

El mágico mundo de Python







La magia de programar:

Todos los días, desde que te levantas, tú expresas lo que piensas y sientes a través del lenguaje.

Como cuando tienes hambre y quieres que tu mamá te prepare algo para comer, tienes que usar tu voz y decirle: ¡Mamá tengo hambre! o colocar cara de tristeza, mientras te tocas la barriguita con una mano y señalas la boca con la otra, para que así ella sepa que tienes el estómago vacío, allí ya estás empleando una forma de comunicación.

Nosotros los humanos tenemos que comunicarnos todo el tiempo con todos aquellos que nos rodean, cosa que intentamos hacer a través de todas las formas posibles.

Al nacer aprendemos una lengua o idioma, a ésta le llamamos lengua materna. Hay muchos lenguajes, como la música que escuchas, los gestos, las imágenes, los avisos de tránsito.

Incluso los colores son una forma de lenguaje, como pasa con el semáforo, que avisa con el color verde si puedes cruzar la calle o con el color rojo si debes detenerte.



4 El Mágico mundo de Python



También tenemos el lenguaje articulado, que son todas esas palabras que tú dices diariamente. Aquí en Venezuela hablamos de muchas maneras: andino, llanero, zuliano, caraqueño, guaro, guayanés, oriental, pemon, wayú, entre otros.

Como la tecnología ha avanzado tanto, las computadoras también pueden comunicarse entre ellas, pero es un lenguaje que no entendemos nosotros los humanos, porque se habla a través de ceros (0) y unos (1), a esto le llaman Lenguaje de máquina, pues sólo lo entienden ellas.

Pero luego se inventaron varias formas de lenguaje entre las computadoras y los humanos, que nos permiten hablar fácilmente con nuestro computador, por medio de códigos escritos que transforman al código de máquina, todo eso que decimos con nuestras palabras, para que así el procesador (el cerebro) de la computadora pueda recibir la información correctamente. Estos son los “*Lenguajes de programación*” que fueron creados para describir, paso por paso, todas las tareas que debe hacer una computadora, lo que nos permitirá realizar actividades informáticas de una forma muy fácil.

Un lenguaje de programación tiene varias ventajas porque es mucho más fácil de entender que un lenguaje máquina y nos permite mayor portabilidad; esto quiere decir, que se adapta fácilmente en diferentes tipos de computadoras.

En esta ocasión jugaremos con el lenguaje de programación Python, que es muy flexible y fácil de usar, con él puedes crear figuras geométricas, dibujos y hasta juegos.

Leer y escribir en Python es realmente sencillo sólo es cuestión de que sigas uno por uno los pasos que se te indiquen para que conozcas la mágica historia de la serpiente encantada Python.



ANUK'PURA: EL HOGAR DE PYTHON Y DE SU MISTERIOSA AMIGA

¡Ven y te cuento el cuento de la gran serpiente mágica!

En un lugar de Canaima de cuyo nombre quisiera contarte, existió la leyenda de una serpiente gigantesca llamada Python que rodeaba a todo el pueblo de Anuk'pura y lo cuidaba desde la profundidad de la tierra.

Hace mucho tiempo atrás, ocurrían cosas maravillosas e insólitas en este boscoso pueblo.

Los niños lo visitaban constantemente porque les encantaba ver cuando el cielo y las estrellas cambiaban de color sin una razón aparente.

Acariciaban las tortugas moradas y azules, paseaban por los misteriosos laberintos -muy comunes por esos lados-, tomaban siestas bajo los Árboles Arcoiris, llamados así porque absorbían toda la luz del sol para cambiar los colores de sus ramas.



6 El Mágico mundo de Python



El imponente tamaño de la serpiente Python les impresionaba, sin contar su poder para comunicarse con todo ser viviente de la zona, lo cual era casi imposible de creer por cualquier persona cuerda.



Aunque, lo más misterioso de esta historia era lo que dormía cerca de la boca de Python, allí decían que descansaba una tortuga mágica llamada Turtle, quien también podía modificar su tamaño y su color.

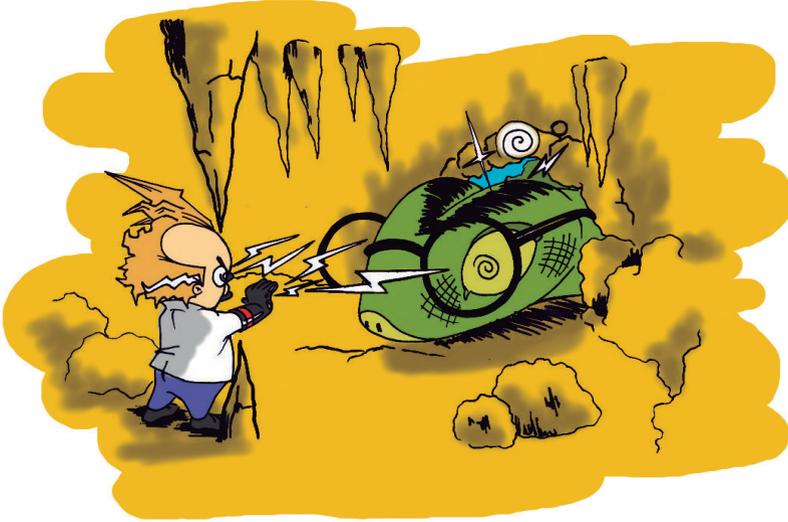
Según la gente, Turtle era la causante de esos hallazgos increíbles, pues Python la despertaba todos los días para que hiciera su tarea diaria: ¡Construir un pueblo mágico donde todos pudiesen convivir!

Labor que no sólo mejoraba las cosechas y la fauna, sino que además llenaba de color y alegría a todos los habitantes de Anuk'pura.



