

CURSO

Python aplicado a finanzas



MÓDULO 1

¿Qué es y para qué **programar**?

- Por medio del lenguaje de programación le damos tareas a las computadoras
- Cada día más dispositivos necesitan ser programados
- Mercado laboral con demanda en aumento



¿Qué es un **algoritmo**?

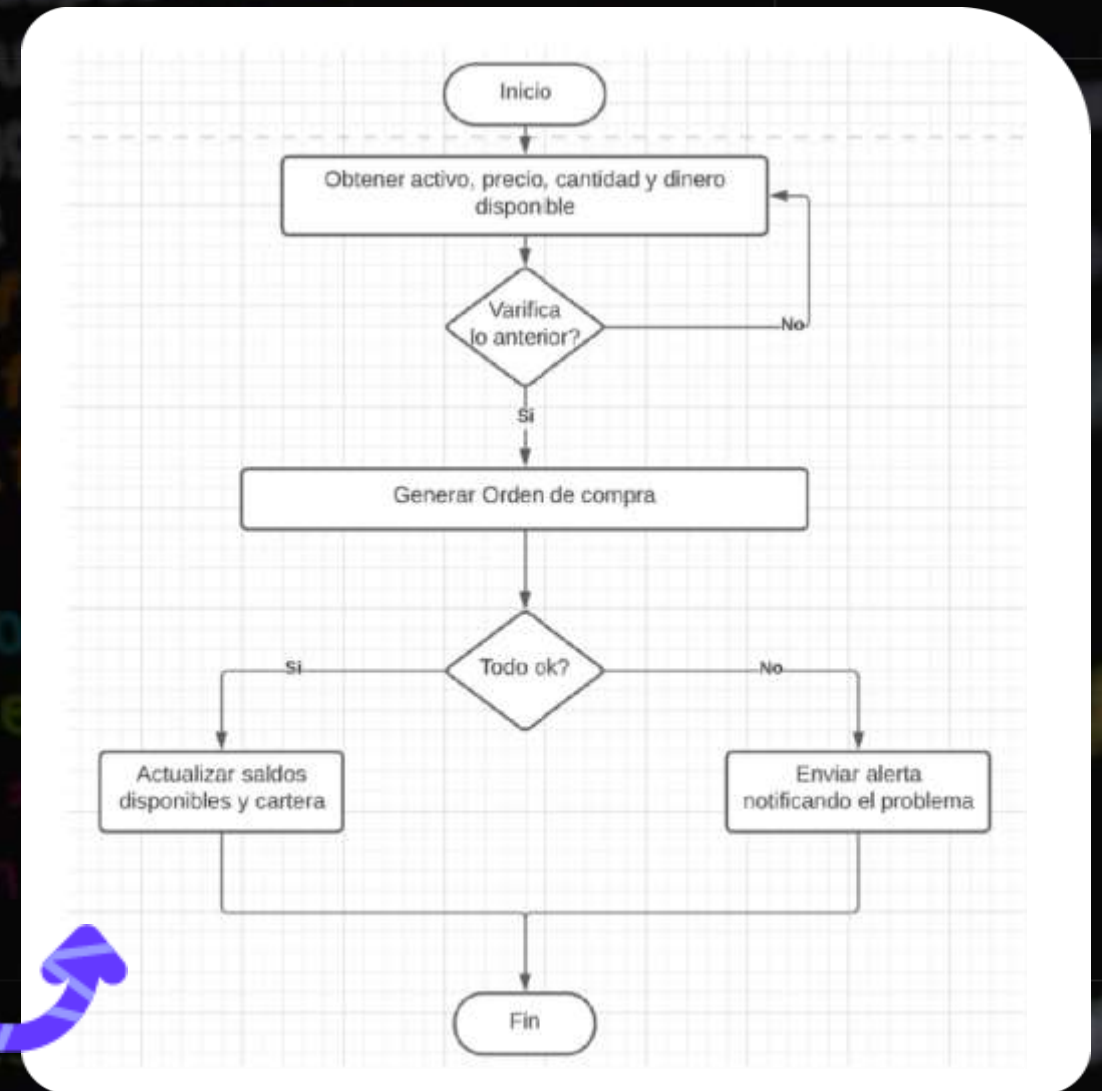
- Conjunto de instrucciones específicas y ordenadas para obtener un resultado concreto.
- Si por ejemplo quiero comprar un activo:



1. Definir activo, precio, cantidad y dinero disponible.
2. ¿Está definido el activo, el precio, la cantidad y dispongo del dinero suficiente?
 - a. Si: Crear la orden de compra.
 - b. No: Volver al punto 1.
3. ¿Todo ok?
 - a. Si: Actualizar saldos disponibles y cartera.
 - b. No: Enviar alerta avisando fallo y motivo.

