**Funciones principales en Turtle Graphics**

<https://www.programoergosum.com/cursos-online/raspberry-pi/245-turtle-graphics-en-python-con-raspberry-pi/mi-primera-figura>

Las funciones principales para animar nuestro objeto son las siguientes:

**forward(distance)**: Avanzar una determinada cantidad de píxeles.  
**backward(distance)**: Retroceder una determinada cantidad de píxeles.  
**left(angle)**: Girar hacia la izquierda un determinado ángulo.  
**right(angle)**: Girar hacia la derecha un determinado ángulo.

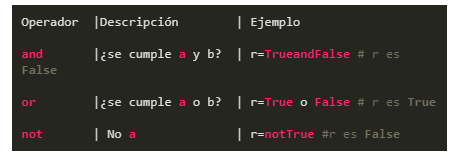
Por otro lado, puede que en ocasiones queramos desplazarnos de un punto a otro sin dejar rastro. Para ello utilizaremos las siguientes funciones:

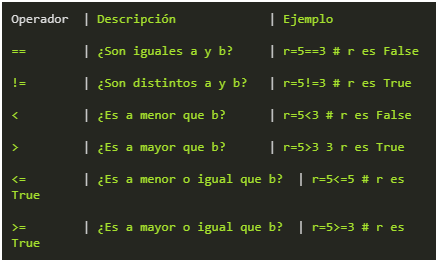
**home(distance)**: Desplazarse al origen de coordenadas.  
**goto((x, y))**: Desplazarse a una coordenada en concreto.  
**pendown()**: Subir el lápiz para no mostrar el rastro.  
**penup()**: Bajar el lápiz para mostrar el rastro.

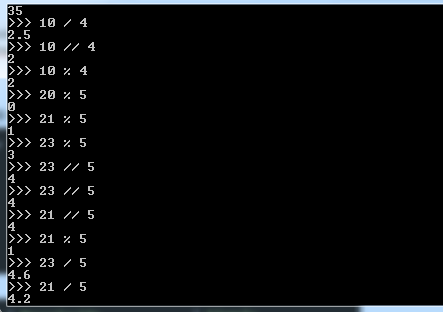
Por último, puede que queramos cambiar el color o tamaño del lápiz. En ese caso utilizaremos las siguientes funciones gráficas:

**shape(‘turtle’)**: Cambia al objeto tortuga.  
**pencolor(color)**: Cambiar al color especificado.  
**pensize(dimension)**: Tamaño de la punta del lápiz.

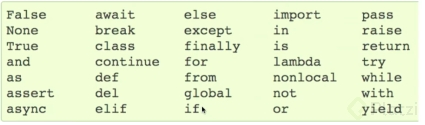
* **OPERADOS MATEMATICOS**







* **PALABRAS RESERVADAS**



* **FUNCIONES IMPORTANTES**

<https://docs.python.org/3/library/functions.html#any>

**Operaciones con Strings en Python**

Los strings tienen varios métodos que nosotros podemos utilizar.

* **upper**: convierte todo el **string** a mayúsculas
* **lower:** convierte todo el **string** a minúsculas
* **find:** encuentra el indice en donde existe un patrón que nosotros definimos
* **startswith:** significa que empieza con algún patrón.
* **endswith:** significa que termina con algún patrón
* **capitalize:** coloca la primera letra en mayúscula y el resto en minúscula
* **in y not in** nos permite saber con cualquier secuencia sin una

subsecuencia o substrings se encuentra adentro de la secuencia mayor.

* **dir:** Nos dice todos los métodos que podemos utilizar dentro de un objeto.
* **help:** nos imprime en pantalla el *docstrings* o comentario de ayuda o instrucciones que posee la función. Casi todas las funciones en Python las tienen.

Ej:



Para limpiar en el cmd de windows usa:

>>> import os

>>> clear = lambda: os.system('cls')

>>> clear()

y despues ya puedes usar solo

>>> clear()

* **METODOS STRING**

|  |  |
| --- | --- |
| * help(str) dir(str) help(str.splitlines) import time help(time)   capitalize() casefold() center(width[, fillchar]) count(sub[, start[, end]]) encode(encoding=“utf-8”, errors=“strict”) endswith(suffix[, start[, end]]) expandtabs(tabsize=8) find(sub[, start[, end]]) format(\*args, \*\*kwargs) format\_map(mapping) index(sub[, start[, end]]) isalnum() isalpha() isdecimal() isdigit() isidentifier() islower() translate(table) upper() zfill(width) | isnumeric() isprintable() isspace() istitle() isupper() join(iterable) ljust(width[, fillchar]) lower() lstrip([chars]) maketrans(x[, y[, z]]) partition(sep) replace(old, new[, count]) rfind(sub[, start[, end]]) rindex(sub[, start[, end]]) rjust(width[, fillchar]) rpartition(sep) rsplit(sep=None, maxsplit=-1) rstrip([chars]) split(sep=None, maxsplit=-1) splitlines([keepends]) startswith(prefix[, start[, end]]) strip([chars]) swapcase() title() |

**CRUD**

(C)reate

(R)ead or retreive

(U)pdate

(D)elete

**Operaciones con strings: Slices en python**

Los *slices* en Python nos permiten manejar secuencia de una manera poderosa.

**Slices** en español significa ““rebanada””, si tenemos una secuencia de elementos y queremos una rebanada tenemos una sintaxis para definir que pedazos queremos de esa secuencia.

secuencia[comienzo:final:pasos]

EJ:

