

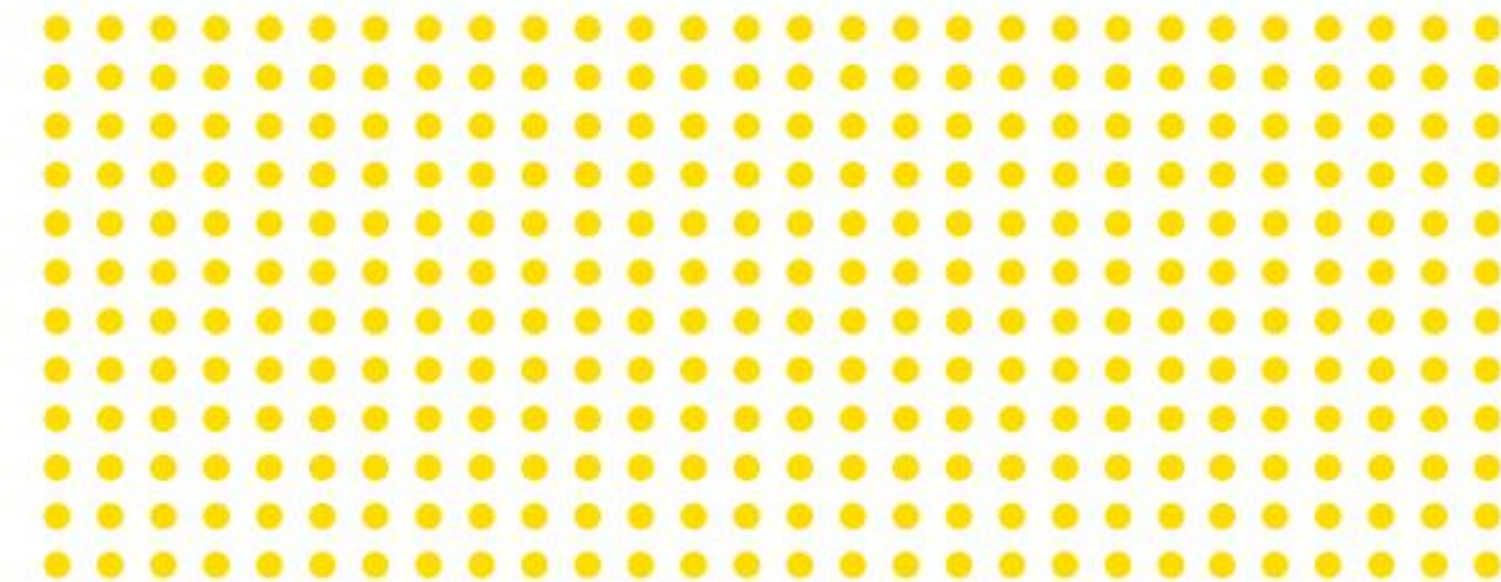
Pensamiento algorítmico: Funciones

Fundamentos de programación

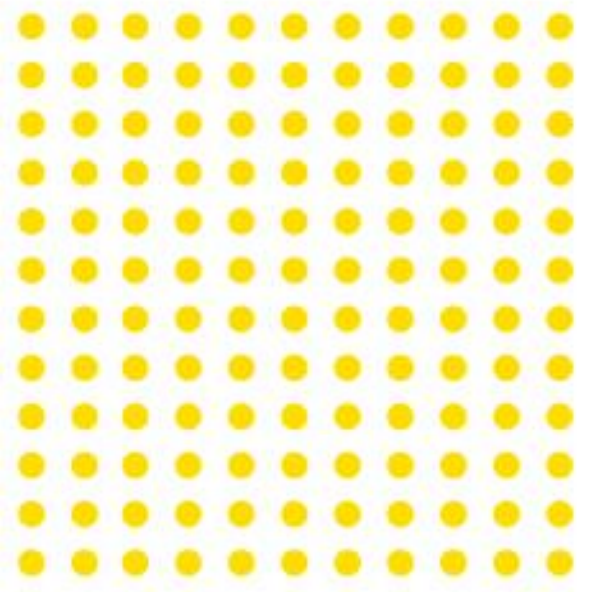