El Mágico mundo de Python 7

Un día, a causa de un embrujo del virulento Mago Desinstalador, Python quedó atrapada en una cueva de oro subterránea que le robó toda su memoria, ahora no recuerda absolutamente nada, ni siquiera a su amiga Turtle, quien sumergida en un sueño profundo, no ha logrado despertarse más.



Desde entonces todo parece gris y sin vida, la alegría se ha marchitado en el pueblo de Anuk'pura.

Muchos encantadores de serpientes intentaron llegar hasta donde Python para regresarle la memoria y ordenarle que despierte a Turtle, pero no se concentraron y se perdieron en el camino.

Nanike' edai (tengo miedo), ¡Estamos ante una emergencia! sólo queda alguien capaz de regresarle la alegría a Anuk'pura.

¡Ese eres tú!



8 El Mágico mundo de Python



Esta historia ha caído en tus manos, Apetöy te ha escogido como el nuevo encantador de serpientes. Eres el responsable de traer la paz de vuelta a su pueblo. ¡Bienvenido al mágico mundo de Python!

Como el nuevo encantador de serpientes, tienes que superar algunos obstáculos que se te presentarán en el camino, para ello debes ir con cuidado y conocer muy bien cada nivel, así la magia de Python se apoderará de ti y podrás ser el mejor encantador de todos los tiempos, son tres niveles que debes superar a lo largo de esta historia.





Nivel I: Háblame de Python
Nivel II: Buscando a Python y a Turtle
Nivel III: Reconstruyendo el pueblo con
Turtle



Nivel I: HÁBLAME DE PYTHON

“El teclado, nuestra varita mágica...”

Python está dentro de tu computador. Para empezar, puedes acceder a nuestro mundo a través del teclado. Existe una serie de encantamientos o atajos, para hacerte fácil controlar los programas; para ello, debes conocer muy bien el poder de tu teclado, como por ejemplo:



TECLA	DESCRIPCIÓN
F1	Nos permite obtener Ayuda del sistema o programa.
ALT+TAB	Nos permite cambiar entre los diferentes programas que se encuentren abiertos.
ALT+F4	Nos permite salir del programa.
MAYÚS+SUPR	Sirve para eliminar un elemento permanentemente.
CTRL+C	Sirve para copiar palabras, párrafos y hasta un documento completo.
CTRL+X	Sirve para cortar palabras, párrafos y hasta un documento completo.
CTRL+V	Sirve para pegar, palabras párrafos y hasta un documento completo.
CTRL+B	Sirve para colocar en negrita las palabras
CTRL+U	Sirve para subrayar las palabras
CTRL+I	Sirve para colocar en cursiva las palabras
Para copiar un archivo de una carpeta a otra sólo debemos mantener presionada la tecla CTRL mientras arrastramos el archivo al lugar donde la queremos copiar.	





Pero... **¿Quién es Python y cómo funciona?**

Python, la gran serpiente, es un ser mitológico que funciona a través de órdenes, es considerado un lenguaje de programación creado para dar instrucciones a las máquinas, como por ejemplo tu computadora.

El pueblo decidió llamarle Python en honor a Monty Python's Flying Circus, los primeros encantadores que embrujaron la pantalla de televisión con una serie muy graciosa para niños grandes.

¡Con Python puedes crear casi lo que quieras! Pero no es fácil hacerlo solito, necesitarás del arduo esfuerzo de la tortuga (Turtle), la fiel aliada que ha acompañado a Python desde que salió de su cascarón, ella recorrerá todo el pueblo y seguirá las órdenes que le solicites para reconstruirlo.

Por ello, debes despertarla de su sueño encantado.

A través de Turtle, podrás manejar los comandos u órdenes para crear comportamientos dentro del programa.

¡Lograrás hacer casi lo que quieras en el pueblo a través de comandos!

Un comando es una orden que tú como usuario, le das a una computadora.





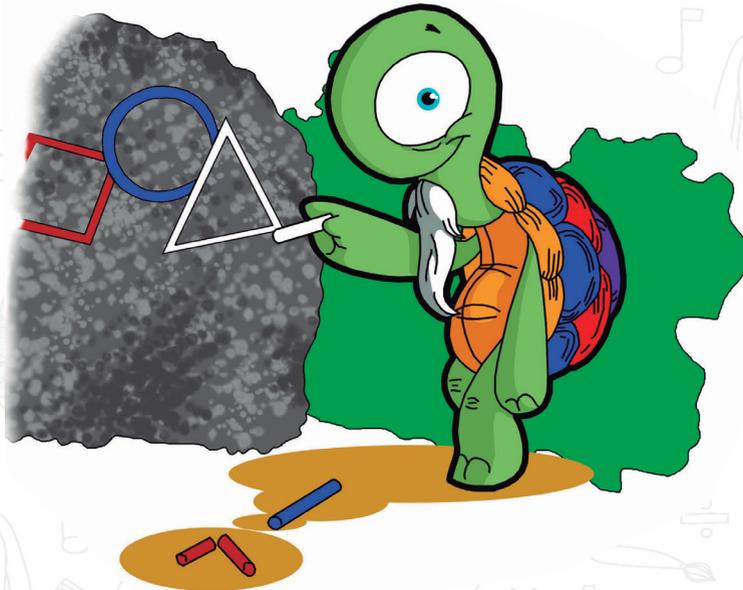
Sirven para indicarle al programa que debe ejecutar una acción y así obtendremos una respuesta de él.

Por ejemplo, puedes cambiar el color de las estrellas o elegir qué forma tendrán las nubes en un día soleado.

Claro, para que las cosas funcionen, siempre es necesario dictar las órdenes en secuencia. Python necesita que aclares cómo quieres hacer las cosas, para poder ejecutarlas.

La manera más fácil es utilizando algoritmos o pasos ordenados, como cuando vamos a preparar galletas y debemos seguir la receta al pie de la letra para que no se nos queme o nos queden crudas; cada paso de la receta es un comando, que realizándolo uno a uno, nos permitirá obtener unas ricas galletas. Ese conjunto de comandos bien definidos y ordenados que permiten realizar las galletas mediante pasos sucesivos (uno después de otro) es lo que llamamos **algoritmo**.

