universidad cenfotec
La U de la informática

Principios de Programación.

Introducción a Rutinas: Conceptos.

Objetivos

universidad cenfotec
La U de la informática

- Comprender el concepto de **abstracción procedimental**.
- Representar en un diagrama de flujo cualquier **rutina**.
- Representar en el lenguaje de programación Python cualquier algoritmo con **rutinas**.

Abstracción procedimental

universidad cenfotec
La U de la informática

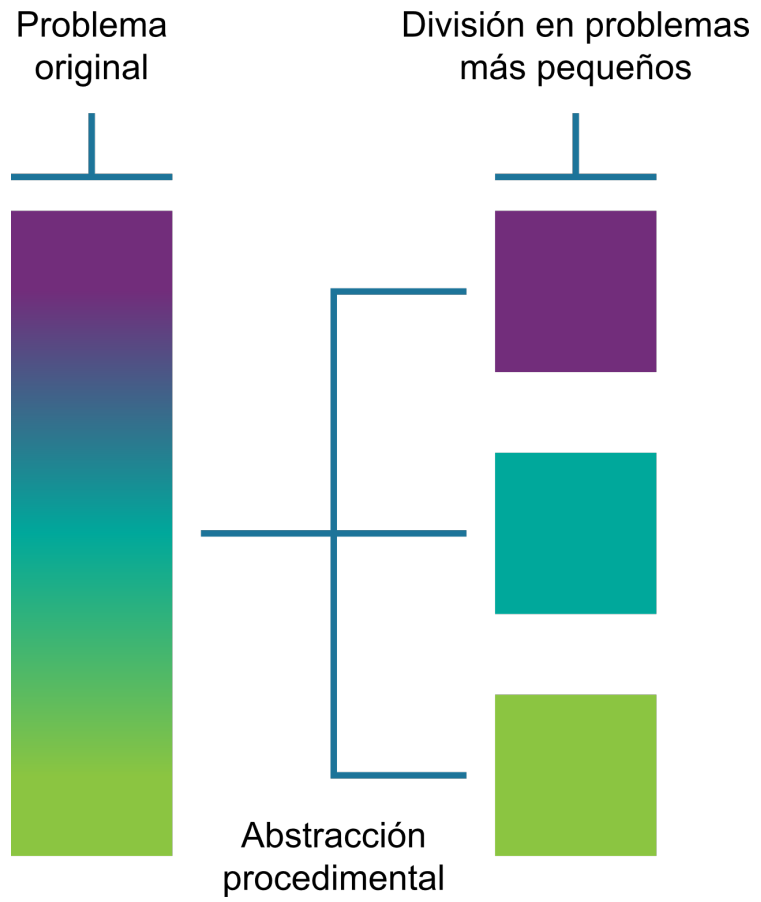
Los algoritmos pueden variar su tamaño dependiendo de la complejidad de los problemas que se plantean. En ocasiones, estos problemas pueden requerir muchos procesos para solucionarse, por lo que manejarlo todo dentro de una única línea de trabajo puede tornarse muy complejo.

Para esto, la experiencia ha demostrado que la mejor manera de resolver un problema grande es dividirlo en problemas más pequeños, más simples, a esta técnica se le conoce como el principio **“divide y vencerás”**.

Abstracción procedimental

El principio **“divide y vencerás”** lo que busca es dividir el problema en partes, generalmente más simples de entender y de analizar, con el fin de encontrar una solución al problema planteado originalmente.

Para cumplir con este principio, se utiliza la **abstracción procedimental**, la cual consiste en dividir el problema en procesos o unidades más pequeñas que resuelvan una parte únicamente, así, la sumatoria de todas las soluciones a los problemas pequeños, conformará la solución al problema original.



Abstracción procedimental

universidad cenfotec
La U de la informática

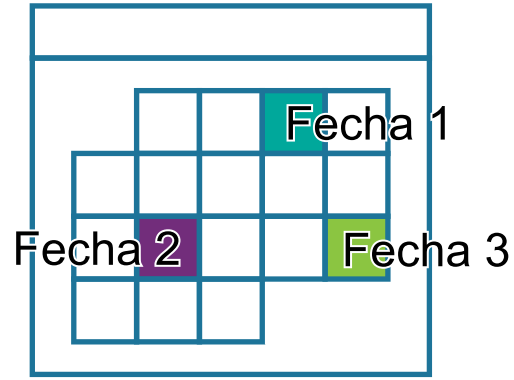
La **abstracción procedimental** nos permite:

- *Identificar estructuras importantes* que más tarde se van a reutilizar y que permiten entender mejor el problema.
- *Reutilizar código o programas* que ya están hechos (que ya se diseñaron, se codificaron, se compilaron, se probaron y funcionan correctamente).
- *Evitar repetir código y esfuerzos*, así los programas se hacen más rápido y con menos probabilidad de errores.
- La *modularización*, es decir, combinar diferentes partes de diferentes programas sin mayor esfuerzo, para armar otros programas nuevos.