Universidad de
los Andes

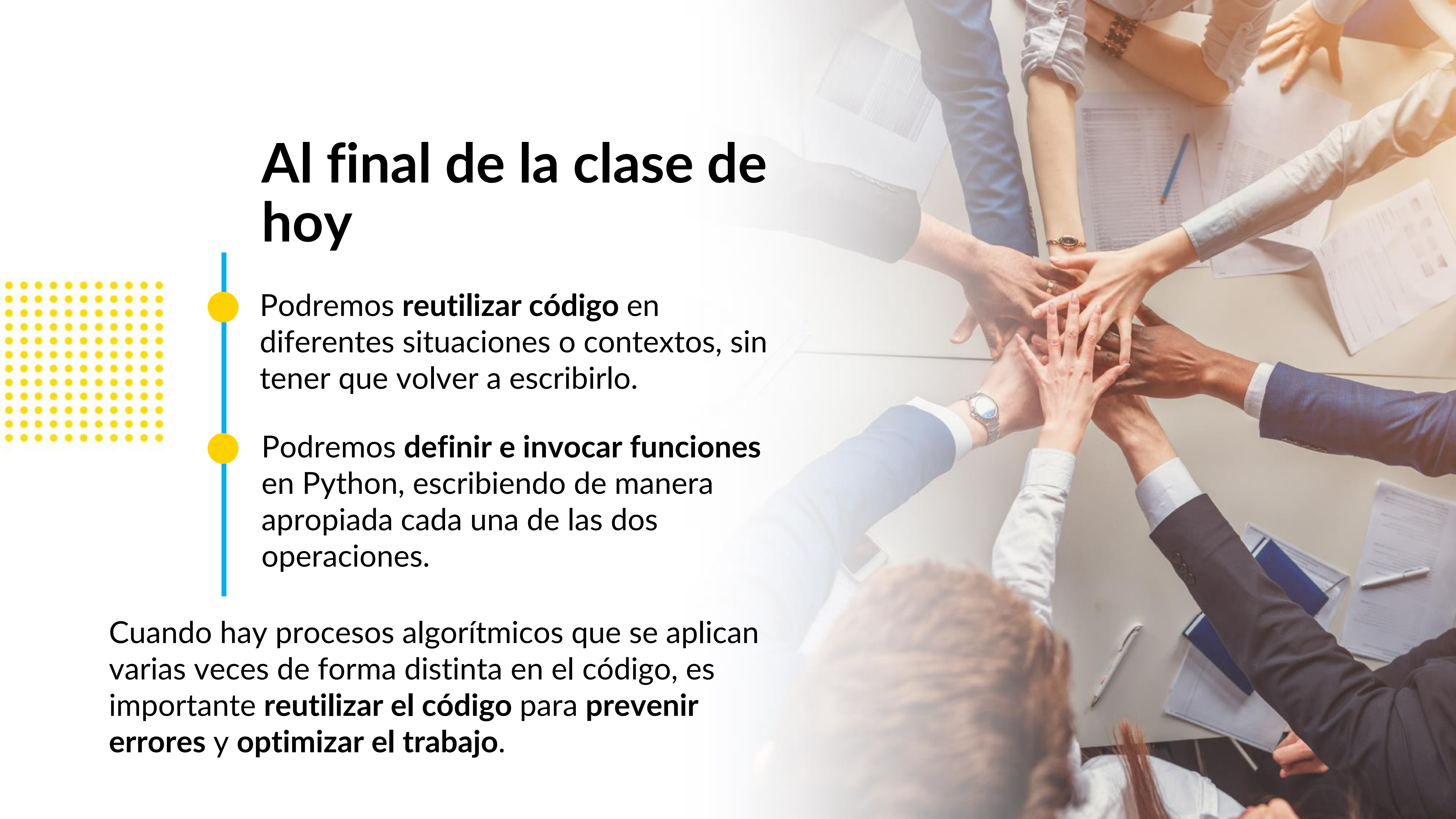
Educación
Continua
Vicerrectoría Académica



