

Introducción

Candelaria Alvarez y Christian Gimenez

31 oct 2020



- 1 ¡Empecemos!
 - Primeros pasos
- 2 Salida y variables
 - Mostramos las notas
 - ¡Quiero escuchar la escala!
 - Microactividad 1
 - Consejos 1
- 3 ¿Preguntas?
 - ¿Preguntas?
- 4 Licencia
 - Licencia de Esta Obra

- 1 ¡Empecemos!
- 2 Salida y variables
- 3 ¿Preguntas?
- 4 Licencia

- 1 ¡Empecemos!
 - Primeros pasos

¿Qué debe hacer el programa?

Recordemos la escala:

Do Re Mi Fa Sol La Si Do

¿cómo sonará?

¿Que debe hacer el programa?

- Mostrar la escala en pantalla
- ¡Quiero escucharla!

¿Qué aprenderemos?

- Salida
- Variables

¿Qué debe hacer el programa?

Recordemos la escala:

Do Re Mi Fa Sol La Si Do

¿cómo sonará?

¿Que debe hacer el programa?

- Mostrar la escala en pantalla
- ¡Quiero escucharla!

¿Qué aprenderemos?

- Salida
- Variables

¿Qué debe hacer el programa?

Recordemos la escala:

Do Re Mi Fa Sol La Si Do

¿cómo sonará?

¿Que debe hacer el programa?

- Mostrar la escala en pantalla
- ¡Quiero escucharla!

¿Qué aprenderemos?

- Salida
- Variables

¿Qué debe hacer el programa?

Recordemos la escala:

Do Re Mi Fa Sol La Si Do

¿cómo sonará?

¿Que debe hacer el programa?

- Mostrar la escala en pantalla
- ¡Quiero escucharla!

¿Qué aprenderemos?

- Salida
- Variables

¿Qué debe hacer el programa?

Recordemos la escala:

Do Re Mi Fa Sol La Si Do

¿cómo sonará?

¿Que debe hacer el programa?

- Mostrar la escala en pantalla
- ¡Quiero escucharla!

¿Qué aprenderemos?

- Salida
- Variables

¿Qué debe hacer el programa?

Recordemos la escala:

Do Re Mi Fa Sol La Si Do

¿cómo sonará?

¿Que debe hacer el programa?

- Mostrar la escala en pantalla
- ¡Quiero escucharla!

¿Qué aprenderemos?

- Salida
- Variables

¿Qué debe hacer el programa?

Recordemos la escala:

Do Re Mi Fa Sol La Si Do

¿cómo sonará?

¿Que debe hacer el programa?

- Mostrar la escala en pantalla
- ¡Quiero escucharla!

¿Qué aprenderemos?

- Salida
- Variables

- 1 ¡Empecemos!
- 2 Salida y variables**
- 3 ¿Preguntas?
- 4 Licencia

2 Salida y variables

- Mostramos las notas
- ¡Quiero escuchar la escala!
- Microactividad 1
- Consejos 1

Primero mostramos las notas

- Importamos lo que necesitamos
 - `tkinter`
 - `showinfo` con otro nombre: `mostrar_info`
- Escribimos la escala directamente.
- Mostramos la escala con `showinfo`.

Primero mostramos las notas

- Importamos lo que necesitamos
 - `tkinter`
 - `showinfo` con otro nombre: `mostrar_info`
- Escribimos la escala directamente.
- Mostramos la escala con `showinfo`.

Primero mostramos las notas

- Importamos lo que necesitamos
 - tkinter
 - `showinfo` con otro nombre: `mostrar_info`
- Escribimos la escala directamente.
- Mostramos la escala con `showinfo`.

Primero mostramos las notas

- Importamos lo que necesitamos
 - tkinter
 - `showinfo` con otro nombre: `mostrar_info`
- Escribimos la escala directamente.
- Mostramos la escala con `showinfo`.

Primero mostramos las notas

- Importamos lo que necesitamos
 - tkinter
 - `showinfo` con otro nombre: `mostrar_info`
- Escribimos la escala directamente.
- Mostramos la escala con `showinfo`.