¿Ves? ¡Es súper fácil!





Nivel II: Buscando a Python

La herramienta principal que tendrás a tu disposición en esta aventura, es tu Canaimita o computador. Allí encontrarás a Python.

No siempre es tan fácil, verás, el último encantador hizo que Python se perdiese por completo, confundiéndola y alejándola de Canaima por mucho tiempo.

Para ésto sólo debes abrir la ventana de **Terminal** Primero dirígete al panel (la barra superior en el escritorio)

Menú aplicaciones (logotipo de canaima): Es el menú que sirve para acceder a las principales aplicaciones del sistema, agrupadas por categorías

Haz clic en el logotipo de canaima, verás un menú.

¡Bien! debes seleccionar accesorios y luego **Terminal**. ¡No lo olvides! El camino es corto: **Aplicaciones -> Accesorio -> Terminal**.

Lo que debes estar viendo es una ventanilla negra -como el universo- con el cursor parpadeante, indicando que está esperando por alguna orden.

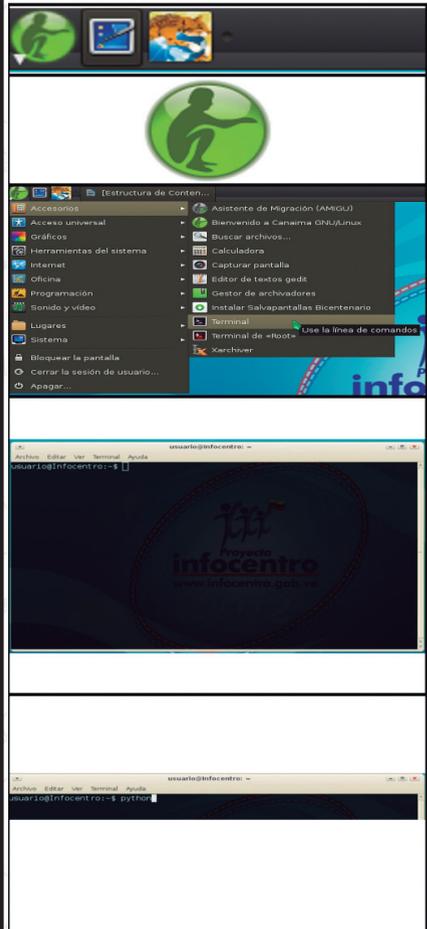
Debe tener escrito el nombre de usuario de la computadora y al final los símbolos. Algo como lo siguiente: `:~$` (dos puntos seguido de un guión ondulado y dolar) o un símbolo # (numeral).

Es decir, podemos encontrarnos con algo como: **usuarios@usuarios:~\$** o **usuarios@usuarios:#**

Ahora llamemos a **python** a ver si nos responde.

Escribe la palabra **python** en el terminal (importante escribir todo en minúsculas) y luego presiona la tecla **Enter**.

Enter

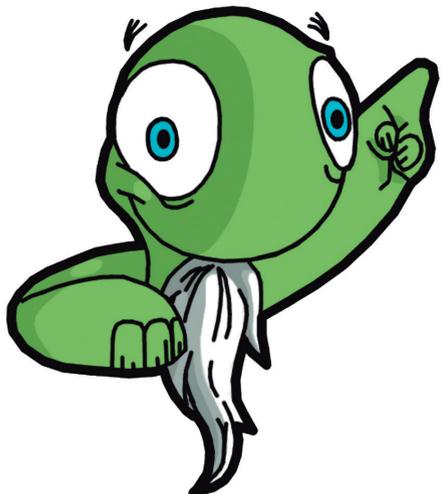




<p>Si python nos ha respondido, debemos ver un mensaje como el siguiente: [GCC4.4.5] on linux2 Type "help", "copyright", "credits" or "licese" for more information >>></p>	<pre> usuario@infocentro: ~/Escritorio Archivo Editar Ver Terminal Ayuda usuario@infocentro:~\$ cd Escritorio usuario@infocentro:~/Escritorio\$ python Python 2.6.6 (r266:84292, Dec 27 2010, 00:02:48) [GCC 4.4.5] on linux2 Type "help", "copyright", "credits" or "license" for more informati >>> </pre>
<p>Eso pequeños Símbolos >>> son el comienzo de una nueva línea de código "El encantamiento" y son que anuncian que se dará una nueva orden. (Para salir del prompt de python presionamos las teclas ctrl + Z)</p>	
<p>Si te aparecen el mensaje "no se encontró la orden" ¡Oh, oh! python no nos ha escuchado, esto quiere decir que nos equivocamos de cueva. Hay que hallar la cueva correcta donde ésta atrapado python. Y para eso necesitamos un nuevo poder. El poder de Instalador.</p>	<pre> usuario@infocentro:~\$ python bash: python: no se encontró la orden usuario@infocentro:~\$ </pre>
<p>¿cómo tener el poder de instalador?</p>	<p>Ve a la pagina instalación de python</p>

Muy Bien encantadores, recordemos que Python está atrapada en una cueva y tenemos que sacarla de ahí.

¿Quieres saber dónde se encuentra? Entonces sigue los pasos que te llevarán hasta nuestra amiga Python:





Primero dirígete al panel (La barra superior en el escritorio)



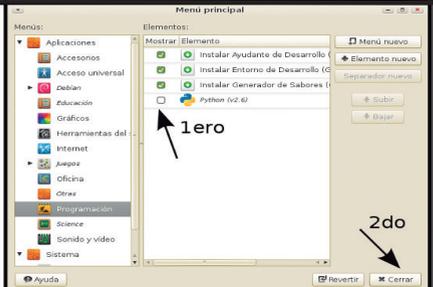
¡Ya estás más cerca!
Sólo debes hacer **clic derecho** con el botón del ratón en **Aplicaciones** y se abrirá una ventana con un menú.
Entonces selecciona **Editar menú**.



