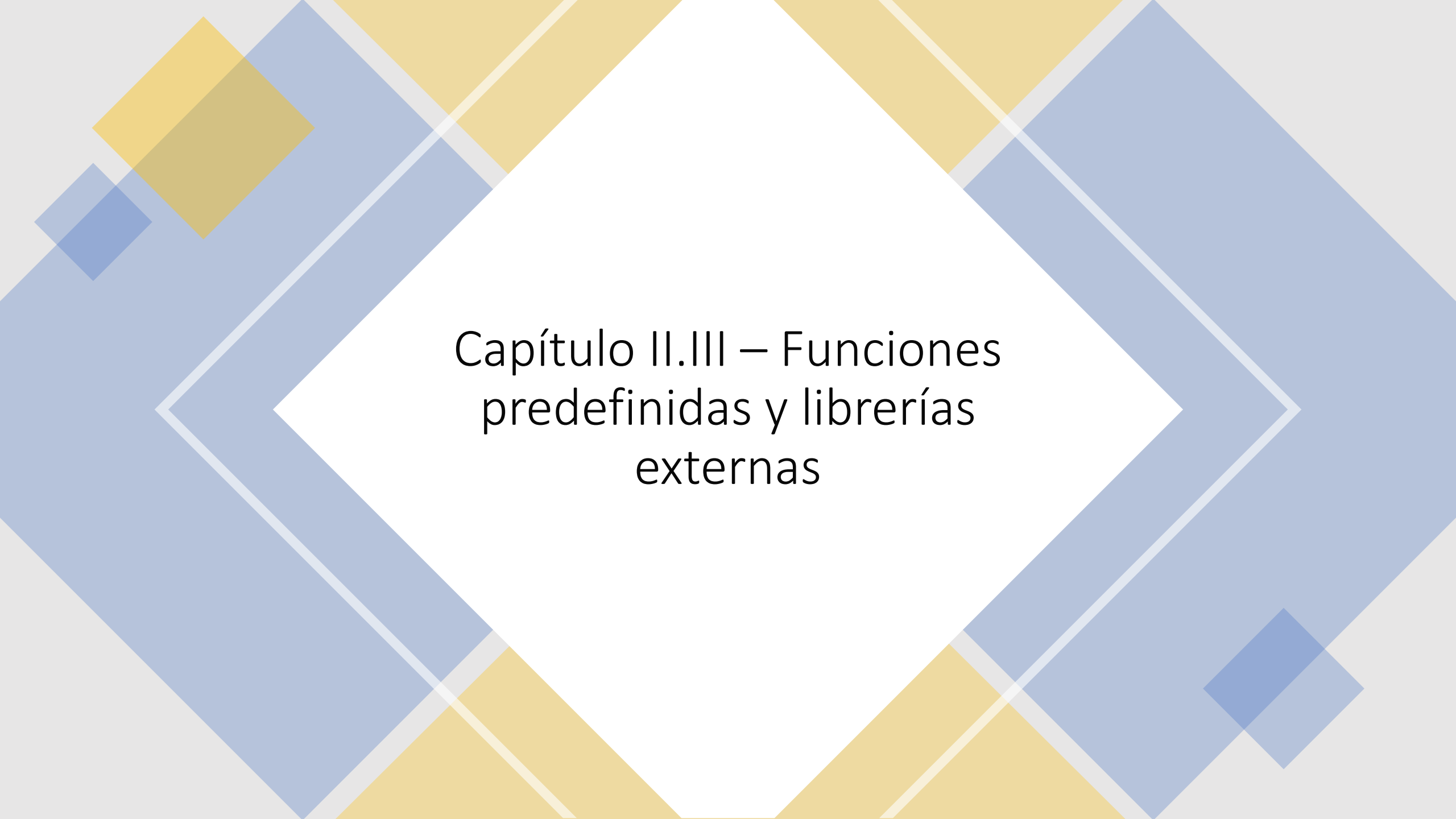
-loops)

