

## Introducción a Python

Python es un lenguaje de programación orientado a objetos, lo usaremos durante la cursada, a medida que necesitemos iremos incorporando nuevos conceptos y herramientas.

A diferencia de otros lenguajes Python es un lenguaje interprete, es decir que ejecuta una instrucción una tras otra a medida que las lee, o uno las escribe. Otros lenguajes como C y C++ que requieren compilar el programa antes de poder ejecutarlo.

Instalen Python 3. Les instala 2 programas "Python 3.xx" y "IDLE (Python 3.xx)". Empezaremos usando el primero, el segundo es un entorno de desarrollo para hacer programas, lo usaremos más adelante.

**NOTA:** En la página de Python van a ver también Python 2, se mantiene por compatibilidad con otras aplicaciones que lo usen, pero Python 2 y Python 3 no son compatibles entre sí. Un programa para Python 2 no se puede ejecutar en Python 3 y viceversa, hay que corregirles las diferencias entre las 2 versiones.

Ahora si, a probar un poco. Ejecuten Python 3. Les tiene que aparecer una ventana con algo similar a esto.

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

### Calculadora

Podemos usar Python como una calculadora, por ejemplo.

```
>>> 3+4
```

Y presionen ENTER, hace el cálculo y muestra el resultado. Prueben otros cálculos:

```
3*4, 9/3, 10/3, 10//3 y 4**2.
```

### **TAREA:**

1) Investiguen cuales son los operadores matemáticos disponibles en Python.

Veamos el siguiente problema. Un pintor necesita calcular la superficie total de las paredes de una habitación. supongamos que la habitación mide 3.50m. x 3.00m. y 2.60m. de altura. ¿cuál es la superficie de las paredes para pintar? Por ejemplo pueden hacer:

3.50\*2.60, 3.00\*2.60, 3.50\*2.60, 3.00\*2.60 y luego sumar los 4 resultados. O lo pueden hacer en una sola operación:

```
3.50*2.60 + 3.00*2.60 + 3.50*2.60 + 3.00*2.60
```

¿Pueden pensar cómo hacer esta operación más corta?

### Variables

Permiten almacenar un valor con un nombre que nos sea útil para realizar operaciones con él. El contenido de una variable lo podemos cambiar cuando uno necesite.

Por ejemplo, para el problema anterior podemos hacer:

```
Largo=3.50
Ancho=3.00
Alto=2.60
```

y luego la cuenta nos queda:

```
Largo*Alto + Ancho*Alto + Largo*Alto + Ancho*Alto
```

Que es mucho mas fácil de leer. Ahora, si tenemos que calcular otra habitación, podemos cambiar los contenidos de las variables, pero el cálculo será el mismo. Calculen la superficie a pintar de una habitación de 4.50 x 3.50 x 2.60.

### Repetición de cálculos

Para hacer cálculos nos sirve una calculadora, con la computadora podemos realizar cálculos en forma repetitiva. Hagamos un ejemplo con el COVID19. **Los datos suministrados NO son reales, sólo son de carácter ilustrativo.** Realicemos una simulación para ver cuántos casos nuevos hay cada día.

Variables:

Anterior: Cantidad de infectados del día anterior.

Nuevos: Cantidad de nuevos infectados.

Total: Cantidad total de infectados.

Parámetros:

Contactos: Cantidad de personas que uno se cruza por día a menos de un metro (que podrían infectarse)

Probabilidad: Probabilidad de infectarse por un contacto.

El algoritmo que vamos a utilizar es:

- 1) Inicializamos los valores.
- 2) Calculamos la cantidad de infectados del nuevo día.
- 3) Actualizamos el total de infectados
- 4) Actualizamos el total de infectados del día anterior (para realizar los cálculos del nuevo día).
- 5) Mostramos los resultados del día
- 6) Repetimos 30 veces desde el punto 2

Para los cálculos tenemos:

$$\text{Nuevos} = \text{Anterior} * \text{Contactos} * \text{Probabilidad}$$

$$\text{Total} = \text{Anterior} + \text{Nuevos}$$

Luego.

$$\text{Anterior} = \text{Total}$$

Probemos con algunos datos:

Anterior: 10

Contactos: 10

Probabilidad: 0.015 (1,5%)

Calculen la cantidad de casos nuevos y el total para el nuevo día.

Ahora veamos la evolución para 30 días. Repetir con cálculos 30 veces es aburrido, dejemos ese trabajo a la computadora.

Como vamos a tener que escribir varias líneas, usen el IDLE (Python 3.x), y abran in archivo nuevo (File -> New File) , para ejecutar el código: Run->Run Module. Todo lo escrito después del # es un comentario y no se ejecuta. Tengan cuidado de mantener la misma cantidad de espacios en cada línea después del for (como está escrito abajo). Python reconoce un bloque de programa por su indentación.

```
# Parámetros
Contactos = 10
Probabilidad = 0.015
# Valores iniciales
Anterior = 10
# Calculos
for dia in range(30):
    Nuevos = Anterior * Contactos * Probabilidad
```

```
Total = Anterior + Nuevos
Anterior = Total
print("Dia:", dia+1, " Infectados:", int(Total))
```

### TAREA:

2) Investiguen las funciones "int()" y "print ()". El comando "for" y la función "range()" las veremos más adelante.

Complicuemos las cosas un poco. El periodo de incubación es de 5 días aproximadamente antes que aparezcan los síntomas. Si tomamos como casos confirmados el 100% de los que presentan síntomas, ¿Cómo podemos mostrar la proyección de casos confirmados para cada día?