¿Listo?
Deberías estar viendo una ventana de **Menú principal**.
Luego hacemos clic en el menú **programación** que se encuentra en el panel izquierdo.



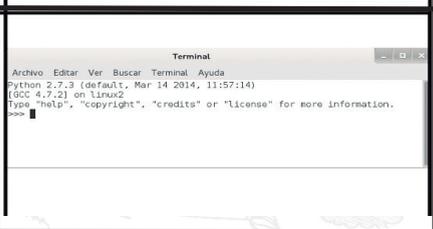
Podrás observar que se despliega un listado de sub-menús en el panel derecho de la ventana (Revisa detalladamente, está allí del lado derecho).
Allí, mis queridos y queridas encantadoras y encantadores vemos a Python.
Selecciona **Python** y haz clic en el botón **Cerrar**.



¡Muy bien! has sacado a python de su cueva.
¡Y al mismo tiempo hemos agregado a python al menú!
Ahora volvamos al menú haciendo **clic izquierdo**
¿Recuerdas el Camino?
Aplicaciones - programación - python.
¡Estamos listo para la aventura!



¡Ohhh! Vemos que se abrió otra ventana con un mensaje como el siguiente:



[GCC4.4.] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>



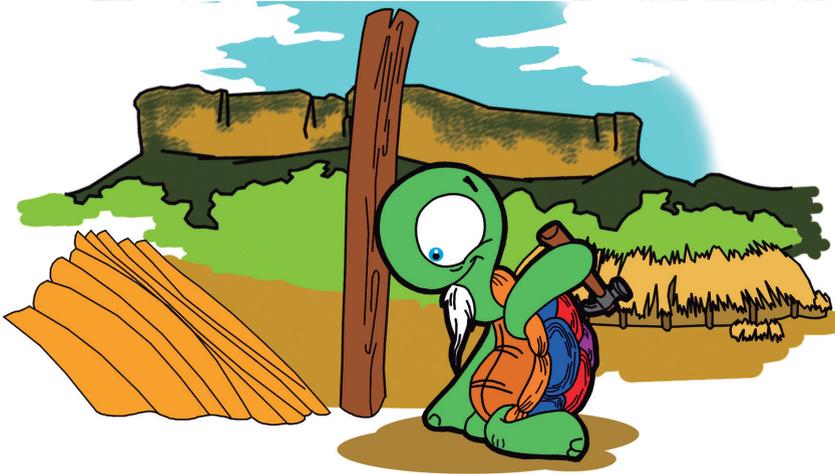
¡Ya has conseguido y rescatado a nuestra amiga la serpiente, ahora debes ordenarle que despierte a la sabia tortuga para ponerle fin a ese horrible embrujo del virulento Mago Desinstalador!

Ahora solo hay que hablar en el lenguaje de la serpiente Python, para que nos lleve a donde queremos llegar.

¡La aventura apenas inicia!

Conociendo a la Tortuga

La Tortuga es el personaje más antiguo del pueblo. Cuenta la leyenda que caminó por años desde montañas desconocidas y levantó poco a poco los cimientos de lo que se convertiría después en el hogar de la serpiente mágica.



Allí la acompañó y la ayudó con las labores del pueblo, pero luego el malvado hechizo hizo que se quedara profundamente dormida, esperando a que alguien digno se convirtiera en el encantador de serpientes y salvador de Anuk'pura.



La tortuga, tiene todos los conocimientos del pueblo y puede multiplicarse cuantas veces lo desee. Es como una librería. Pero te preguntará **¿a qué me refiero cuando digo que es como una librería?**, pues fíjate, se le llama “librería o módulo” a la agrupación de órdenes que están disponibles para ser usadas en la construcción de un programa. Con la librería de turtle podemos crear tortugas y darle órdenes para que dibujen formas y patrones, esas formas y patrones están bien definidas dentro del módulo, por ejemplo, para dibujar un círculo, el módulo define que debe tener un tamaño y eres tú quien le dirá de qué tamaño será.

Ahora debemos encontrar a Turtle para luego despertarla y comenzar la reconstrucción de Anuk’pura, ¡vamos a buscarla!

Para saber si Turtle (**La Tortuga**) se encuentra cerca de python, vamos a abrir python.

Vamos al menú Aplicaciones haciendo clic **izquierdo** con el ratón.

¿Recuerdas el camino?

Aplicaciones - programación - python



```
usuario@infocentro: ~  
Archivo Editar Ver Terminal Ayuda  
usuario@infocentro:~$ python  
Python 2.6.4 (r2648402, Dec 27 2010, 09:02:40)  
[GCC 4.4.5] on linux2  
Type "help()", "copyright()", "credits()" or "license()" for more information.  
>>>
```

Python tiene el poder para comunicarse con todo ser viviente de la zona, así que cuando Python nos esté escuchando le decimos que despierte a Turtle ¿cómo?, pues hablaremos en el lenguaje de Python tecleando dos palabras mágicas.

import turtle y luego presiona **Enter**.

Si lo que ves ahora son los símbolos **>>>** es porque Python se está comunicando con Turtle.



Si por alguna causa aparece la frase **please install the Pythontk package**, debemos usar los poderes de instalador para **instalar el paquete indicado, en este caso Python-tk**.

Gracias a tu ayuda liberamos a Python, le regresaremos su memoria y despertamos a Turtle para que mejorara las cosas en Anuk'pura, ahora debes reconstruir el pueblo, crear estrellas, caminos e ingeniártelas para construir el laberinto más grande que haya tenido Anuk'pura; esto no se ha hecho antes, pero tú lo harás posible para que el pueblo sea renombrado a lo largo de la historia. Puedes hacer el laberinto del color que desees.

¡Es hora de comenzar!