Al final de la clase de hoy

- 
- Podremos **reutilizar código** en diferentes situaciones o contextos, sin tener que volver a escribirlo.
 - Podremos **definir e invocar funciones** en Python, escribiendo de manera apropiada cada una de las dos operaciones.

Cuando hay procesos algorítmicos que se aplican varias veces de forma distinta en el código, es importante **reutilizar el código** para **prevenir errores** y **optimizar el trabajo**.





Pensamiento algorítmico - Funciones

Pensamiento algorítmico

- Reutilizar operaciones y algoritmos

Trabajando con funciones

- Definir funciones
- Invocar funciones

Al cuaderno de Python.

Reutilizar algoritmos

Pensamiento algorítmico



Ejercicio (1/2)

Por favor explique en el chat cómo llegar a la biblioteca más cercana desde donde se encuentra.

Piense en donde usted está ubicado/a.



Ejercicio (2/2)

Imagine que se acaba de mudar a una nueva ciudad que nunca había visitado.

Se encuentra en el aeropuerto.

Escriba en el chat instrucciones para llegar a la biblioteca más cercana.





Hay instrucciones que se usan varias veces

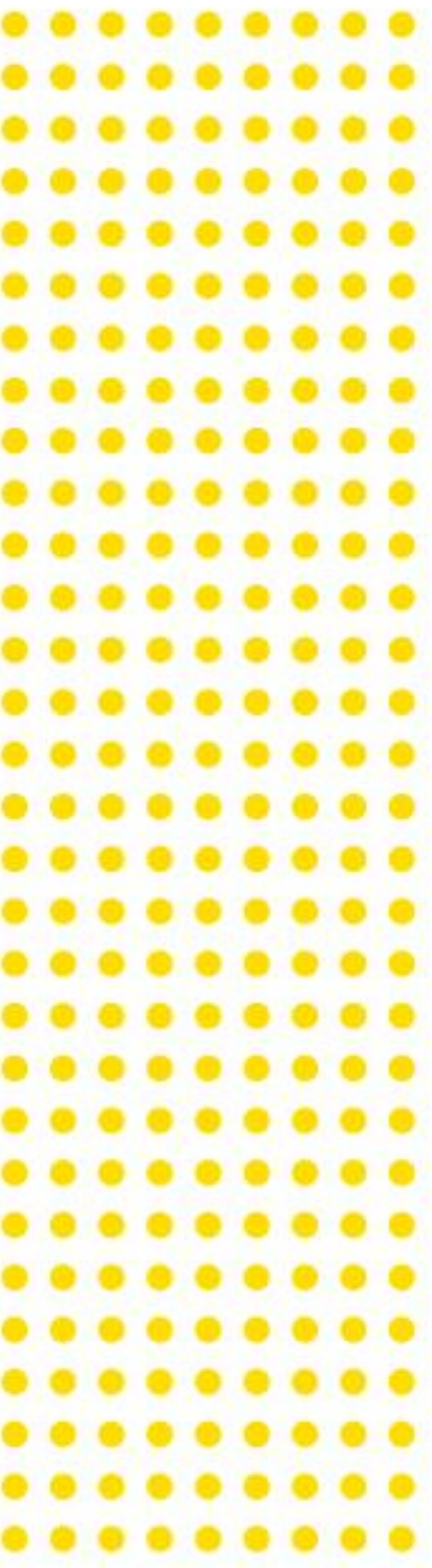


- Ayer vimos en los ejemplos que cuando elaboramos instrucciones, hay algunas más generales y otras más específicas.
- Frecuentemente nos vamos a encontrar en situaciones en las que podemos escribir código **más específico que se puede usar una sola vez, o más general que se puede usar varias veces.**

Usamos información

Para hacer un código más general, podemos identificar información clave qué utilizar y consultarla.