dentro de una

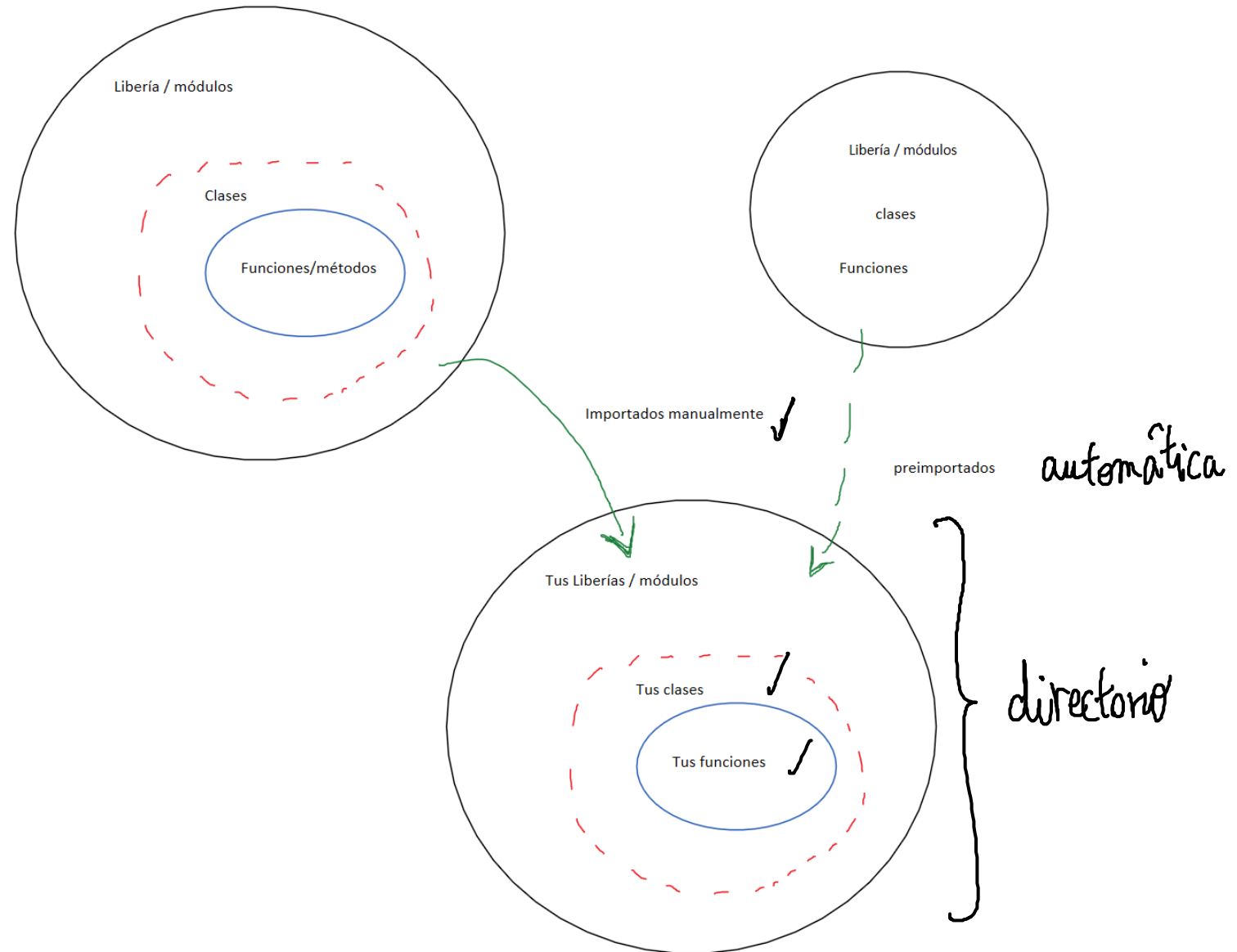
```
{"C", "Python", "Java"};  
{
```



[I@7637f22



Capítulo II.III – Funciones predefinidas y librerías externas



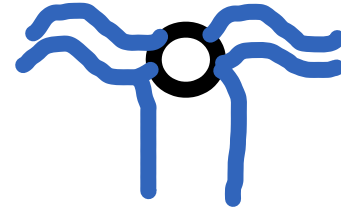
Funciones preimportadas en Java



- `int[] miArreglo = new int[] {12, -9, 5, 0, 8, 11, 10};`
- `miArreglo.clone();`
- `miArreglo[1]` // accede al elemento en la posición indicada
- `miArreglo.length;` // devuelve el tamaño del arreglo
- `String chiste = "Python es mejor que Java ";`
- `chiste.length();` // devuelve el tamaño
- `chiste.charAt(5);` // devuelve el índice del carácter
- `chiste.toLowerCase();` // convierte todo en minúscula
- `chiste.toUpperCase();` // convierte todo en mayúscula;
- `char[] miListaCaracteres = oracion.toCharArray();` // Lo convierte a arreglo de caracteres



Clases preimportadas en Java



- Math

- System

- String

- ...

`println`

Funciones preimportadas en Python

- `miLista = [8.9, 1.3, 5.5, -2.4, 3.1, 9.8]`
- `len(miLista)` # devuelve la cantidad de elementos en la lista
- `sum(miLista)` # devuelve la suma de los números en la lista
- `miLista[5]` # devuelve el elemento en la posición indicado por el índice
