



El futuro digital  
es de todos

MinTIC

# Misión TIC2022

SEMANA 4

Diego Iván Oliveros Acosta



**UNIVERSIDAD  
DE ANTIOQUIA**  
Facultad de Ingeniería



## CICLO 1


[FORMACIÓN POR CICLOS]

# Fundamentos de Programación

Presentación del curso





A close-up photograph of several bees on a golden honeycomb. The bees are in various positions, some facing the camera and others with their backs to it. The honeycomb cells are hexagonal and filled with a golden substance. The lighting is bright, highlighting the textures of the bees' bodies and the honeycomb.

# **Módulos, paquetes, bibliotecas y frameworks de Python**

Diego Iván Oliveros Acosta

# Agenda

*Semana 4 -1*



**El modulo**

**El paquete**

**La biblioteca**

**El framework**

**Resumen**

**Ejercicios**

**Biblografia**





# Módulos

- *¿Cómo crear un script?*
- *¿Qué pasa si el Código crece?*
- Instancia del intérprete
- variable global `__name__`



*“las definiciones de un módulo pueden ser importadas a otros módulos o al módulo principal”*

Archivo .py que tiene código / instrucción ejecutable de Python.

```
def welcome_message(user_name):  
    return "Welcome " + user_name  
  
import modulo  
print (modulo.welcome_message("Teacher Diego"))
```

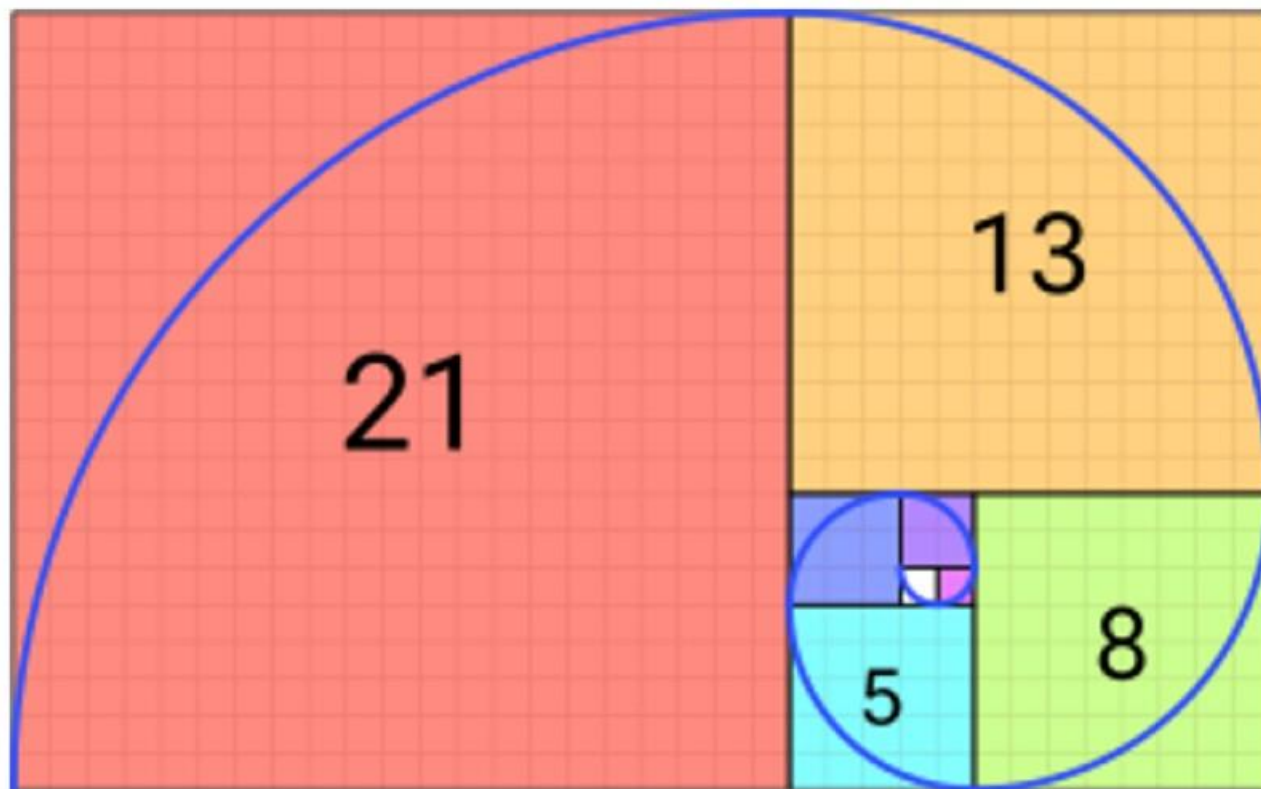


# Ejercicio:

Desarrolle las funciones que permitan calcular la serie de Fibonacci como enteros y luego como lista.

Luego en un archivo aparte, invoque las funciones creadas para calcular la serie de 1000 y 100 respectivamente.

Asígnela a un nombre local de manera que se pueda usar frecuentemente en el código.





# Sintaxis de declaración de los **módulos**

```
from fibo import fib, fib2

from fibo import *

import fibo as fib

from fibo import fib as Fibonacci

from . import echo
from .. import formats
from ..filters import equalizer
```

## Nota






Por razones de eficiencia, cada módulo es importado solo una vez por sesión del intérprete.

