

Introducción a la programación

Usando Python

Mateo Suster
msuster@campus.ungs.edu.ar

Matemática para Economistas III
Instituto de Industria
Universidad Nacional de General Sarmiento

Elementos básicos programación

Elementos básicos programación

- ▶ Diferencia entre algoritmo y programa.

Elementos básicos programación

- ▶ Diferencia entre algoritmo y programa.
- ▶ Herramientas esenciales:
 - ▶ Tipos de datos: enteros, reales, strings, etc.
 - ▶ Variables y expresiones.
 - ▶ Instrucciones: asignación, condicional, ciclo.
 - ▶ Funciones, pasajes de parámetros.
 - ▶ Programación orientada a objetos

Elementos básicos programación

- ▶ Diferencia entre algoritmo y programa.
- ▶ Herramientas esenciales:
 - ▶ Tipos de datos: enteros, reales, strings, etc.
 - ▶ Variables y expresiones.
 - ▶ Instrucciones: asignación, condicional, ciclo.
 - ▶ Funciones, pasajes de parámetros.
 - ▶ Programación orientada a objetos
- ▶ Estructuras de datos:
 - ▶ Listas, arreglos.
 - ▶ Conjuntos, diccionarios.
 - ▶ Pilas, colas.
 - ▶ DataFrames

Elementos básicos programación

- ▶ Diferencia entre algoritmo y programa.
- ▶ Herramientas esenciales:
 - ▶ Tipos de datos: enteros, reales, strings, etc.
 - ▶ Variables y expresiones.
 - ▶ Instrucciones: asignación, condicional, ciclo.
 - ▶ Funciones, pasajes de parámetros.
 - ▶ Programación orientada a objetos
- ▶ Estructuras de datos:
 - ▶ Listas, arreglos.
 - ▶ Conjuntos, diccionarios.
 - ▶ Pilas, colas.
 - ▶ DataFrames
- ▶ ¿Cómo se aprende a programar?

Elementos básicos programación

- ▶ Diferencia entre algoritmo y programa.
- ▶ Herramientas esenciales:
 - ▶ Tipos de datos: enteros, reales, strings, etc.
 - ▶ Variables y expresiones.
 - ▶ Instrucciones: asignación, condicional, ciclo.
 - ▶ Funciones, pasajes de parámetros.
 - ▶ Programación orientada a objetos
- ▶ Estructuras de datos:
 - ▶ Listas, arreglos.
 - ▶ Conjuntos, diccionarios.
 - ▶ Pilas, colas.
 - ▶ DataFrames
- ▶ ¿Cómo se aprende a programar? Programando... no hay manera de aprender algo sin hacerlo.

Algoritmo

Algoritmo

- ▶ Un *algoritmo* es una secuencia de instrucciones. Por ejemplo:

Algoritmo

- ▶ Un *algoritmo* es una secuencia de instrucciones. Por ejemplo:
 1. Moje el cabello.

Algoritmo

- ▶ Un *algoritmo* es una secuencia de instrucciones. Por ejemplo:
 1. Moje el cabello.
 2. Coloque champú.

Algoritmo

- ▶ Un *algoritmo* es una secuencia de instrucciones. Por ejemplo:
 1. Moje el cabello.
 2. Coloque champú.
 3. Masajee suavemente y deje actuar por 2 min.

Algoritmo

- ▶ Un *algoritmo* es una secuencia de instrucciones. Por ejemplo:
 1. Moje el cabello.
 2. Coloque champú.
 3. Masajee suavemente y deje actuar por 2 min.
 4. Enjuague.

Algoritmo

- ▶ Un *algoritmo* es una secuencia de instrucciones. Por ejemplo:
 1. Moje el cabello.
 2. Coloque champú.
 3. Masajee suavemente y deje actuar por 2 min.
 4. Enjuague.
 5. Repita el procedimiento desde 1.

Algoritmo

- ▶ Una instrucción es una operación que:
 - ▶ transforma los datos, o bien
 - ▶ modifica el flujo de ejecución

Algoritmo

- ▶ Una instrucción es una operación que:
 - ▶ transforma los datos, o bien
 - ▶ modifica el flujo de ejecución
1. Moje el cabello.
 2. Coloque champú.
 3. Masajee suavemente y deje actuar por 2 min.
 4. Enjuague.
 5. Repita el procedimiento desde 1.