- `miLista.append(10)` # agrega un nuevo número a la lista
- `miLista.pop()` # remueve el primer elemento, y decrece el tamaño de la lista
- `miLista.index(8.9)` # devuelve el índice de un número
- `listCaracteres = "Vamos a programar"` # lista de caracteres
- `listCaracteres[8]` # lo mismo que el anterior
- `len(listCaracteres)` # devuelve el tamaño de la lista
- `listCaracteres.index("a")` # devuelve el índice del carácter indicado
- `listCaracteres.upper()` # convierte en mayúscula todos los caracteres
- `listCaracteres.lower()` # convierte en minúscula



Problema ejemplar

- Dado un entero no negativo m , genera e imprime una lista que contiene m números aleatorios y calcula su suma. El rango de números aleatorios es entre 1 y 6.

¿cómo lo esquematizarías?

Importación manual de librerías externas

- Si las clases y sus funciones predefinidas no son suficiente para resolver un problema, podemos **importarlas**:

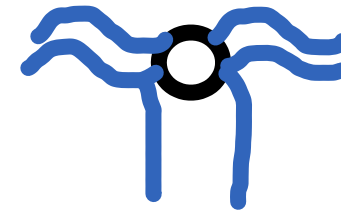
```
import random as rd  
from random import randint  
from random import randint as ri
```



```
import java.util.Random;
```



Librerías útiles (para Python)