Mostramos las notas...

```
from tkinter import Tk
from tkinter.messagebox import showinfo as mostrar_info

root = Tk()
root.withdraw()

mostrar_info('¡La escala!', 'do re mi fa sol la si do')
```

¿Por qué root...?

root es la ventana principal. Con `withdraw()` la ocultamos.
Toda aplicación de Tk necesita de su intérprete y una ventana principal.

¿Por qué `from ... import`?

`from ... import` nos permite importar alguna función o clase desde una biblioteca.
Son elementos que se definen en otros archivos y que se pueden descargar e instalar.

Mostramos las notas...

```
from tkinter import Tk
from tkinter.messagebox import showinfo as mostrar_info

root = Tk()
root.withdraw()

mostrar_info('¡La escala!', 'do re mi fa sol la si do')
```

¿Por qué root...?

root es la ventana principal. Con `withdraw()` la ocultamos.
Toda aplicación de Tk necesita de su intérprete y una ventana principal.

¿Por qué from ... import?

`from ... import` nos permite importar alguna función o clase desde una biblioteca.
Son elementos que se definen en otros archivos y que se pueden descargar e instalar.

Mostramos las notas...

```
from tkinter import Tk
from tkinter.messagebox import showinfo as mostrar_info

root = Tk()
root.withdraw()

mostrar_info('¡La escala!', 'do re mi fa sol la si do')
```

¿Por qué root...?

root es la ventana principal. Con `withdraw()` la ocultamos.
Toda aplicación de Tk necesita de su intérprete y una ventana principal.

¿Por qué from ... import?

`from ... import` nos permite importar alguna función o clase desde una biblioteca.
Son elementos que se definen en otros archivos y que se pueden descargar e instalar.

¡A probar!

```
from tkinter import Tk
from tkinter.messagebox import showinfo as mostrar_info

root = Tk()
root.withdraw()

mostrar_info('¡La escala!', 'do re mi fa sol la si do')
```

¡Prueben algunos cambios!

- ¡Prueben cambiar la escala por otro contenido!
- ¡Prueben mostrar más de una ventana!

¡A probar!

```
from tkinter import Tk
from tkinter.messagebox import showinfo as mostrar_info

root = Tk()
root.withdraw()

mostrar_info('¡La escala!', 'do re mi fa sol la si do')
```

¡Prueben algunos cambios!

- ¡Prueben cambiar la escala por otro contenido!
- ¡Prueben mostrar más de una ventana!

¡Se repite mucho!

A medida que programamos...

Cuando usamos nuestra canción, sería tedioso:

- escribir la canción cada vez que la necesitamos
- cambiar las notas cada vez que aparece nuestra canción

¿Y si guardamos la canción con un nombre?

¿Cómo se guardan cosas?

¡Usaremos variables!

Primero, usaremos un nombre para nuestra variable: `cancion`

```
cancion = 'do re mi fa sol la si do'
```

¿Cómo se usan?

Simplemente, escribimos el nombre donde queremos usarlo:

```
mostrar_info('¡La escala!', cancion)
```

¡Se repite mucho!

A medida que programamos...

Cuando usamos nuestra canción, sería tedioso:

- escribir la canción cada vez que la necesitamos
- cambiar las notas cada vez que aparece nuestra canción

¿Y si guardamos la canción con un nombre?

¿Cómo se guardan cosas?

¡Usaremos variables!

Primero, usaremos un nombre para nuestra variable: `cancion`

```
cancion = 'do re mi fa sol la si do'
```

¿Cómo se usan?

Simplemente, escribimos el nombre donde queremos usarlo:

```
mostrar_info('¡La escala!', cancion)
```

¡Se repite mucho!

A medida que programamos...

Cuando usamos nuestra canción, sería tedioso:

- escribir la canción cada vez que la necesitamos
- cambiar las notas cada vez que aparece nuestra canción

¿Y si guardamos la canción con un nombre?

¿Cómo se guardan cosas?