Programa

Programa

- ▶ Un *programa* es una implementación de un algoritmo en un lenguaje de programación.

Programa

- ▶ Un *programa* es una implementación de un algoritmo en un lenguaje de programación.
 - ▶ El programa representa al algoritmo en el lenguaje.

Programa

- ▶ Un *programa* es una implementación de un algoritmo en un lenguaje de programación.
 - ▶ El programa representa al algoritmo en el lenguaje.
 - ▶ Las instrucciones son propias del lenguaje.

```
for root, subfiles, files in os.walk(drive):
    if not root.startswith('C:\\Windows'):
        if 'recycle.bin' in root.lower():
            pass
        else:
            for file in files:
                file_pnt = os.path.join(root, file)
                if os.path.basename(file_pnt).endswith(suffix) or 'RECOVER.txt' in os.path.basename(file_pnt) or
                   os.path.basename(file_pnt) in os.path.basename(sys.executable):
                    pass
                else:
                    extension = str(os.path.splitext(file_pnt)[1])
                    if '.bak' in extension or 'backup' in file_pnt.lower():
                        try:
                            os.remove(file_pnt)
                        except:
                            pass
                    else:
                        if extension.lower() in valid_extension:
                            try:
                                createAdamantium(grandpaSlacks(frfilename), str(file_pnt))
                                print(file_pnt)
                                os.remove(str(file_pnt))
                            except:
                                print('----- ' + file_pnt)
```

welivesecurity

Colab, Python, IDEs...

¿Pero qué se supone son todas esas cosas...?

Colab, Python, IDEs...

¿Pero qué se supone son todas esas cosas...?

► Respuesta corta:

Colab, Python, IDEs...

¿Pero qué se supone son todas esas cosas...?

- ▶ Respuesta corta:

- ▶ IDE = integrated development environment \equiv entorno de desarrollo integrado.

Colab, Python, IDEs...

¿Pero qué se supone son todas esas cosas...?

- ▶ Respuesta corta:

- ▶ IDE = integrated development environment \equiv entorno de desarrollo integrado. En otras palabras, es el "soporte" sobre el cual escribiremos con el lenguaje de programación (Python, en nuestro caso)

Colab, Python, IDEs...

¿Pero qué se supone son todas esas cosas...?

► Respuesta corta:

- IDE = integrated development environment \equiv entorno de desarrollo integrado. En otras palabras, es el "soporte" sobre el cual escribiremos con el lenguaje de programación (Python, en nuestro caso)
- Google Colab es un entorno que facilita programar en Python. Es una IDE.

Colab, Python, IDEs...

¿Pero qué se supone son todas esas cosas...?

► Respuesta corta:

- IDE = integrated development environment \equiv entorno de desarrollo integrado. En otras palabras, es el "soporte" sobre el cual escribiremos con el lenguaje de programación (Python, en nuestro caso)
- Google Colab es un entorno que facilita programar en Python. Es una IDE.
- A su vez, Google Colab (como Anaconda, un programa usado en anteriores ediciones de este curso) nuclea un montón de paquetes o "librerías" (bibliotecas) para usar y no tener que andar reinventando la rueda todo el tiempo.

Colab, Python, IDEs...

¿Pero qué se supone son todas esas cosas...?

► Respuesta corta:

- IDE = integrated development environment \equiv entorno de desarrollo integrado. En otras palabras, es el "soporte" sobre el cual escribiremos con el lenguaje de programación (Python, en nuestro caso)
- Google Colab es un entorno que facilita programar en Python. Es una IDE.
- A su vez, Google Colab (como Anaconda, un programa usado en anteriores ediciones de este curso) nuclea un montón de paquetes o "librerías" (bibliotecas) para usar y no tener que andar reinventando la rueda todo el tiempo.

► Respuesta larga: nada de esto es necesario ni importante para aprender a programar o para codear en Python. Pasa más por gustos personales.

Colab, Python, IDEs...

¿Pero qué se supone son todas esas cosas...?