Podemos utilizar el siguiente algoritmo:

- 1) Inicializamos los valores.
- 2) Calculamos la cantidad de infectados del nuevo día.
- 3) Actualizamos el total de infectados
- 4) Guardamos temporalmente el total de infectados en una lista.
- 5) Actualizamos el total de infectados del día anterior (para realizar los cálculos del nuevo día).
- 6) Mostramos los resultados del día.
- 7) Repetimos 5 veces desde el punto 2.
- 8) Calculamos la cantidad de infectados del nuevo día.
- 9) Guardamos temporalmente el total de infectados en una lista.
- 10) Actualizamos el total de infectados del día anterior (para realizar los cálculos del nuevo día).
- 11) Recuperamos el total de infectados de hace 5 días (casos confirmados).
- 12) Mostramos los resultados del día.
- 13) Repetimos 25 veces desde el punto 8.

Como ven tenemos que ir guardando el total de casos de cada día, para recuperarlos 5 días después. Python tiene una estructura de datos que nos puede ayudar, las listas.

```
# Parámetros iniciales
Contactos = 10
Probabilidad = 0.015
Incubacion = 5
dias = 30
# Valores iniciales
Anterior = 10
Total = Anterior
# Cola
Cola=[]

# Calculos
for dia in range(Incubacion):
    Nuevos = Anterior * Contactos * Probabilidad
    Total += Nuevos
    Anterior=Total
    Cola.append(Total)
    print ("Dia:", dia+1)
    print (" Infectados:", int(Total))

for dia in range(Incubacion, dias):
    Nuevos = Anterior * Contactos * Probabilidad
    Total += Nuevos
```

```

Anterior=Total
Cola.append(Total)
Confirmados = Cola.pop(0)
print ("Dia:",dia+1)
print ("  Infectados:",int(Total))
print("  Confirmados:" , int(Confirmados))

```

### TAREA:

3) Investigar los tipos de datos que puede utilizar python.

Por último veamos el efecto de la cuarentena. Cuando se declara la cuarentena, se reduce la cantidad de contactos con otras personas al 20% (2 contactos por persona por día). En este caso queremos que el usuario pueda ingresar el día que se inicia la cuarentena.

Veamos el algoritmo:

- 1) Inicializamos los valores.
- 2) Se ingresa el día en que se inicia la cuarentena.
- 3) Si el día coincide con el inicio de la cuarentena, entonces se reduce la cantidad de contactos al 20%
- 4) Calculamos la cantidad de infectados del nuevo día.
- 5) Actualizamos el total de infectados
- 6) Guardamos temporalmente el total de infectados en una lista.
- 7) Actualizamos el total de infectados del día anterior (para realizar los cálculos del nuevo día).
- 8) Mostramos los resultados del día.
- 9) Repetimos 5 veces desde el punto 2.
- 10) Si el día coincide con el inicio de la cuarentena, entonces se reduce la cantidad de contactos al 20%
- 11) Calculamos la cantidad de infectados del nuevo día.
- 12) Guardamos temporalmente el total de infectados en una lista.
- 13) Actualizamos el total de infectados del día anterior (para realizar los cálculos del nuevo día).
- 14) Recuperamos el total de infectados de hace 5 días (casos confirmados).
- 15) Mostramos los resultados del día.
- 16) Repetimos 25 veces desde el punto 10.

Antes de ver el código, fíjense en el algoritmo y el código anterior, hay algunas partes que se repiten. En la implementación (y en el diseño de la solución, o algoritmo) podemos reducir la cantidad de código al reutilizar las partes que se repiten.

```

# Parámetros iniciales
Contactos = 10
ContactosCuarentena = 2
Probabilidad = 0.015
Incubacion = 5
dias = 30
# Valores iniciales
Anterior = 10
Total = Anterior
# Cola
Cola=[]
#Ingreso de valores iniciales
Cuarentena = int(input("Ingresa el día de inicio de la cuarentena:"))

def CalculoInfectados(Total,Anterior>Contactos,Probabilidad):

```

```

    Nuevos = Anterior * Contactos * Probabilidad
    Total += Nuevos
    return Total

# Calculos
for dia in range(Incubacion):
    if dia==Cuarentena:
        Contactos=ContactosCuarentena
    Total = CalculoInfectados(Total, Anterior, Contactos, Probabilidad)
    Anterior=Total
    Cola.append(Total)
    print ("Dia:",dia+1)
    print ("    Infectados:",int(Total))

for dia in range(Incubacion,dias):
    if dia==Cuarentena:
        Contactos=ContactosCuarentena
    Total = CalculoInfectados(Total, Anterior, Contactos, Probabilidad)
    Anterior=Total
    Cola.append(Total)
    Confirmados = Cola.pop(0)
    print ("Dia:",dia+1)
    print ("    Infectados:",int(Total))
    print ("    Confirmados:" , int(Confirmados))

```

**Felicidades, acaban de terminar sus primeros programas en Python.**

#### **TAREA:**

- 3) Investigar la diferencia entre "=" y "==".
- 4) Investigar la función "input ()".
- 5) Con lo visto en este apunte: realizar el algoritmo y el programa para ayudar al pintor con su presupuesto. El programa debe pedir las dimensiones de una habitación, y calcular la superficie total de las paredes. Opcionalmente puede pedir el rendimiento de la pintura en metros cuadrados cubiertos por litro, calcular e informar la cantidad de litros requeridos.



(CC)Luis Pérez – 2021

<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.es>