¡Usaremos variables!

Primero, usaremos un nombre para nuestra variable: `cancion`

```
cancion = 'do re mi fa sol la si do'
```

¿Cómo se usan?

Simplemente, escribimos el nombre donde queremos usarlo:

```
mostrar_info('¡La escala!', cancion)
```

¡Se repite mucho!

A medida que programamos...

Cuando usamos nuestra canción, sería tedioso:

- escribir la canción cada vez que la necesitamos
- cambiar las notas cada vez que aparece nuestra canción

¿Y si guardamos la canción con un nombre?

¿Cómo se guardan cosas?

¡Usaremos variables!

Primero, usaremos un nombre para nuestra variable: `cancion`

```
cancion = 'do re mi fa sol la si do'
```

¿Cómo se usan?

Simplemente, escribimos el nombre donde queremos usarlo:

```
mostrar_info('¡La escala!', cancion)
```

¡Se repite mucho!

A medida que programamos...

Cuando usamos nuestra canción, sería tedioso:

- escribir la canción cada vez que la necesitamos
- cambiar las notas cada vez que aparece nuestra canción

¿Y si guardamos la canción con un nombre?

¿Cómo se guardan cosas?

¡Usaremos variables!

Primero, usaremos un nombre para nuestra variable: `cancion`

```
cancion = 'do re mi fa sol la si do'
```

¿Cómo se usan?

Simplemente, escribimos el nombre donde queremos usarlo:

```
mostrar_info('¡La escala!', cancion)
```

¡Se repite mucho!

A medida que programamos...

Cuando usamos nuestra canción, sería tedioso:

- escribir la canción cada vez que la necesitamos
- cambiar las notas cada vez que aparece nuestra canción

¿Y si guardamos la canción con un nombre?

¿Cómo se guardan cosas?

¡Usaremos variables!

Primero, usaremos un nombre para nuestra variable: `cancion`

```
cancion = 'do re mi fa sol la si do'
```

¿Cómo se usan?

Simplemente, escribimos el nombre donde queremos usarlo:

```
mostrar_info('¡La escala!', cancion)
```

¡Se repite mucho!

A medida que programamos...

Cuando usamos nuestra canción, sería tedioso:

- escribir la canción cada vez que la necesitamos
- cambiar las notas cada vez que aparece nuestra canción

¿Y si guardamos la canción con un nombre?

¿Cómo se guardan cosas?

¡Usaremos variables!

Primero, usaremos un nombre para nuestra variable: `cancion`

```
cancion = 'do re mi fa sol la si do'
```

¿Cómo se usan?

Simplemente, escribimos el nombre donde queremos usarlo:

```
mostrar_info('¡La escala!', cancion)
```

¡Se repite mucho!

A medida que programamos...

Cuando usamos nuestra canción, sería tedioso:

- escribir la canción cada vez que la necesitamos
- cambiar las notas cada vez que aparece nuestra canción

¿Y si guardamos la canción con un nombre?

¿Cómo se guardan cosas?

¡Usaremos variables!

Primero, usaremos un nombre para nuestra variable: `cancion`

```
cancion = 'do re mi fa sol la si do'
```

¿Cómo se usan?

Simplemente, escribimos el nombre donde queremos usarlo:

```
mostrar_info('¡La escala!', cancion)
```

¿Qué cambiarían?

```
from tkinter import Tk
from tkinter.messagebox import showinfo as mostrar_info

root = Tk()
root.withdraw()

mostrar_info('¡La escala!', 'do re mi fa sol la si do')
```

¡A probar!

```
from tkinter import Tk
from tkinter.messagebox import showinfo as mostrar_info

root = Tk()
root.withdraw()

cancion = 'do re mi fa sol la si do'
mostrar_info('¡La escala!', cancion)
```

2 Salida y variables

- Mostramos las notas
- ¡Quiero escuchar la escala!
- Microactividad 1
- Consejos 1

¿La escuchamos?