Por ejemplo, formulando instrucciones del estilo: preguntar la ubicación de la librería más cercana



Dos formas de usar información

La función requiere información directamente que luego va a utilizar siempre que sea invocada: **parámetros**.



Cuando invocamos la función, esta le pregunta a alguien más por información clave, y la utiliza cuando es invocada para su proceso.



En Python

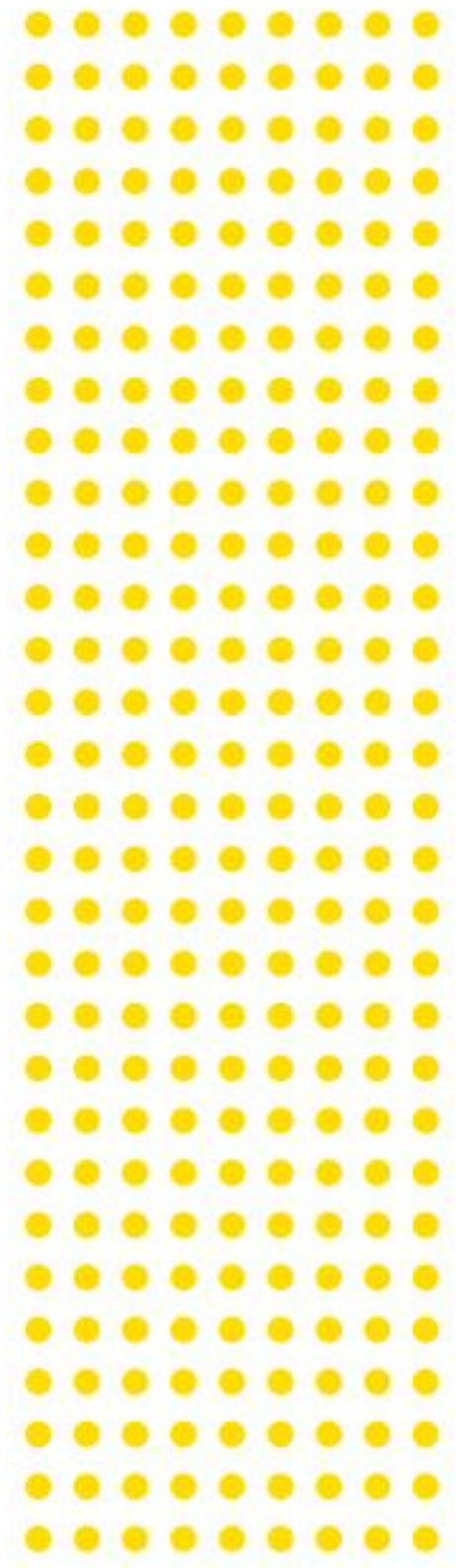
Trabajando con funciones



Para reutilizar código usamos funciones

Una función es un código que suele ser relativamente general y cumple una (y solo una, por buena práctica) tarea concreta.

- 1. Hay que definirlas:** definir el paso a paso (relativamente general) para lograr la tarea concreta.
- 2. Hay que invocarlas:** se las llama y se les da toda la información que necesitan para activarlas.



Ejercicio

Cuáles de las siguientes son buenos ejemplos de funciones:

- Calcular algún estadístico con los datos
- Calcular el promedio y calcular la varianza
- Cargar los datos desde la nube
- Dibujar un gráfico con los colores preferidos



Ejercicio

Cuáles de las siguientes son buenos ejemplos de funciones:

- Calcular algún estadístico con los datos **No es específica**
- Calcular el promedio y calcular la varianza **Son 2 tareas**
- Cargar los datos desde la nube **Bien**
- Dibujar un gráfico con los colores preferidos **Bien**



Para definir

def es la palabra clave

Variable para representar el parámetro del número de tasas a preparar.