Lo primero que debemos hacer es crear un lienzo en el cual Turtle nos ayudará a dibujar el pueblo de Anuk'pura, (un lienzo es una base de tela que los artistas usan para pintar).

Usa tus poderes para crear en lienzo.

Escribe **turtle.setup(800,600)** y presionamos la tecla **Enter**

Esta orden hará que aparezca un lienzo en el que podamos interactuar con la tortuga.

Los números indican el tamaño que tendrá el lienzo y podemos cambiarlos como deseemos.

```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
Python 2.7.3 (default, Mar 14 2014, 11:57:14)
[GCC 4.7.2] on linux2
Type "help", "copyright", "credits" or "licen
>>>
>>> import turtle
>>> turtle.setup(800,600)
>>> |
```

Python Turtle Graphics



Nivel III:

Reconstruyendo el pueblo con Turtle



Existen muchos comandos u órdenes que nos permiten jugar con un millón de posibilidades en nuestro lienzo de Anuk'pura. Recuerda que ahora eres el encantador de esta historia, de tu creatividad depende que Python restaure el orden en el pueblo. Lo que necesitaremos será ingresar a Python, posteriormente buscaremos a la tortuga (como practicamos antes) y una vez que tengamos el lienzo abierto, podemos darle rienda suelta a la imaginación.

¡Despertemos a Turtle!

Como la Tortuga es un ser milenario, duerme muchas horas para poder concentrarse y manejar la cantidad de conocimientos que posee.

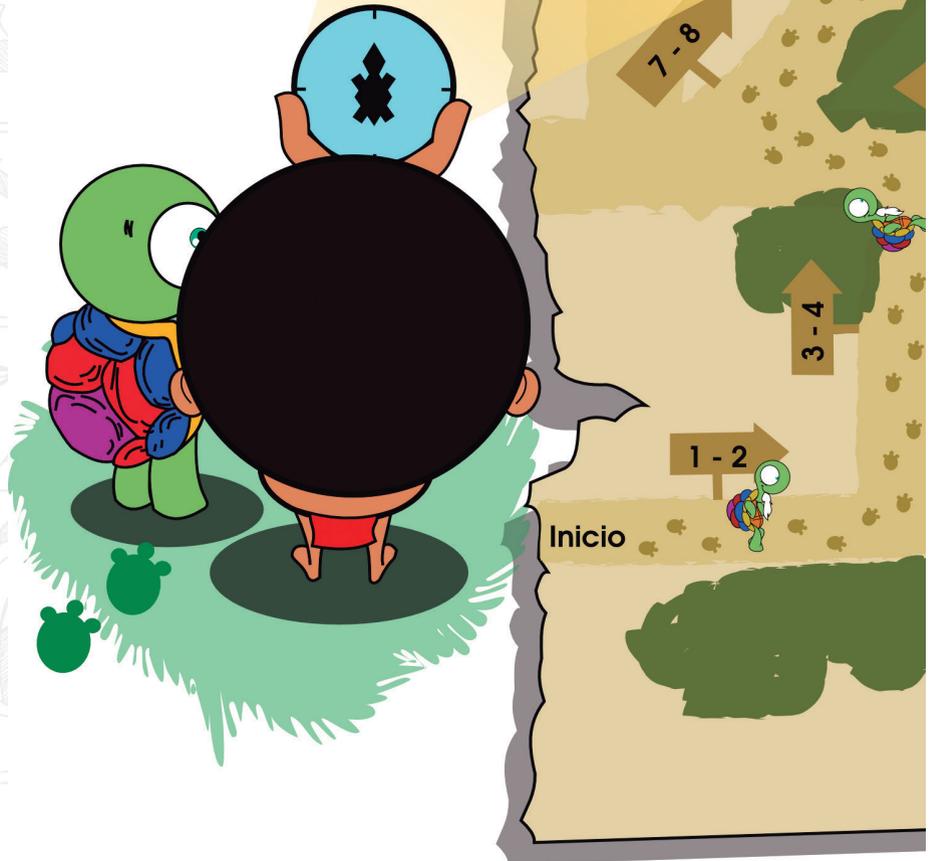
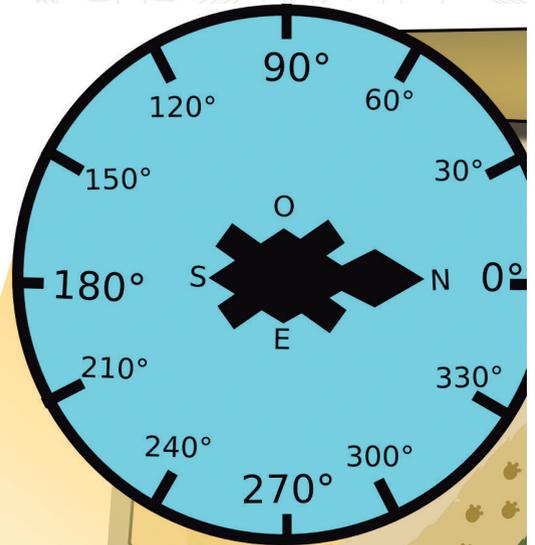
El malvado hechizo del Mago Desinstalador hizo que Turtle no pueda escuchar a Python y por ello no puede despertarse.

