



Guía de ejercicios

Módulo 4

Ejercicio 1. Saludo	3
Ejercicio 2. Formulario de registro	3
Ejercicio 3. Validación de rango	4
Ejercicio 4. Operaciones aritméticas	4
Ejercicio 5. Área y perímetro de un rectángulo	4
Ejercicio 6. Área y circunferencia de un círculo	5
Ejercicio 7. Intercambio de valores	5
Ejercicio 8. Verificar que un número sea de dos dígitos	5
Ejercicio 9. Manipulación de strings hexadecimales	6

Recordatorios

Para leer datos de entrada podemos usar la función `input()` que devuelve una cadena de caracteres. Equivale al `Leer` de PSeInt.

Para mostrar por pantalla podemos usar `print()` que equivale al `Escribir` de PSeInt. Al utilizar los Notebooks también podemos simplemente poner el nombre de la variable y se mostrará su contenido. Sin embargo es mejor utilizar explícitamente `print()` para los ejercicios en caso que se quieran editar o ejecutar en un IDE o línea de comandos.

Ejercicio 1. Saludo

Escribe un programa que escriba dos líneas: tu nombre y una frase sobre tí.

El resultado debería ser algo así:

```
Satoshi Nakamoto
Creador de Bitcoin
```

Ejercicio 2. Formulario de registro

Escribe un programa que recoja datos para formalizar el registro de un usuario en un servicio online. Queremos recoger el nombre, la edad, el sueldo anual aproximado y si el usuario dispone de un monedero Ethereum (solo es necesario preguntar sí o no).

Tras recoger todos esos datos se ha de mostrar un resumen de todos los datos leídos.

El resultado debería ser algo así:

```
Introduzca su nombred
Introduzca su edadd
Introduzca su sueldod anuald
¿Dispone de un monedero Ethereum? (Si/No)s

Resumen del registro
Nombre: d
Edad: d
Sueldo: d
Monedero Ethereum: False
```

Ejercicio 3. Validación de rango

Escribe un programa que pregunte al usuario cuántos Bitcoin tiene, validando que el valor introducido esté dentro del rango 0 a 21000000.

El resultado debería ser algo así:

```
Introduce tu saldo en BTC: 326.23
```

```
Saldo valido: True
```

Ejercicio 4. Operaciones aritméticas

Escribir un programa que pida dos valores y muestre por pantalla el resultado de aplicar cada uno de los operadores aritméticos a los valores introducidos

El resultado debería ser algo así:

```
Introduce el primer valor: 3
Introduce el segundo valor: 66.4
```

```
Suma: 69.4
Resta: -63.400000000000006
Multiplicación: 199.20000000000002
División: 0.04518072289156626
División entera: 0.0
Módulo: 3.0
Potencia: 4.795692334871177e+31
```

Ejercicio 5. Área y perímetro de un rectángulo

Escribir un programa que calcule el área y perímetro de un rectángulo dadas su base y altura.

El resultado debería ser algo así:

```
Introduce la base del rectangulo: 34
Introduce la altura del rectangulo: 3.2
```

```
Area: 108.80000000000001
Perimetro: 74.4
```

Ejercicio 6. Área y circunferencia de un círculo

Escribir un programa que calcule el área y perímetro de un círculo dado su radio.

Como comentamos en la parte de PSeInt los lenguajes de programación acostumbran a incorporar las principales constantes y funciones matemáticas como Pi. En Python podemos acceder a esa constante con `math.pi` y para ello tenemos que importar el módulo matemático con `import math`. No es necesario entender este concepto ahora mismo, lo veremos más adelante.

El resultado debería ser algo así:

```
Introduce radio: 2.7

Circunferencia: 16.964600329384883
Area: 22.902210444669596
```

Ejercicio 7. Intercambio de valores

Escribir un programa que solicite dos valores para asignar a las variables A y B e intercambie los valores de esas variables.

El resultado debería ser algo así:

```
Valor para A: hola
Valor para B: mundo

A: mundo
B: hola
```

Ejercicio 8. Verificar que un número sea de dos dígitos

Escribir un programa que lea un número entero y compruebe si el número tiene dos dígitos. El resultado debería ser algo así:

```
Introduce un número de dos dígitos: 33

¿El número introducido tiene dos dígitos? True
```

Ejercicio 9. Manipulación de strings hexadecimales

Dado una dirección de un contrato extraer el valor hexadecimal sin el prefijo.

Valor esperado: `dAC17F958D2ee523a2206206994597C13D831ec7`

El resultado debería ser algo así:

```
'dAC17F958D2ee523a2206206994597C13D831ec7'
```