Dos puntos para indicar el grupo de código que describe el algoritmo

```
def prepararCafe(numeroTasas):  
    print('Aquí tienes tus ' + str(numeroTasas) + ' tasas.')  
    print('¡Buen provecho!')
```

Grupo de código descrito por la indentación

Usamos el nombre de la variable para tomar el valor que nos manden cuando la invoquen

Para invocar

Invocamos el nombre de la función

Valor de la información requerida: parámetro.
Si hay más de uno **van en orden**.

`prepararCafe(3)`

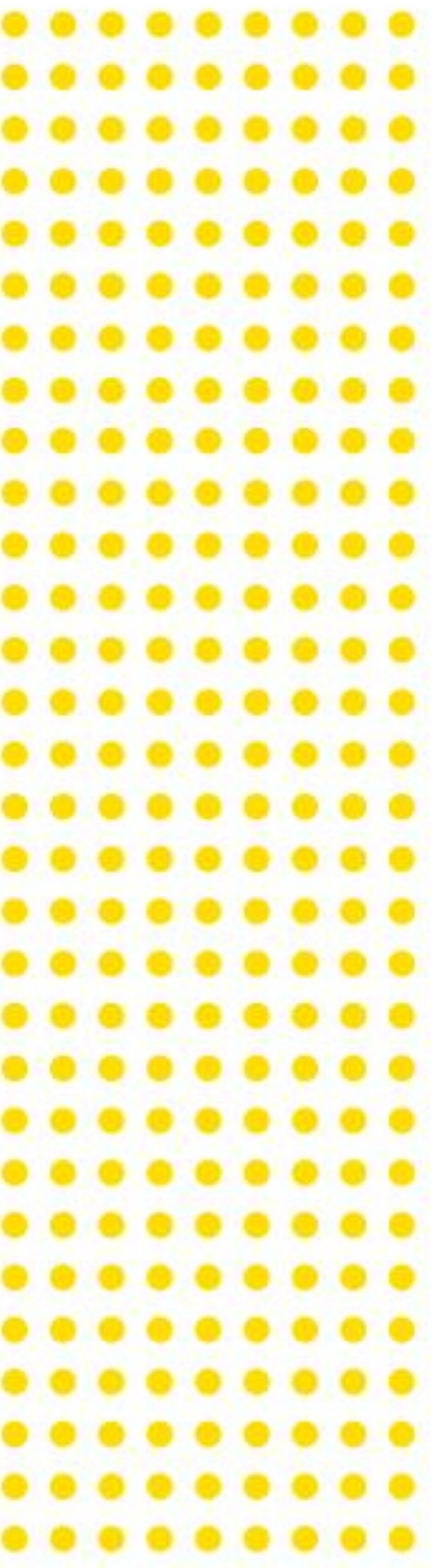
Aquí tienes tus 3 tasas.
¡Buen provecho!

Se ejecuta todo el bloque de código descrito como parte del algoritmo.

Return

A veces queremos que, al invocar una función, esta nos devuelva un resultado para poder usarlo.

Para esto empleamos la palabra **return**



Para invocar

```
def sumarNumeros(numero1, numero2):  
    return numero1 + numero2
```

```
miValorFavorito = sumarNumeros(10, 12)  
print(miValorFavorito)
```

```
22
```




Con todo esto...

1. Ya sabemos cómo pensar en términos de instrucciones **más generales o más específicas**, según el caso que sea más pertinente.
2. Sabemos reutilizar código mediante funciones, y estamos en la capacidad de **definir e invocar** una función.
3. Sabemos reutilizar código mediante funciones, y estamos en la capacidad de **definir e invocar** una función.



¡Gracias!

Aprendiendo juntos a lo largo de la Vida

educacioncontinua.uniandes.edu.co

Síguenos en **EdcoUniandes**