Ejemplo 1

Para entender un poco más el concepto observe el siguiente ejemplo. Suponga que en un programa desarrollado para un banco se manejan varias fechas que tienen diferentes significados:

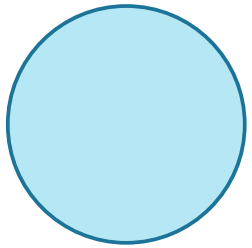
- Fecha de nacimiento del cliente.
- Fecha de formalización del préstamo.
- Fecha de pago de las cuotas.
- Fecha de vencimiento del pago, etc.



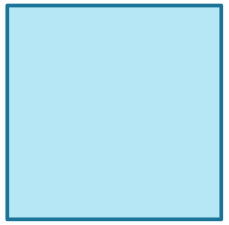
Es importante que cada una de estas fechas sea una fecha válida, para asegurarse de esto puede incluir en el programa un código con estructuras condicionales que valide cada una de las fechas. Pero ¿qué inconvenientes pueden presentarse al escribir varias veces el mismo código?

Ejemplo 2

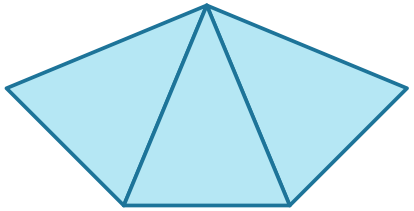
Para solucionar un problema se tiene que hallar el área y el perímetro de varios lotes que tienen diferentes formas geométricas. Suponga que los lotes se componen de círculos, triángulos y cuadrados, de la siguiente forma:



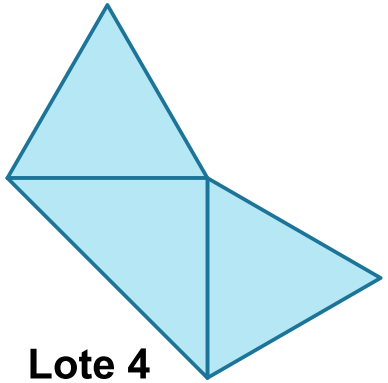
Lote 1



Lote 2



Lote 3



Lote 4

Ejemplo 2 (continuación)

Para hallar el área y el perímetro de los lotes entonces se calcula el área de cada figura, se calcula el perímetro de cada figura y luego se suman para tener el área y el perímetro de cada uno de los lotes.

Para hallar el área y el perímetro de cada uno de los lotes se usan las siguientes fórmulas:

- Área de un círculo = ***π * radio²***.
- Perímetro de un círculo (circunferencia) = ***2 * π * radio***.
- Área de un triángulo = ***(base * altura) / 2***.
- Perímetro de un triángulo = ***lado1 + lado2 + lado3***.
- Área de un cuadrado = ***lado1 * lado2***.
- Perímetro de un cuadrado = ***lado1 + lado2 + lado3 + lado4***.

Con esta solución, se puede hallar el área y el perímetro de cualquier lote conformado por estas figuras geométricas.

Recapitulación

universidad cenfotec
La U de la informática

Aplicación de la abstracción procedimental:

- En el ejemplo 1 se identificó como estructura importante la validación de una fecha, de tal manera que, al implementarse esta validación, pueda ser **reutilizada** cada vez que se necesite para solucionar este o inclusive otros problemas que impliquen validación de fechas, sin necesidad de volver a comenzar a desarrollar el código de la validación desde cero, si no **reutilizándolo y evitando repetir esfuerzos**.

- En el ejemplo 2 se identificaron como estructuras importantes el cálculo de las áreas y los perímetros de las diferentes figuras geométricas, para poder calcular el área y el perímetro de los lotes. Es decir, que **se dividió el problema original en varios problemas más pequeños**, de los cuales es más fácil hallar la solución. Luego, la unión de la solución de estos problemas pequeños conforma la solución al problema original más grande.