Por lo tanto, si cambias tus módulos, **debes reiniciar el intérprete**. Si es un solo módulo que quieres probar de forma interactiva, usa [`importlib.reload\(\)`](#),

## Por ejemplo:

```
import importlib; importlib.reload(modulename)
```

GitHub > Programaci-n-C1-Grupo-9- > Código Clase > S4 2022 05 10

Nombre	Fecha de modificación	Tipo	Tamaño
 __pycache__	10/05/2022 11:04 a. m.	Carpeta de archivos	
 fibo.py	10/05/2022 11:01 a. m.	Archivo de origen ...	1 KB
 fiboimp.py	10/05/2022 11:15 a. m.	Archivo de origen ...	1 KB
 modulo.py	10/05/2022 10:50 a. m.	Archivo de origen ...	1 KB
 usemodulo.py	10/05/2022 10:55 a. m.	Archivo de origen ...	1 KB





## Paquetes

- Una colección de módulos (un «paquete»)
- Estructurar el espacio de nombres de módulos

+++

- El nombre del módulo A.B designa un submódulo B en un paquete llamado A.
- Contiene un archivo especial `__init__.py`



## Archivo especial `__init__.py`

```
user/ # top level package
__init__.py
get/ # first subpackage
    __init__.py
    info.py
    points.py
    transactions.py
create/ # second subpackage
    __init__.py
    api.py
    platform.py
```





## Posible estructura Expresada en términos de un sistema jerárquico de archivos:

```
sound/                                Top-level package
  __init__.py                         Initialize the sound package
  formats/                           Subpackage for file format conversions
    __init__.py
    wavread.py
    wavwrite.py
    aiffread.py
    aiffwrite.py
    auread.py
    auwrite.py
    ...
```

```
effects/                             Subpackage for sound effects
  __init__.py
  echo.py
  surround.py
  reverse.py
  ...
  filters/                           Subpackage for filters
    __init__.py
    equalizer.py
    vocoder.py
    karaoke.py
    ...
```



## Forma clásica de importación de paquetes

+++

- `from user.get import info`
- `# imports info module from get package`
- `from user.create import api`
- `#imports api module from create package`

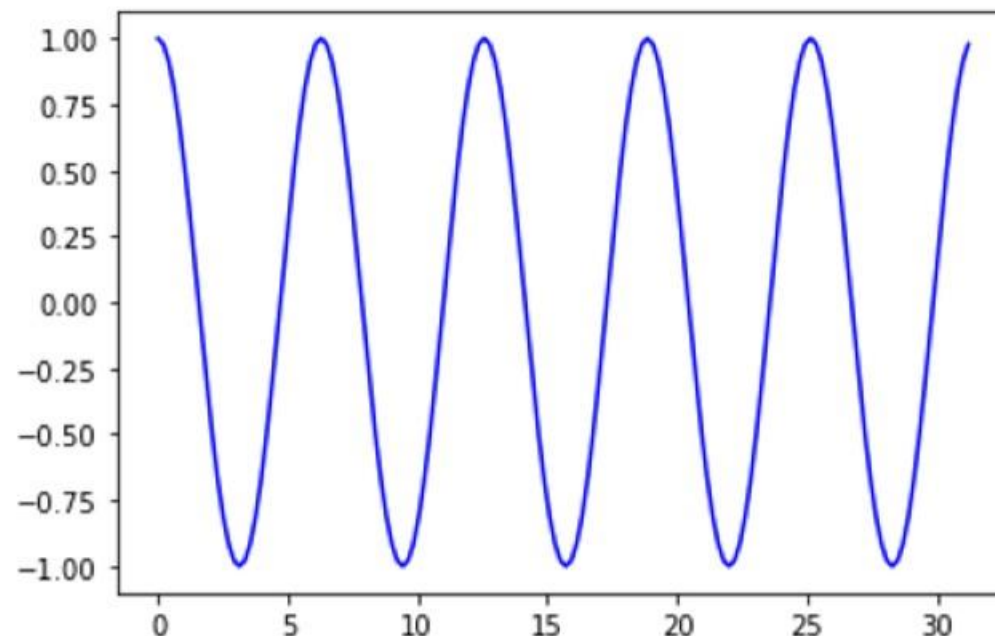




# Ejercicio:

Utilizando la librería  
numpy

Crea una onda coseno y  
représentalá gráficamente.



```
numpy.arange([start, ]stop, [step, ] dtype=None, *, like=None)
```

@scalapp.co

# Python

## *Bibliotecas y frameworks*



*Es una colección de varios paquetes.*

No hay diferencia entre el paquete y la biblioteca de Python conceptualmente.

*Es una colección de varias bibliotecas que crea el flujo de código.*

Tomemos un ejemplo de [Django](#)

**Tiene varias bibliotecas integradas:**

**Auth, user, database connector etc.**



# Referencias

<https://github.com/DiegOliveros/MISSION-TIC-2020>

<https://nbviewer.jupyter.org/>

<https://rise.readthedocs.io/en/stable/installation.html>

<https://jupyter.org/try>

<https://docs.python.org/>

<https://pythontutor.com/>

<https://www.jetbrains.com/es-es/pycharm/download/#section=windows>

<https://www.facebook.com/Scalapp.co>

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