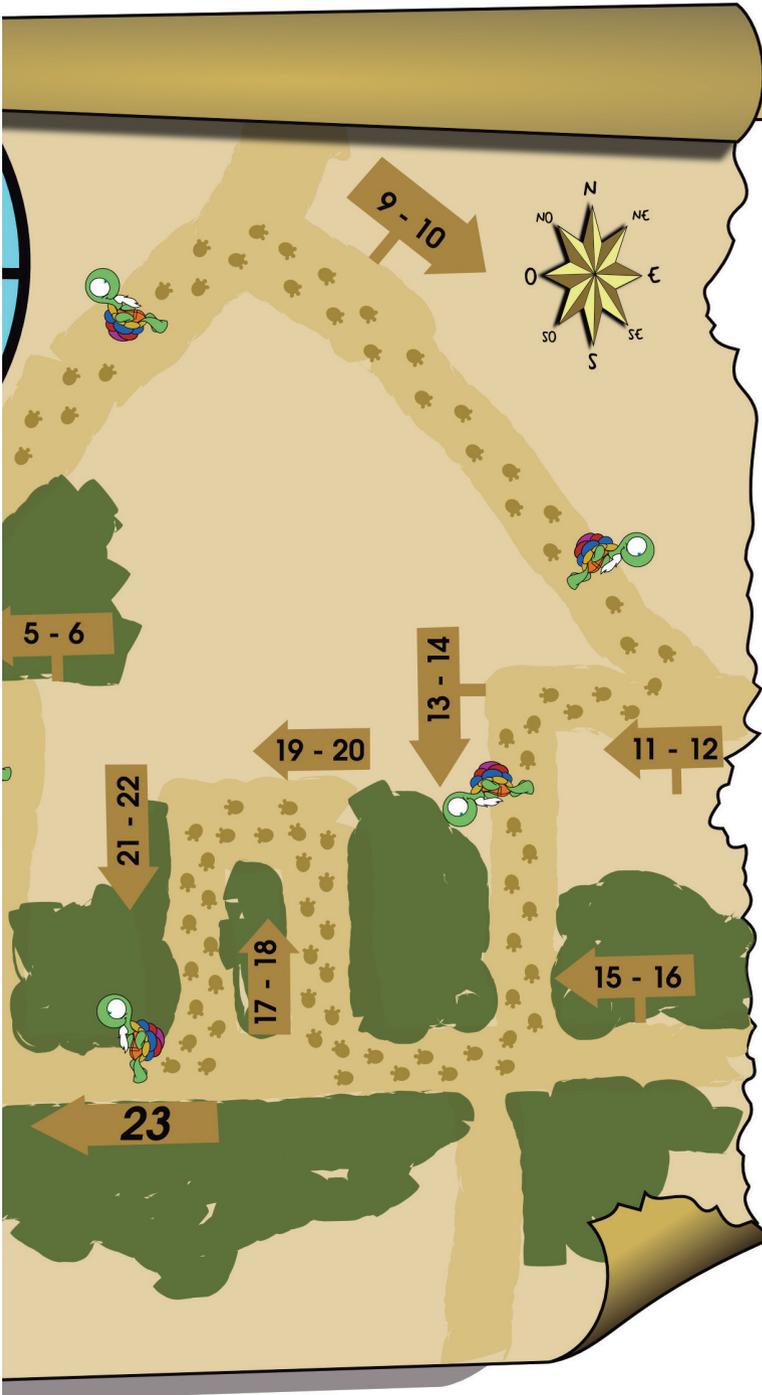
Tú encantador o encantadora eres la persona encargada de despertar a Turtle.





¡Sigue la ruta para construir una de las casas de Anuk'pura!





1. forward (100)
2. left (90)
3. Forward (100)
4. left (90)
5. forward (25)
6. left (225) o right (135)
7. forward (150)
8. left (270) o right (90)
9. forward (150)
10. left (225) o right (135)
11. forward (25)
12. left (90) o right (270)
13. forward (100)
14. left (270) o right (90)
15. forward (54)
16. left(270) o right (90)
17. forward (50)
- 18.left (90) o right (270)
- 19.forward (53)
20. left (90) o right (270)
21. forward (50)
22. left (270) o right (90)
23. forward (54)



¡Usa tus poderes y tu varita mágica!

Para despertar a Turtle con la varita (nuestro teclado) usaremos unas palabras mágicas, pero antes para que nos pueda escuchar desde su más profundo sueño debemos cambiarle el nombre.

Por ejemplo si queremos llamarla José, las palabras mágicas serán:

```
jose = turtle.Turtle()
luego presionamos Enter
```

```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
Python 2.7.3 (default, Mar 14 2014, 11:57:14)
[GCC 4.7.2] on linux2
Type "help", "copyright", "credits" or "licen
>>>
>>> import turtle
>>> turtle.setup(800,600)
>>> jose=turtle.Turtle()
>>> █
```

¿Si aparece una flecha en el lienzo, es que la tortuga se ha despertado de su profundo sueño.

Para ver en qué parte del lienzo se encuentra dormido Turtle la haremos aparecer tecleando:

```
jose.shape("turtle")
luego presionamos Enter
```

¡Listo! Hemos despertado a Turtle. Ahora con la ayuda de turtle podrás darle vida y colocar a nuestro pueblo.

Tu puedes fortalecer a Turtle, empieza por cambiar su tamaño, utilizando el siguiente comando:

```
jose.turtlesize(2)
seguido de presionar Enter
```

Son tan geniales las Tortugas, que su tamaño puede variar entre (1) e infinito. Sin embargo, diez (10) ya es bastante grande.

¡Anímate a probar con el tamaño que desees.

```
Archivo Editar Ver Buscar Terminal Ayuda
Python 2.7.3 (default, Mar 14 2014, 11:57:14)
GCC 4.7.2] on linux2
ype "help", "copyright", "credits" or "licen
>>>
>>> import turtle
>>> turtle.setup(800,600)
>>> jose=turtle.Turtle()
>>> jose.shape("turtle")
>>> jose.turtlesize(2)
>>> █
```



¡Encantador tienes que darle vida al pueblo!

Parece ser mucho trabajo para ti solo ***¿verdad?*** pero tranquilo, no tienes que colocarte la ropa de constructor por ahora, ni preocuparte de la cantidad de estrellas y figuras que debes hacer, pues para eso tenemos a nuestra amiga Turtle, ella se encargará de hacer posible tus ocurrencias, tú eres el creador y ella será como tu pincel en nuestro lienzo de Anuk'pura, ***¡vamos a empezar!***

¿Sabías que Turtle entiende el idioma inglés?, pues sí, usando algunas palabras mágicas en inglés, hacemos que cambie de color y de tamaño como lo hicimos anteriormente, también podemos hacer que se mueva y que dibuje.