Abstracción procedimental

El proceso de **abstracción procedimental** comprende tres fases:

- **Fase de análisis**: consiste en entender el problema que se quiere resolver, que generalmente un problema grande. En esta fase además, se identifican cuáles datos se necesitan para resolver el problema y cuáles datos son los que se producen como resultado de la solución del problema.

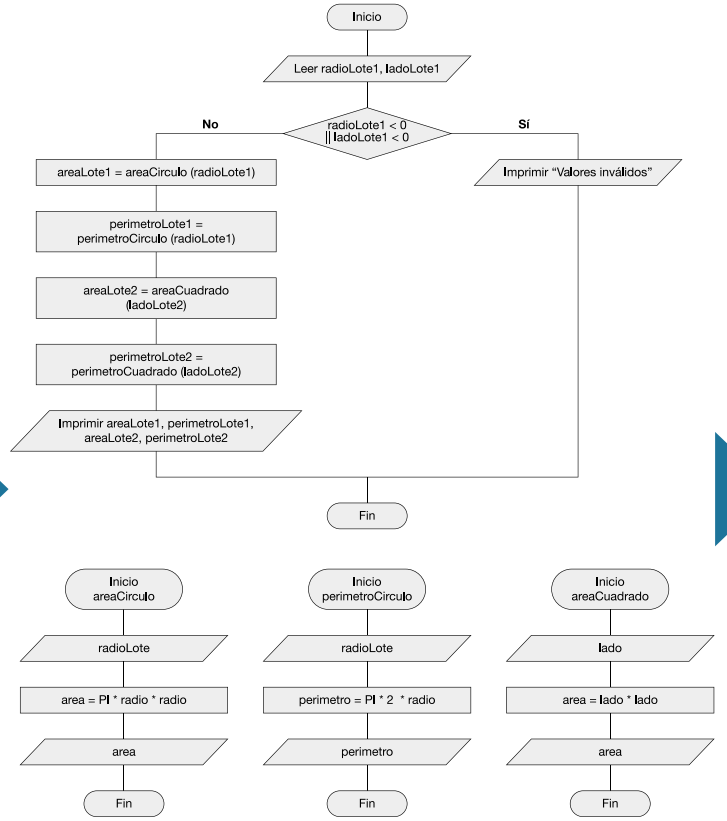
- **Fase de diseño o modelado**: consiste en identificar los procesos o unidades más pequeñas de las cuales se puede hallar una solución más fácilmente, modelar la solución de cada uno de estos procesos y unir estos modelos para conformar la solución al problema que se quiere resolver.

- **Fase de implementación**: consiste en traducir los modelos a un lenguaje de programación para automatizar la solución al problema. La programación de los pequeños modelos genera pequeños bloques de código, llamados **rutinas**.

Fases de la abstracción procedimental

universidad cenfotec
La U de la informática

ENTRADAS			
Descripción	Notación		Ejemplo
	Nombre	Tipo de dato	
Radio del lote circular	radioLote1	float	5.5
Lado del lote cuadrado	ladoLote2	float	5.8
SALIDAS			
Descripción	Notación		Ejemplo
	Nombre	Tipo de dato	
Valor del área del Lote 1	areaLote1	float	143.5
Valor del perímetro del Lote 1	perimetroLote1	float	122.5
Valor del área del Lote 2	areaLote2	float	123.5
Valor del perímetro del Lote 2	perimetroLote2	float	172.5




```python
import math
radioLote1 = 0.0
ladoLote2 = 0.0
areaLote1 = 0.0
perimetroLote1 = 0.0
areaLote2 = 0.0
perimetroLote2 = 0.0
PI = math.pi

radioLote1 = float(input('Por favor escriba el radio del Lote 1:'))
ladoLote2 = float(input('Por favor escriba el lado del Lote 2: '))

#Definición de las rutinas

def calcularAreaCirculo(pRadio):
    area = 0
    area = PI * pRadio * pRadio
    return area

def calcularPerimetroCirculo(pRadio):
    perimetro = 0
    perimetro = 2 * PI * pRadio
    return perimetro


def calcularAreaCuadrado(pLado):
    area = 0
    area = pLado * pLado
    return area

def calcularPerimetroCuadrado(pLado):
    perimetro = 0
    perimetro = pLado * 4
    return perimetro

# Flujo principal

if (radioLote1 < 0 or ladoLote2 < 0):
    print('Por favor ingrese valores positivos.')
else:
    areaLote1 = calcularAreaCirculo(radioLote1)
    perimetroLote1 = calcularPerimetroCirculo(radioLote1)
    print(f'El área del Lote 1 es: {areaLote1}')
    print(f'El perímetro del Lote 1 es: {perimetroLote1}')

    areaLote2 = calcularAreaCuadrado(ladoLote2)
    perimetroLote2 = calcularPerimetroCuadrado(ladoLote2)
    print(f'El área del Lote 2 es: {areaLote2}' )
    print(f'El perímetro del Lote 2 es: {perimetroLote2}' )
```