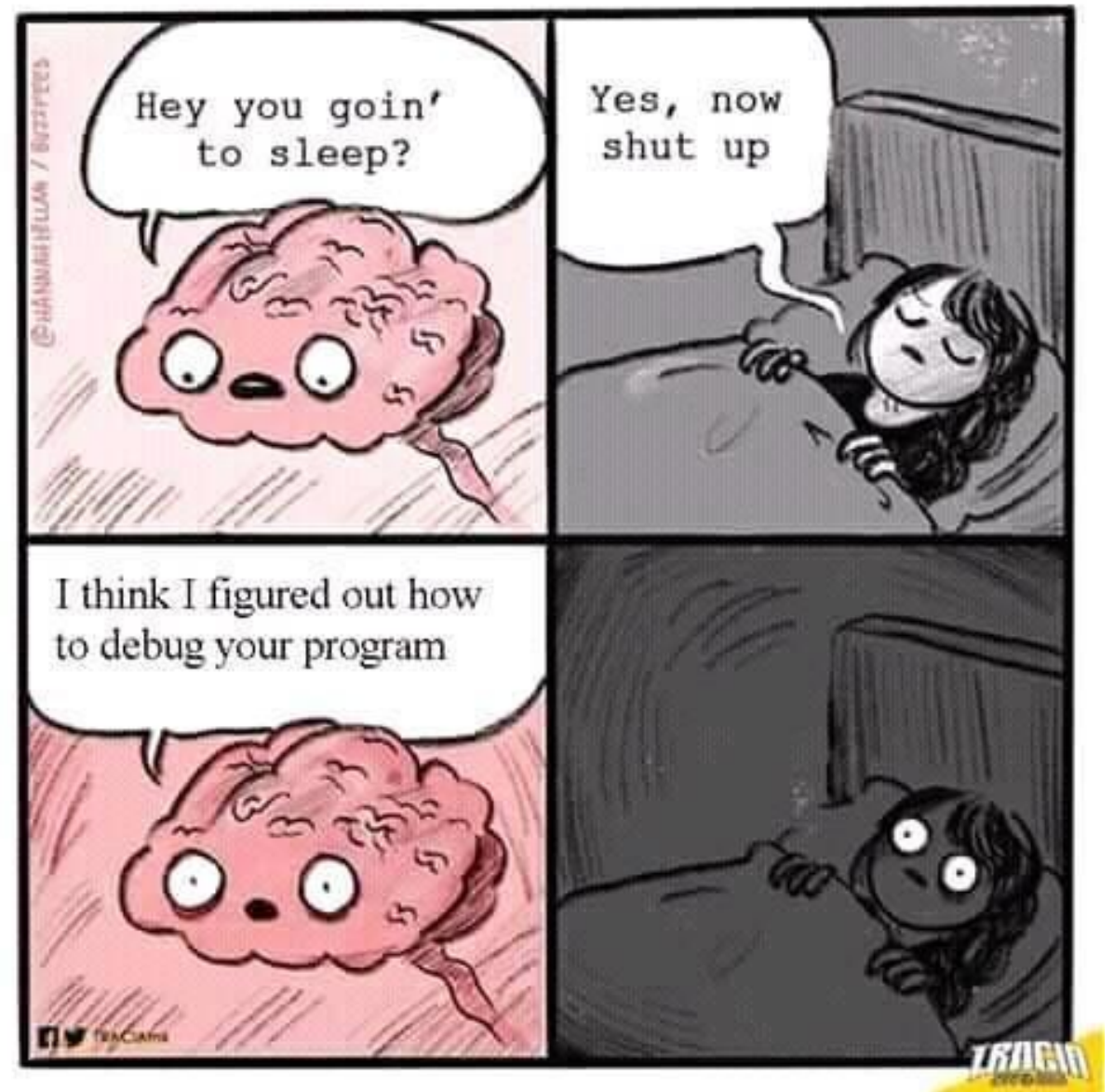
- `import collections`
- `import network`
- `import numpy`
- `import torch`
- `import keras`
- `import pandas`
- `import scipy`
- `import matplotlib`

Puedes definir tus propias funciones dentro de clases y agruparlas en librerías y contribuir las con la comunidad de programación

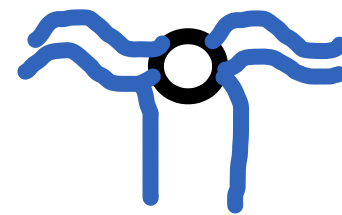


conda install

Capítulo II.IV – Introducción a depuración



<http://starecat.com/page/624/>



Errores comunes en programación

- Errores durante compilación: causado por sintaxis principalmente

Canta la canción "Por una cabeza"

- Errores durante ejecución: índice fuera de límites, StackOverflow (desborde de pila)

Canta “Por una Cabeza” de Carlos Gardel, sin mover la lengua.