Ahora vamos a conocer algunas palabras mágicas que usarás como poder especial para que turtle se mueva por el lienzo y comience a reconstruir el pueblo de anukpura.

forward	En español significa adelante
backward	En español significa hacia atrás

¡Muy buen trabajo! Ahora hagamos que Turtle se mueva por nuestro lienzo, para ello usamos las palabras mágicas en inglés.

Debes escribir el nombre que le dimos a Turtle seguido de un punto y luego el tipo de movimiento, pero tenemos que decirle cuantos pasos debe dar y para eso usamos los paréntesis (), dentro de estos debe ir el número de pasos que dará nuestra tortuga.

Es decir, si tecleamos **`jose.forward(100)`** le estaremos indicando a la tortuga que debe moverse **cien** pasos hacia adelante.



<code>jose.forward()</code>	Con este comando indicarás que se mueva hacia adelante
<code>jose.backward()</code>	Con este comando indicarás que se mueva hacia atrás.

¡Haz que Turtle se mueva cambiando el número de pasos que va dentro de los paréntesis!

¿Has visto que salió una línea recta? Turtle puede dejar su rastro a medida que camina, ¿sabes por qué? porque posee un lápiz mágico llamado **Pen**, cuyo color es el mismo de Turtle, y usaremos ese poder de Turtle para reconstruir Anuk'pura.

***¿ Quisieras saber cómo cambiar el color de mi lápiz?
¿ Conoces los colores en inglés?***



Repasemos

yellow = amarillo	blue = azul	red = rojo	pink = rosado
orange = anaranjado	purple = morado	green = Verde	lightgreen = verde claro
grey = gris	brown = marrón	skyblue = celeste	coral = coral
violet = violeta	maroon = marrón rojizo	tan = bronceado	black = negro
darkblue =azul oscuro	yellowgreen =verde claro	wheat = trigo	

Puedes usar un comando para cambiar el color de pen, sin que Turtle cambie de color.

```
>>> jose.pencolor("purple")
>>> jose.forward(50)
>>> □
```

Para cambiar de color del lápiz usamos los paréntesis y escribimos el color en inglés entre comillas

```
jose.pencolor("purple")
```

¿También puedes hacer que la líneas de **pen** sea más gruesa solo hay que usar la palabra mágica **pensize()** y dentro del paréntesis colocar el tamaño de la línea.

```
>>> jose.pensize(8)
>>> jose.forward(50)
>>> █
```

Por ejemplo: **jose.pensize(8)**



Pero Turtle también se puede mover sin dejar rastro, esto lo haces diciéndole a Turtle que levante a **Pen** con el comando siguiente: **jose.penup()**

Para que vuelva a bajar el lápiz usas el comando: **jose.pendown()**

Para borrar todo lo que ha dibujado la tortuga usa el comando: **jose.clear()**

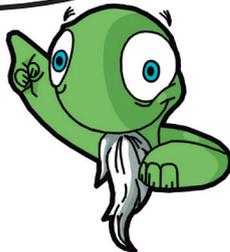
Bueno, ahora Turtle se encuentra muy pilas y con ganas de dar un paseo por el pueblo, para saber por dónde empezar a construir sus figuras geométricas; ella quiere girar su cuerpecito, para que pueda hacerlo debes ubicar su cabeza, que inicialmente apunta a la derecha.

Turtle gira del mismo lado hacia donde señala su cabeza.

Cabeza de la tortuga



¿Qué sabes de girar?



Girar es cuando das una vuelta completa, es decir, empiezas en un punto llamado punto de inicio, das la vuelta y finalizas quedando en el mismo punto donde empezaste.

¡Has dado una vuelta completa o lo que llamamos giro de 360 grados!

Pero también puede dar media vuelta y allí quedarás de espaldas al punto de inicio, entonces habrás girado 180 grados. Si giras a tu derecha o a tu izquierda, habrás dado un giro de 90 grados.



Para que Turtle gire hacia una dirección debemos indicarle a cuántos grados debe girar, estos grados pueden ser de 1 hasta 360, pero también Turtle necesita saber hacia cuál dirección, girará, esto se lo decimos en inglés.

right	En español significa derecha
left	En español significa izquierda

Para que Turtle gire usamos los siguientes comandos:

jose.right()	con este comando indicarás que se mueve hacia la derecha
jose.left()	con este comando indicarás que se mueva hacia la izquierda

No olvides encantador o encantadora que Turtle solo podrá girar si le indicas a cuántos grados debe hacerlo y eso se lo decimos colocándolo dentro del paréntesis.

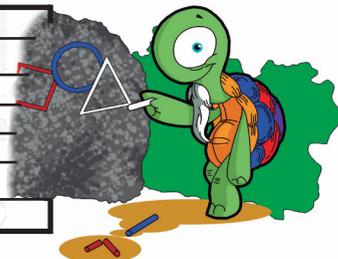
Haz que turtle gire a su izquierda o derecha, usando el comando siguiente: jose.left(90) o jose.right(90) .	

Haz que la tortuga gire indicándole otros grados, solo recuerda que debe ser un número entre 1 y 360.