Diagrama de flujo

- Es la representación gráfica de un algoritmo



Computador

PARTES EXTERNAS

- Monitor
- Armazon: Carcasa o gabinete del CPU.
- Teclado.
- Mouse.
- Puertos y conectores.
- Parlantes.
- Micrófono.
- Cámara Web

PARTES INTERNAS

- Procesador (CPU).
- Placa madre.
- Memoria RAM.
- Memoria ROM.
- Discos.
- Placa Base o placa madre.
- Tarjetas
(Video, sonido, red)



Computador

a



Vínculo entre partes (Externas e internas)

- BIOS
- Sistema Operativo (OS)
 - ✓ Windows
 - ✓ Linux
 - ✓ MacOS

Python



- Es un lenguaje de programación interpretado, dinámico y multiplataforma.
- Su filosofía hace hincapié en la legibilidad de su código.
- Es multiparadigma ya que:
 - Soporta parcialmente la orientación a objetos.
 - Programación imperativa.
 - En menor medida, programación funcional.

Herramientas de **trabajo**

Collaboratory:

- Permite ejecutar y programar en Python en tu navegador sin requerir configuración y se guarda en la nube.
- Da acceso gratuito a GPUs y TPUs.
- Permite compartir contenido fácilmente.

repl.it:

Un entorno de desarrollo online como colab.

Spyder:

Entorno de desarrollo integrado multiplataforma de código abierto para programación científica en Python.

PyCharm:

Entorno de desarrollo integrado para programación informática, para Python.

Herramientas de **trabajo**

Jupyter Notebook:

Entorno informático interactivo basado en la web para crear documentos de Jupyter Notebook:

- Es un documento JSON.
- Sigue un esquema versionado.
- Contiene una lista ordenada de celdas de entrada/salida que pueden contener.

Dentro de anaconda vienen instalados **Spyder** y **Jupyter Notebook**.

<https://repl.it/languages/python3>

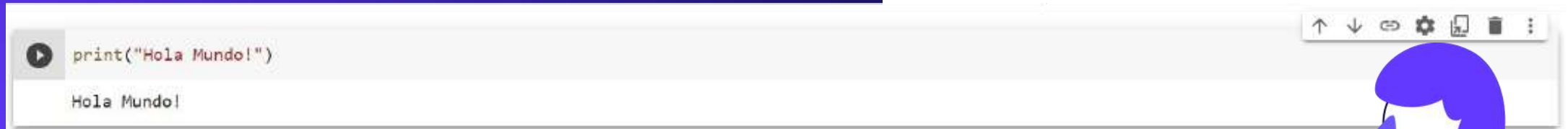
<https://colab.research.google.com/>

<https://jupyter.org/try>

<https://www.anaconda.com/products/individual>

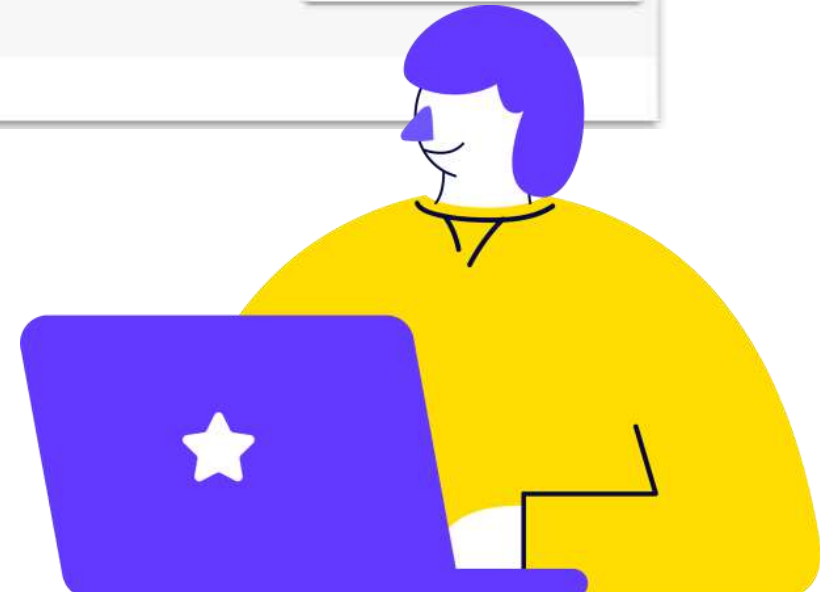
Nuestro Primer Programa **Hola Mundo**

- Primer programa en Python:



A screenshot of a Python terminal window. The window has a title bar with standard icons (up, down, left, right, search, settings, and a menu). The terminal content shows a prompt character followed by the command `print("Hola Mundo!")` on the first line. The second line shows the output `Hola Mundo!`.

```
print("Hola Mundo!")  
Hola Mundo!
```



¿Qué son las **variables**?

- Ingreso de valores por teclado por medio de la función **input()**

```
[2] nombre = "Ignacio Guardines"
```

```
[1] precio = 174.4  
precio
```

```
174.4
```

```
[3] cantidad_nominales_ypfd = 100
```

```
ingresenombre = input()
```



¿Qué son y cuáles son los tipos de **datos básicos**?