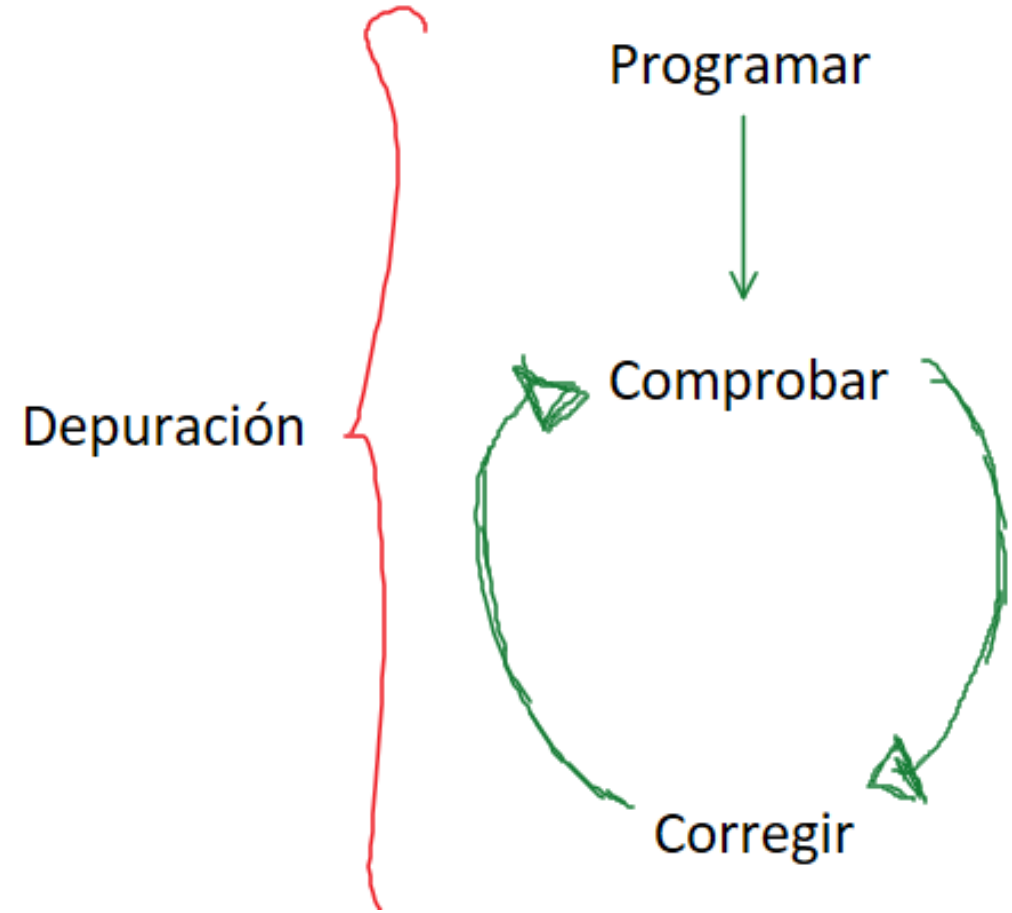
<https://www.etsy.com/listing/935328437/tango-dancing-couple-machine-embroidery>

- Errores lógicos: todo funciona bien, pero el problema no fue resuelto
- Dado un entero no negativo m , genera e imprime una lista que contiene m números aleatorios. El rango de números aleatorios es entre 1 y 6.



Solución de errores: depuración

Programar es como bailar: es un proceso iterativo de mejoramiento

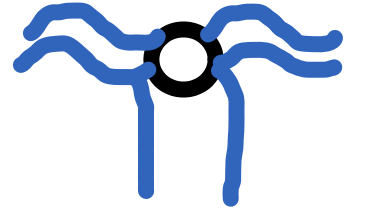


Formas de depuración

- Imprimir los resultados
- Escribir archivos de depuración



Problema ejemplar: Revertir un arreglo



- Dado un arreglo de enteros, escribe una función que lo invierta: Por ejemplo, dado $[1, 2, 3, 4, 5]$, conviértelo a $[5, 4, 3, 2, 1]$.

¿Cómo lo esquematizarías?