Análisis **Diseño** **Implementación**

Rutinas

universidad cenfotec
La U de la informática

- **Una rutina** es la implementación de los procesos más pequeños identificados en la fase de modelado, que en conjunto, van a conformar la solución de un problema más grande.

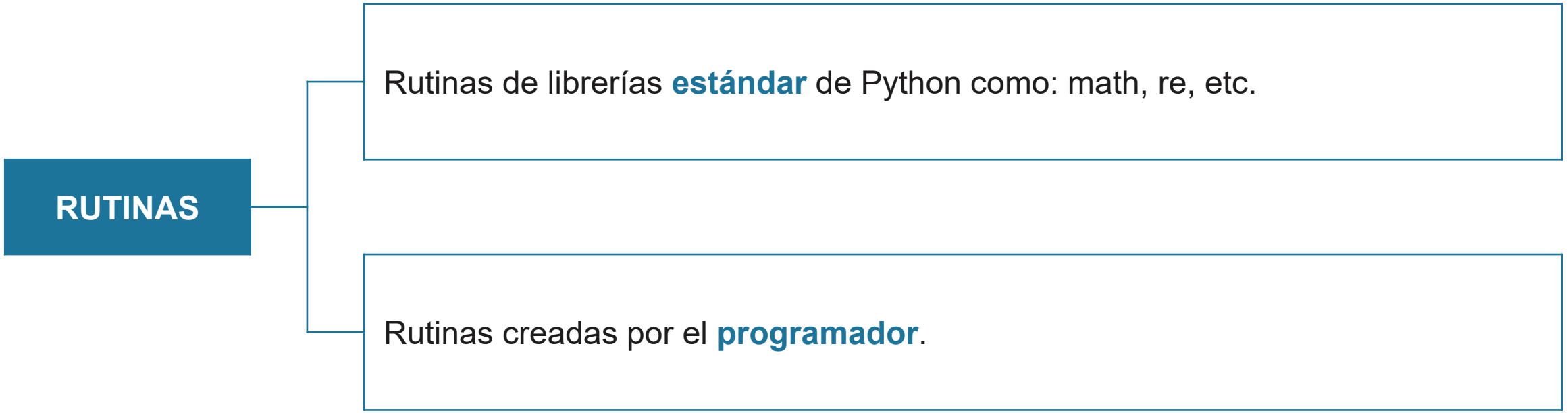
- Es además, una sección o bloque de código independiente que contiene una serie de instrucciones, tiene una **responsabilidad u objetivo definido y realiza una tarea específica**.

- La definición anterior encierra dos conceptos fundamentales:
  - Una rutina es **independiente** del resto del programa.
  - Una rutina tiene una **responsabilidad definida**, se crea con un objetivo en mente y su responsabilidad es cumplir dicho objetivo.

Rutinas

universidad cenfotec
La U de la informática

Las rutinas pueden ser clasificadas por su origen, así:



Rutinas (conceptos importantes)

universidad cenfotec
La U de la informática

Existen dos conceptos fundamentales a la hora de hablar de rutinas, el primero de ellos son los **valores de retorno**.

Valor de retorno.

- Un valor de retorno es el valor de **salida** que produce una *función* después de ejecutar un proceso. ***Ejemplo***: el área de un círculo, el salario de una persona, etc.
- Esta salida **no** es una impresión en pantalla, sino que está representada por una variable, que tendrá tipo de dato, nombre y valor.
- El **valor** de esta variable es adquirido durante la ejecución de la *función* y es justamente el valor de retorno.

Rutinas (conceptos importantes)

universidad cenfotec
La U de la informática

El segundo concepto fundamental en relación a la construcción de rutinas son los **parámetros**.