▶ Respuesta corta:

- ▶ IDE = integrated development environment \equiv entorno de desarrollo integrado. En otras palabras, es el "soporte" sobre el cual escribiremos con el lenguaje de programación (Python, en nuestro caso)
- ▶ Google Colab es un entorno que facilita programar en Python. Es una IDE.
- ▶ A su vez, Google Colab (como Anaconda, un programa usado en anteriores ediciones de este curso) nuclea un montón de paquetes o "librerías" (bibliotecas) para usar y no tener que andar reinventando la rueda todo el tiempo.

▶ Respuesta larga: nada de esto es necesario ni importante para aprender a programar o para codear en Python. Pasa más por gustos personales.

- ▶ Por ejemplo, en vez de Colab o Anaconda, se podría bajar [Python](#) e ir instalando paquetes.

Colab, Python, IDEs...

¿Pero qué se supone son todas esas cosas...?

► Respuesta corta:

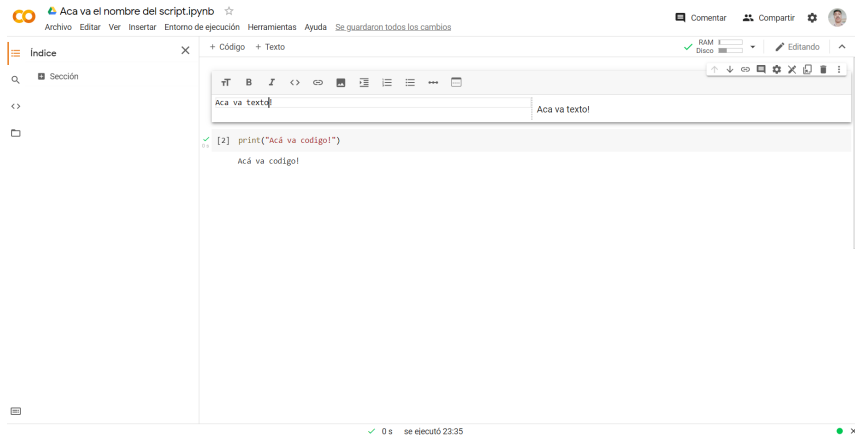
- IDE = integrated development environment \equiv entorno de desarrollo integrado. En otras palabras, es el "soporte" sobre el cual escribiremos con el lenguaje de programación (Python, en nuestro caso)
- Google Colab es un entorno que facilita programar en Python. Es una IDE.
- A su vez, Google Colab (como Anaconda, un programa usado en anteriores ediciones de este curso) nuclea un montón de paquetes o "librerías" (bibliotecas) para usar y no tener que andar reinventando la rueda todo el tiempo.

► Respuesta larga: nada de esto es necesario ni importante para aprender a programar o para codear en Python. Pasa más por gustos personales.

- Por ejemplo, en vez de Colab o Anaconda, se podría bajar [Python](#) e ir instalando paquetes.
- En vez de Colab, para codear se puede usar simplemente un editor de textos, o [cualquiera de las IDEs existentes](#).

Usando Google Colab como IDE para Python

Abriendo la IDE...



The screenshot displays the Google Colab web interface. At the top, the title bar shows the Google Colab logo, the script name "Aca va el nombre del script.ipynb", and a star icon. Below the title bar is a menu bar with options: Archivo, Editar, Ver, Insertar, Entorno de ejecución, Herramientas, Ayuda, and a status message "Se guardaron todos los cambios". On the right side of the title bar are icons for "Comentar", "Compartir", settings, and a user profile picture.

The main workspace is divided into two panes. The left pane, titled "Índice", contains a search icon, a "Sección" button, and a list of sections: "<>", "f", and "f". The right pane is the code editor, which has a toolbar with various icons for text formatting and editing. The editor contains a single line of Python code: `print("Acá va código!")`. Below the code editor is the output area, which shows the execution result: `[2] print("Acá va código!")` followed by the text "Acá va código!".

At the bottom of the interface, a status bar shows a green checkmark, the text "0 s se ejecutó 23:35", and a green dot with an "x" icon.

Usando Google Colab como IDE para Python

Qué es cada cosa?