Para escuchar cómo suena debemos:

```
Importar musica
```

```
from musica import tocar_cancion
```

```
¡Pedirle que la ejecute!
```

Al final, agregar:

```
tocar_cancion(cancion)
```

¡Pero no funciona!

¡Usemos el Bot de Juan!

En vez de `tocar_cancion(cancion)`, agregamos:

Opción 1: Copiamos a mano

Ehh... Pero... Me duelen los dedos...

Opción 2: Se copia automáticamente

Instalar pyperclip

```
from pyperclip import copy
# ...
copy(cancion)
```

¡Pero no funciona!

¡Usemos el Bot de Juan!

En vez de `tocar_cancion(cancion)`, agregamos:

Opción 1: Copiamos a mano

Ehh... Pero... Me duelen los dedos...

Opción 2: Se copia automáticamente

Instalar pyperclip

```
from pyperclip import copy
# ...
copy(cancion)
```

¡Pero no funciona!

¡Usemos el Bot de Juan!

En vez de `tocar_cancion(cancion)`, agregamos:

Opción 1: Copiamos a mano

Ehh... Pero... Me duelen los dedos...

Opción 2: Se copia automáticamente

Instalar `pyperclip`

```
from pyperclip import copy
# ...
copy(cancion)
```

¡Pero no funciona!

¡Usemos el Bot de Juan!

En vez de `tocar_cancion(cancion)`, agregamos:

Opción 1: Copiamos a mano

Ehh... Pero... Me duelen los dedos...

Opción 2: Se copia automáticamente

Instalar `pyperclip`

```
from pyperclip import copy
# ...
copy(cancion)
```

¡Pero no funciona!

¡Usemos el Bot de Juan!

En vez de `tocar_cancion(cancion)`, agregamos:

Opción 1: Copiamos a mano

Ehh... Pero... Me duelen los dedos...

Opción 2: Se copia automáticamente

Instalar `pyperclip`

```
from pyperclip import copy
# ...
copy(cancion)
```

¡Pero no funciona!

Opción 3: Mostrar con tkinter

```
from tkinter.simpledialog import askstring as pedir_dato
# ...
pedir_dato('Resultado', 'Presionar Ctrl-C', initialvalue=cancion)
```

Última opción: Imprimir a consola

Imprimimos a consola y copiamos desde ahí:

```
print(cancion)
```

¡Pero no funciona!

Opción 3: Mostrar con tkinter

```
from tkinter.simpledialog import askstring as pedir_dato
# ...
pedir_dato('Resultado', 'Presionar Ctrl-C', initialvalue=cancion)
```

Última opción: Imprimir a consola

Imprimimos a consola y copiamos desde ahí:

```
print(cancion)
```

2 Salida y variables

- Mostramos las notas
- ¡Quiero escuchar la escala!
- **Microactividad 1**
- Consejos 1

¿y si saludamos primero?

```
from tkinter import Tk
from tkinter.messagebox import showinfo as mostrar_info

root = Tk()
root.withdraw()

cancion = 'do re mi fa sol la si do'
mostrar_info('¡La escala!', cancion)
```

¿Y si primero saludamos?

- Intente agregar un saludo personalizado, algo como "¡Hola SU_NOMBRE!"
- Se puede agregar una breve explicación del programa.

¿y si saludamos primero?

```
from tkinter import Tk
from tkinter.messagebox import showinfo as mostrar_info

root = Tk()
root.withdraw()

cancion = 'do re mi fa sol la si do'
mostrar_info('¡La escala!', cancion)
```

¿Y si primero saludamos?

- Intente agregar un saludo personalizado, algo como "¡Hola SU_NOMBRE!"
 - Pruebe guardar su nombre en una variable y mostrarlo con 'Hola ' + nombre.
- Se puede agregar una breve explicación del programa.

¿y si saludamos primero?

```
from tkinter import Tk
from tkinter.messagebox import showinfo as mostrar_info

root = Tk()
root.withdraw()

cancion = 'do re mi fa sol la si do'
mostrar_info('¡La escala!', cancion)
```

¿Y si primero saludamos?