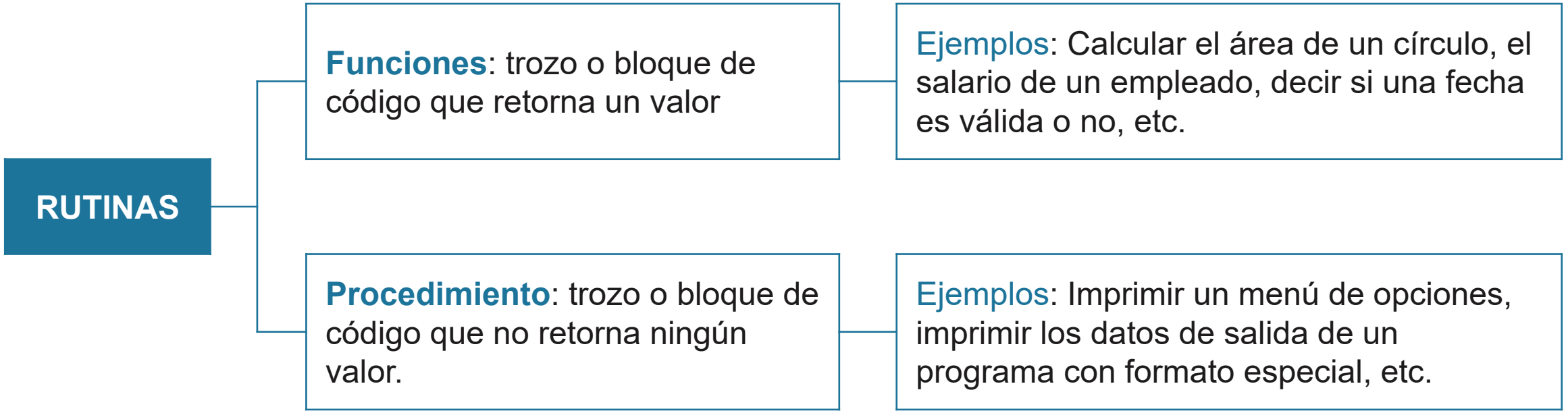
Parámetro.

- Un parámetro es un **valor de entrada** a la rutina, necesario para que ésta pueda lograr su objetivo.
- Un parámetro está representado por una **variable**, que tendrá tipo de dato, nombre y valor.
- El valor del parámetro es adquirido **antes** de la rutina, se usa dentro de la rutina y puede ser modificado durante la ejecución de la rutina.

Rutinas (clasificación de rutinas por su objetivo)

universidad cenfotec
La U de la informática

Las rutinas pueden ser clasificadas por su objetivo dentro del programa de la siguiente forma:



Rutinas (clasificación de rutinas por su objetivo)

