

PROGRAMACIÓN PARA CIENCIAS EXACTAS  
Talleres FIFA (Federación Interestudiantil de Físicos de Argentina)



### Trámites

Luego, para comenzar, en caso de que el IDE de las PC no funcione, se puede probar programar online en la web: <https://www.python.org/shell/>

Ahora sí.

### Ejercicios propuestos

#### Unidad 1. Comandos y variables

Las variables son formas de almacenar información. La diversidad de esta información da origen a la diversidad de tipos de variables. Los comandos son operaciones que se pueden hacer sobre las variables para transformarlas, leerlas u obtener otras cosas de ellas. Recomendamos utilizar el comando *type(variable)* al final de cada ítem para que Python nos devuelva el tipo de variable con la que estamos trabajando.

##### ■ Comandos *print* e *input*

Haga que Python realice las siguientes tareas:

1. Escriba el texto 'Hola mundo'.
2. Deje un espacio de tabulación o un salto de línea entre *hola* y *mundo* (pruebe con `\t` y `\n`).
3. Guarde en una variable el texto 'Me tiemblan los dedos de la emoción de saber que estoy programando'.
4. Guarde en dos variables dos textos, y que imprima ambas variables separadas por un espacio de tabulación. (¿Se puede usar el `+` para eso?)
5. Pruebe escribir `a=int(input('Dale un valor a a: '))` ¿Qué tipo de variable es *a*? ¿Por qué sería necesario el comando *int*? ¿Cambia la función del `+` con el tipo de variable que vincula?
6. Aplique el comando *len(variable)* sobre una de las variables con texto y vea qué devuelve.

##### ■ Variables *integer* y *float*

¿Qué diferencia hay entre ambos tipos de variables?

1. Guarde en una variable *a* el valor 7, en otra variable *b* el valor 9, y que los sume.

2. Pruebe ahora restarlas, multiplicarlas y dividir las. ¿Qué otras operaciones matemáticas se pueden hacer con Python?
3. Guarde en una variable el resultado de la división de dos variables *integer*. ¿Qué tipo de variable es este resultado?
4. Averigüe: ¿Cuántas cifras puede almacenar una variable tipo *float*? <sup>1</sup>
5. ¿Qué tipo de variable devuelve Python al aplicar *round(x,n)*? (siendo *x* un *float* y *n* la cantidad de decimales que uno desea redondear).

■ Variables *boolean*

1. ¿Qué valor obtendrá de  $5 > 5$ ? ¿Y de  $5 >= 5$ ?
2. ¿Qué obtendrá de escribir  $4 == 5 == 3! = 2$ ? ¿Son útiles los paréntesis?
3. ¿Valen los operadores  $==$  y  $!=$  para *strings*?

■ Variables tipo *list* y tipo *tuple*

¿Qué diferencia hay entre ambos tipos de variables?

1. Arme una *lista* con las materias en las que debe el examen final y asígnela a una variable.
2. Pasó el cuatrimestre y nos colgamos todos. Agregá un elemento a la lista anterior (*append*) y pedile que te muestre el segundo elemento.
3. ¿Cuál es la diferencia entre *extend* y *append*? ¿Y con *insert*?
4. ¿Qué diferencia hay entre los comandos *remove* y *pop*? ¿Qué hace el comando *sort*?
5. Armada una lista, aplíquele la función *len*.
6. Pruebe *range(10)*, *range(3,13)*, *range(2,26,2)*. ¿Qué tipo de variable es el resultado?
7. Reconozca algunos de estos métodos en la descripción de las listas usando *help(list)* (para ver más en el *help* presionar *enter* y para salir *q*).

## Unidad 2. Bibliotecas (o *libraries* en inglés)

Las bibliotecas contienen funciones pensadas y distribuidas para extender las posibilidades de Python. Lo primero que debemos hacer es cargar la biblioteca con el comando *import*. Para resolver este ejercicio usaremos la biblioteca *math*.

Probar luego de eso resolver los siguientes ejercicios.

1. Calcular el seno de  $\pi$  en radianes. Si no te gusta, cambialo a grados (funciones *degrees* o *radians*)
2. Calcular *arctan*(1/2).
3. Calcular  $2^{3^4}$  y  $e^\pi$
4. Calcular  $\log_3(25)$ .
5. Calcular  $e^{\ln(x)}$  siendo *x* el número que quiera.

**Pregunta aislada:** ¿Qué pasa si se aplica la función *type* sobre la respuesta a *type* sobre una variable?

---

<sup>1</sup>Pista: <https://docs.python.org/3/tutorial/floatingpoint.html>

### Unidad 3. Condicionales y funciones

Una función es un paquete aislado e independiente de acciones a realizar. Pueden ser definidas y nunca utilizadas. Definir funciones como serie de acciones a realizar con una o más variables nos da la posibilidad de aplicar ese paquete más de una vez sin necesidad de repetir código.

Implemente las siguientes funciones (*def* o *lambda*) en Python:

1. **doble**(n): muestre el doble de n.
2. **promedio3**(n1, n2, n3): calcule y muestre el promedio de tres números pero redondeando para arriba.
3. **iguales**(n1,n2): devuelve True si  $n1=n2$ .
4. **divisible**(n,d): devuelve True si n es divisible por d.
5. **factorial**(n): devuelve el valor factorial de n.
6. **primo**(n): devuelve True si n es un número primo.
7. **norma**(x,y,z): devuelve la distancia de un punto al cero de un espacio euclidiano.

### Unidad 4. Gráficos

Para encaminarnos al próximo encuentro, resulta básico y muy útil saber graficar funciones de una variable respecto a otra. Para esto hay funciones ya armadas en las bibliotecas *Numpy* y *Matplotlib*.

Para experimentos numéricos, procederemos siempre de la misma manera:

1. Crearemos un dominio de valores equiespaciados con *numpy.linspace()*.
2. Crearemos una imagen asociada a una función  $f(x)$  definida por nosotros.
3. Aportaremos ruido a los datos, simulando "mediciones", con funciones como *numpy.random.rand()*
4. Graficaremos  $x$  vs  $y$  mediante las funciones *scatter* o *plot*, creando una grilla, dándole límites al dominio y la imagen mostrados y dándole nombres tanto a los ejes como al gráfico.

Interesante sería ver cómo hacer varios gráficos juntos (*subplot*) o cómo hacer gráficos de tipo histograma, cosas que abarcaremos en el próximo encuentro.

### Unidad 5. Juegos en Python

Esta sección tiene un par de desafíos de aplicación de lo visto hasta ahora. Son totalmente resolubles a esta altura del aprendizaje.

Intente crear los siguientes juegos en Python:

1. ¡Adivina un número del uno al diez! (que el usuario pruebe números del uno al diez y el programa le diga si adivinó o no)
2. Piedra, papel o tijera (Variante para valientes: Piedra, papel, tijera, lagarto y spock).
3. Ta-te-tí.