Resumen

- Depuración es como practicar tango.
- Puedes chequearlo manualmente o escribir archivos de depuración
- En Python, el proceso de escribir código y depuración es similar a Java



Bibliografía

- Textos a consultar:
- Capítulo 3, sección 3.7 de Introduction to Programming Using Java <http://math.hws.edu/javanotes/index.html>
- Capítulo 8 de Introduction to Programming Using Java <http://math.hws.edu/javanotes/index.html>

Ejercicios

- Algunos ejercicios son traducidos desde: <http://www.beginwithjava.com/java/loops/questions.html>
1. Dado un entero m, chequea si este contiene dos dígitos, devolviendo "True". Si tiene más dígitos imprime "tiene más de dos dígitos".
 2. En java, imprime una lista de números que tú inicialices. Puedes usar un bucle mejorado. Haz lo mismo en Python, pero sin un bucle.
 3. Dado una lista de enteros (e.g. [3,7,5, 8, 10, 1, 4, 2]), imprime con una condición cada uno de los valores de la lista elevado al cuadrado. La condición es que los valores impresos no pueden exceder 25.
 4. Dado una lista de caracteres (e.g. ['a', 'c', 'D', 'P', 'j']), solo imprime los valores que están en mayúscula.

1. Dado un entero n , imprime una lista de números elevados al cubo, pero de orden descendiente a partir de n .
2. Dado un entero n , imprime una sucesión Fibonacci hasta el término n .
3. Dado un entero n , calcular la suma de la sucesión aritmética $1 + 1/2 + 1/3 + 1/4 + 1/5 + \dots + 1/n$
4. Dado un arreglo de enteros, calcula el promedio de los valores dado en este. Trata de no usar funciones externas como `sum()` o `len()`.
5. Dado un entero m , imprime la siguiente estructura. Lo que está entre paréntesis no es necesario imprimir.
 - 1111...1 (m veces)
 - 222..2($m-1$ veces)
 - 33..3 ($m-2$ veces)
 -
 - m (1 vez)



Capítulo II.V – Resolviendo diversos problemas

Objetivos

- Revisitar los conceptos del presente capítulo en Python
 - Identificar qué temas fueron difíciles de consolidar y practicar
- Introducir Jupyter Notebook a través de Anaconda



Problemas

- Define una función que dado un radio en decimal, este calcula el volumen de una esfera. La fórmula a usar es $V = \frac{4}{3}\pi r^3$
 - Temas involucrados: manipular variables, importación de librerías, depuración

- Dado un entero m, imprime la siguiente estructura. Lo que está entre paréntesis no es necesario imprimir.

1111...1 (m veces)

222..2(m-1 veces)

33..3 (m-2 veces)

....

m(1 vez)

- Temas involucrados: manipular variables, tipos de variables, bucles
- Dado un arreglo de enteros, calcula el promedio de los valores dado en este. Trata de no usar funciones externas como `sum()` o `len()`.
 - Temas involucrados: bucles, funciones predefinidas

- **Dado una lista de caracteres (e.g. ['a', 'c', 'D', 'P', 'j']), solo imprime los valores que están en mayúscula.**
 - **Temas involucrados: tipos de variables, bucles, estructuras de control**
- **Dado un entero m, chequea si este contiene dos dígitos, devolviendo “tiene 2 dígitos”. Si tiene más dígitos imprime “tiene más de dos dígitos”. Si tiene solo un dígito, imprime “solo tiene un dígito”.**
 - **Temas involucrados: tipos de variables, estructuras de control**
- **Chequea si una lista es palíndroma, lo cual significa que la lista es igual tanto si lo vez de izquierda a derecha o viceversa.**
 - **Temas involucrados: variables, estructuras de control, bucles, depuración**
- **Dado un entero n, imprime un sucesión Fibonacci hasta el término n .**
 - **Temas involucrados: bucles, depuración**

Referencias

- Si quieres más ejercicios puedes revisar HackerRank, LeetCode, o cualquier otro sitio web que encuentres
- Algunos ejercicios son traducidos desde el libro Python Programming de John M. Zelle y desde el sitio web:
<http://www.beginwithjava.com/java/loops/questions.html>