universidad cenfotec
La U de la informática

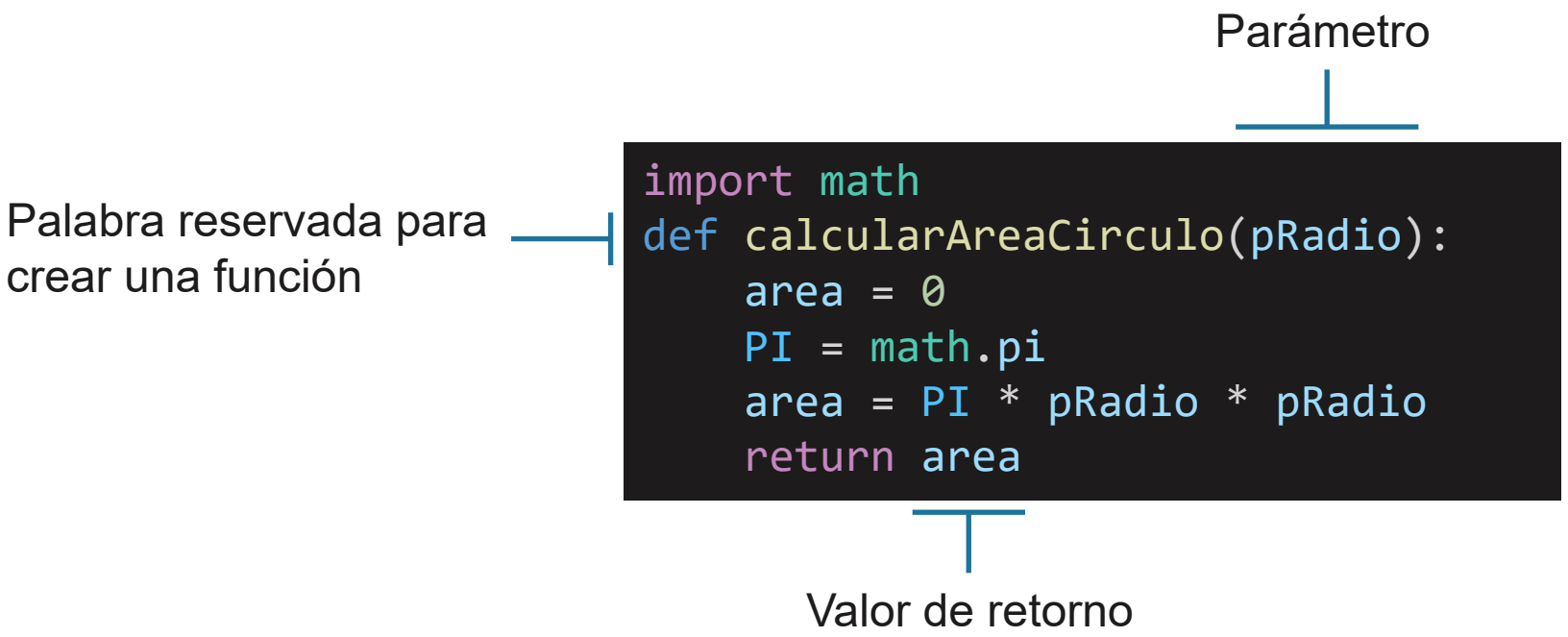
Funciones

- Una función es una rutina diseñada para realizar una tarea muy específica y **retornar un valor**.
- El valor de retorno de una función puede ser **de cualquier tipo de dato**: *boolean*, *char*, *int*, *float*, *double*, *String*, etc.
- Una función puede recibir cero o más **parámetros**.
- Ejemplos: Calcular el mayor de dos números y retornar el valor del número mayor, verificar si una fecha es válida y retornar un valor booleano dependiendo del resultado, verificar si un año dado es bisiesto y retornar un valor booleano dependiendo del resultado, etc.

Rutinas (clasificación de rutinas por su objetivo)

universidad cenfotec
La U de la informática

Un ejemplo de una **función** en Python que calcula el área de un círculo es:



Rutinas (clasificación de rutinas por su objetivo)

universidad cenfotec
La U de la informática

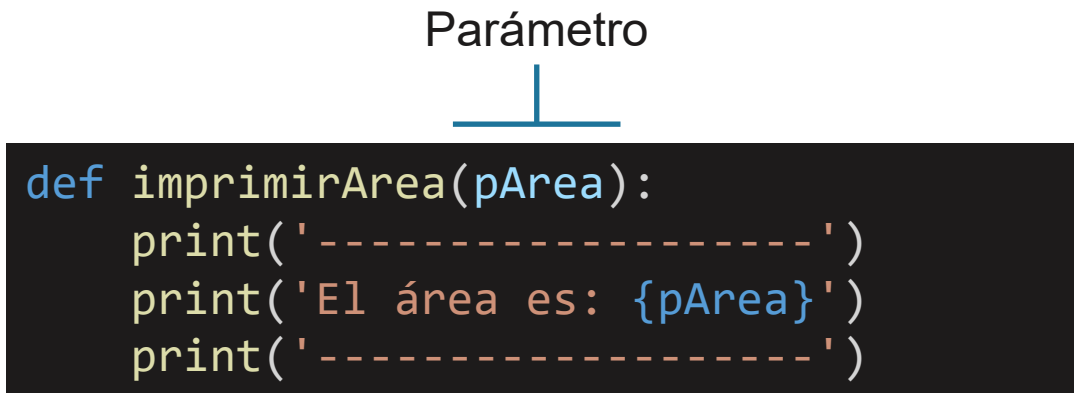
Procedimientos

- Un **procedimiento** es una rutina diseñada para realizar tareas muy específicas y tiene la característica que **no genera ningún valor de retorno**.
- Un procedimiento puede recibir cero o más **parámetros**.
- Ejemplos: imprimir un menú de opciones para guiar al usuario en el uso del programa, imprimir el valor de algunos datos de salida con un formato específico, imprimir encabezados al inicio de los programas para dar mayor información al usuario, etc.

Rutinas (clasificación de rutinas por su objetivo)

universidad cenfotec
La U de la informática

Un ejemplo de un procedimiento en Python que imprime el área de una figura con cierto formato es el siguiente:



Más adelante se ampliará el tema de cómo analizar, diseñar y programar procedimientos.


universidad
cenfotec_
La U de la informática