- Intente agregar un saludo personalizado, algo como "¡Hola SU_NOMBRE!"
 - Pruebe guardar su nombre en una variable y mostrarlo con 'Hola ' + nombre.
- Se puede agregar una breve explicación del programa.

¿y si saludamos primero?

```
from tkinter import Tk
from tkinter.messagebox import showinfo as mostrar_info

root = Tk()
root.withdraw()

cancion = 'do re mi fa sol la si do'
mostrar_info('¡La escala!', cancion)
```

¿Y si primero saludamos?

- Intente agregar un saludo personalizado, algo como "¡Hola SU_NOMBRE!"
 - Pruebe guardar su nombre en una variable y mostrarlo con 'Hola ' + nombre.
- Se puede agregar una breve explicación del programa.

2 Salida y variables

- Mostramos las notas
- ¡Quiero escuchar la escala!
- Microactividad 1
- Consejos 1

- Lo ideal es que todos los `from ... import` estén al principio.
- Tratar de no usar acentos en las variables.
- Usar guiones bajos o CamelCase en vez de espacios: `hola_mundo` o `holaMundo`.
 - Si se usa una forma, tomarla como convención y no cambiarla en la mitad del código.
- Siempre, **siempre** usar nombres de variables representativas. Prohibido usar `mn = "Pepe"` en vez de `mi_nombre = "Pepe"`.

Consejos 1

- Lo ideal es que todos los `from ... import` estén al principio.
- Tratar de no usar acentos en las variables.
- Usar guiones bajos o CamelCase en vez de espacios: `hola_mundo` o `holaMundo`.
 - Si se usa una forma, tomarla como convención y no cambiarla en la mitad del código.
- Siempre, **siempre** usar nombres de variables representativas. Prohibido usar `mn = "Pepe"` en vez de `mi_nombre = "Pepe"`.

Consejos 1

- Lo ideal es que todos los `from ... import` estén al principio.
- Tratar de no usar acentos en las variables.
- Usar guiones bajos o CamelCase en vez de espacios: `hola_mundo` o `holaMundo`.
 - Si se usa una forma, tomarla como convención y no cambiarla en la mitad del código.
- Siempre, **siempre** usar nombres de variables representativas. Prohibido usar `mn = "Pepe"` en vez de `mi_nombre = "Pepe"`.

- Lo ideal es que todos los `from ... import` estén al principio.
- Tratar de no usar acentos en las variables.
- Usar guiones bajos o CamelCase en vez de espacios: `hola_mundo` o `holaMundo`.
 - Si se usa una forma, tomarla como convención y no cambiarla en la mitad del código.
- Siempre, **siempre** usar nombres de variables representativas. Prohibido usar `mn = "Pepe"` en vez de `mi_nombre = "Pepe"`.

- Lo ideal es que todos los `from ... import` estén al principio.
- Tratar de no usar acentos en las variables.
- Usar guiones bajos o CamelCase en vez de espacios: `hola_mundo` o `holaMundo`.
 - Si se usa una forma, tomarla como convención y no cambiarla en la mitad del código.
- Siempre, **siempre** usar nombres de variables representativas. Prohibido usar `mn = "Pepe"` en vez de `mi_nombre = "Pepe"`.

Outline

- 1 ¡Empecemos!
- 2 Salida y variables
- 3 ¿Preguntas?
- 4 Licencia

- 3 ¿Preguntas?
 - ¿Preguntas?

¿Preguntas?

¡No se queden con dudas!

¿Preguntas?

- 1 ¡Empecemos!
- 2 Salida y variables
- 3 ¿Preguntas?
- 4 Licencia**

- 4 Licencia
 - Licencia de Esta Obra

Excepto en los lugares que se ha indicado lo contrario:

Esta obra se distribuye bajo una Licencia Creative Commons Atribución-SinDerivadas 4.0 Internacional.



CC-By-ND

Excepto en los lugares que se ha indicado lo contrario:

Esta obra está licenciada bajo la Licencia Creative Commons Atribución-SinDerivadas 4.0 Internacional. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nd/4.0/>.