- Los tipos de datos son un conjunto de símbolos y operaciones que se pueden hacer con dichos símbolos:

Números:

- Números enteros
- Números flotantes

Valores de verdad (Booleanos)

- True
- False

Cadena de caracteres o Strings (palabras)

```
[5] marketId = "NYSE"  
    ordType = "Limit"  
    side = "Buy"  
  
    price = 187.25  
    orderQty = 1000
```

```
[6] variable1, variable2 = "Hola", 2
```

Segundo Programa en Python

- Uso de la función **print()**
- Concatenar palabras
- Otras funciones para manipular
Cadenas de Caracteres

```
[ ] palabra = "hola" + " Mundo!"  
    print(palabra)
```

```
hola Mundo!
```



Operadores y

Operaciones

Operadores y operaciones aritméticas:

Entorno informático interactivo basado en la web para crear documentos de **Jupyter Notebook**:

- "+" **Suma** ($10 + 5 = 15$)
- "-" **Resta** ($10 - 5 = 5$)
- "*" **Multiplicación** ($20 * 4 = 80$)
- "/" **División** ($21 / 5 = 4.2$)
- "%" **Resto de la división**, también conocido como "módulo" ($21 \% 5 = 1$)
- "**" **Potencia** ($12 ** 2 = 144$)
- "// **División entera** (Sin decimales) ($22 // 5 = 4$; $22.0 // 5.0 = 4.0$)

```
[ ] resultado = 1000 + 0.15
```

```
[ ] resultado
```

```
1000.15
```

```
[ ] saldo_disponible = 10000 - 220  
saldo_disponible
```

```
9780
```

```
[ ] saldo_disponible = saldo_disponible - 262
```

```
[ ] saldo_disponible
```

```
9518
```

```
[ ] contador = 0  
contador = contador + 1  
contador = contador + 1  
contador = contador + 1  
contador = contador + 1  
contador
```

```
4
```

```
[ ] # Multiplicacion  
resultado = 2 * 3 * 2  
resultado
```

```
12
```

```
[ ] resultado8 = resultado / 5
```

```
[ ] resultado8
```

```
2.4
```

```
[ ] resultado // 5
```

```
2
```

Operadores y Operaciones lógicas

- **or**: Devuelve un valor de verdad, es verdadero si al menos uno es verdadero.
- **and**: Devuelve un valor de verdad, es verdadero si ambas variables son verdaderas.
- **not**: Devuelve un valor de verdad, invierte el valor de verdad de la variable a la que afecta.

```
[ ] esHorarioMercado = True  
    esLunes = True  
    esLunes and esHorarioMercado
```

```
True
```

```
[ ] not esLunes
```

```
False
```

```
[ ] tengoMargen = True  
    compreDolarFuturo = False  
    tengoMargen and not compreDolarFuturo
```

```
True
```

```
[ ] # Tabla de verdad  
    # Y logico True cuando todo es True.  
    # O logico True cuando al menos uno es True.  
    # Not invierte el valor de verdad sobre el que opera.
```

Operadores de relaciones

- Son símbolos que se usan para comparar dos valores.
- Si el resultado es correcto la expresión es verdadera, caso contrario es falsa.
- ">" **Mayor estricto**, devuelve un valor de verdad resultado si es o no mayor el contenido de la primer variable respecto de la segunda, devuelve True si es así, False caso contrario.
- "<" **Menor estricto**, devuelve un valor de verdad resultado si es o no menor el contenido de la primer variable respecto de la segunda, devuelve True si es así, False caso contrario.
- ">=" **Mayor o igual**, similar al mayor estricto, agrega la igualdad.
- "<=" **Menor o igual**, similar al menor estricto, agrega la igualdad.
- "==" **Igualdad**, devuelve True si ambas variables son iguales, False caso contrario.
- "!=" **Distinto**, devuelve True si ambas variables son distintas, False caso contrario.

```
[ ] 10 < 12
```

```
True
```

```
[ ] 16 != 15
```

```
True
```


Estilos de escritura para nombres de variables de más de una palabra

- camelCase
- snake_case
- PascalCase
- CONSTANTS_STYLE



Operadores de Asignación

Operador	Ejemplo	Equivalencia
<code>+=</code>	<code>x += 2</code>	<code>x = x + 2</code>
<code>-=</code>	<code>x -= 2</code>	<code>x = x - 2</code>
<code>*=</code>	<code>x *= 2</code>	<code>x = x * 2</code>
<code>/=</code>	<code>x /= 2</code>	<code>x = x / 2</code>
<code>%=</code>	<code>x %= 2</code>	<code>x = x % 2</code>
<code>//=</code>	<code>x //= 2</code>	<code>x = x // 2</code>
<code>**=</code>	<code>x **= 2</code>	<code>x = x ** 2</code>