Para que Turtle vuelva a al centro del lienzo decimos: **jose.home()**

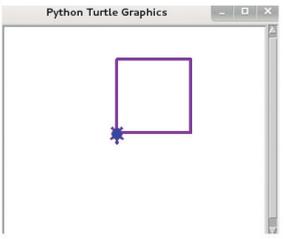
Para que Turtle borre todo y vuelva al principio decimos: **jose.reset()**

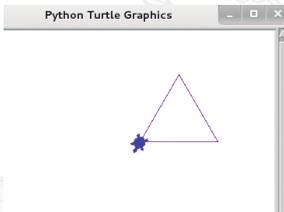
INGLÉS	SIGNIFICADO EN ESPAÑOL
reset	reiniciar
home	hogar
up	arriba
down	abajo
pen	lápiz
size	tamaño

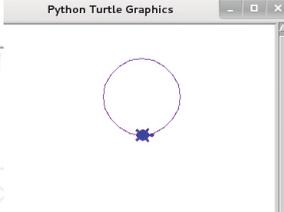




Cómo te has convertido en encantador, tu le darás las órdenes a Turtle para que dibuje el pueblo de Anuk'pura.

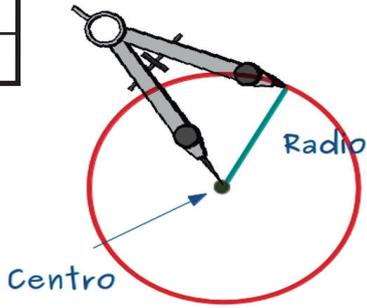
<p>Ordena que turtle dibuje un cuadrado de tamaño 100 pasos y de color morado. Usa los comandos aprendidos hasta ahora.</p> <p>¡¡Ayudadita!! Jose.left(90) jose.forward(100)</p>	<pre>>>> jose.pencolor("purple") >>> jose.pensize(4) >>> jose.forward(100) >>> jose.left(90) >>> jose.forward(100) >>> jose.left(90) >>> jose.forward(100) >>> jose.left(90) >>> jose.forward(100) >>></pre> 
--	---

<p>Ordena que turtle dibuje un triángulo de tamaño 100 pasos y de color morado. Usa los comandos aprendidos hasta ahora.</p> <p>¡¡Ayudadita!! jose.left(120) jose.forward(100)</p>	<pre>>>> jose.color("blue") >>> jose.pencolor("purple") >>> jose.forward(100) >>> jose.left(120) >>> jose.forward(100) >>> jose.left(120) >>> jose.forward(100) >>></pre> 
--	--

<p>Ordena que turtle dibuje un círculo de radio 50 y de color morado. Usa los comandos aprendidos hasta ahora.</p> <p>¡¡Ayudadita!! jose.circle(50)</p>	<pre>>>> jose.color("blue") >>> jose.pencolor("purple") >>> jose.circle(50) >>></pre> 
---	---

en inglés undo significa deshacer
en inglés circle significa círculo

El radio de un círculo es la medida desde el centro a cualquiera de sus extremos.





Puedes deshacer el último paso que realizó Turtle con el comando: **jose.undo()**

Las tortugas son seres de movimientos lentos, por lo que es un poco más complejo decirle a Turtle que se mueva, podemos cambiarle la velocidad para que pinte más rápido o más lento.

Usa el comando **speed()** para asignar la velocidad de la tortuga, toma un valor entre 1 y 11. El más rápido es el 11 y 1 el más lento. Ejemplo: **jose.speed(2)**

Rellenando figuras

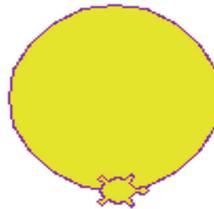
Es hora de que Turtle empiece a colorear figuras, para eso usarás 3 comandos **fillcolor("color")**, **begin_fill()**, y **end_fill()**

Con el comando **fillcolor("color")**, le indicas a Turtle el color que usará para colorear.

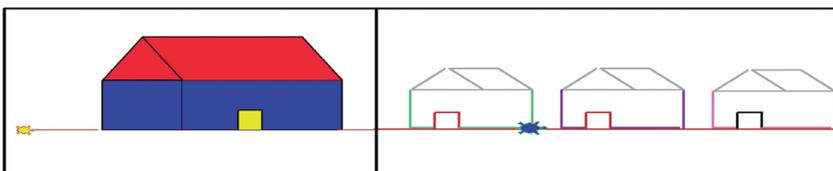
Usa **begin_fill()** para indicarle a Turtle que dibuje una figura también que la rellene y luego usa **end_fill()** para decirle que puede dejar de rellenar.

Encanta a Turtle pintando un círculo de color amarillo, usa los comandos siguientes:

```
jose.fillcolor("yellow")
jose.begin_fill()
jose.circle(50)
jose.end_fill()
```



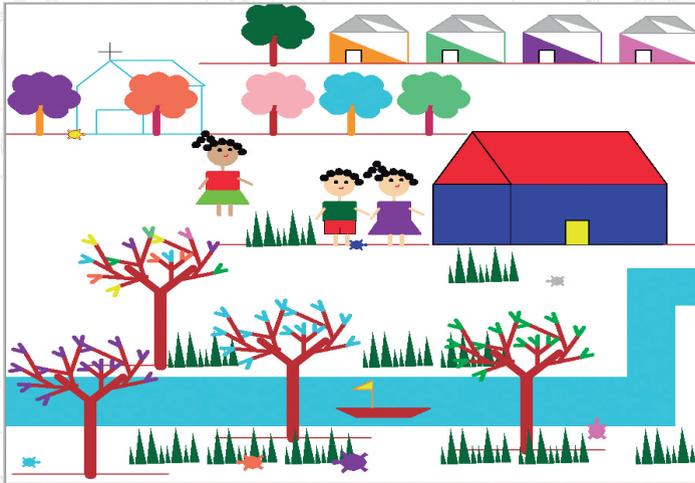
Los encantadores poseen poderes de creatividad e ingenio, y tú eres uno de ellos. Ayuda a Turtle a dibujar las casas de Anuk'pura. ***¡Usa tu varita mágica!***





¡Reconstruye cada parte del pueblo!

Este es un ejemplo.



¿Cómo guardo lo que llevo, para no perder nada?



Muy bien encantador, seguramente ya has reconstruido algunas partes de Anuk'pura, y si cierras el intérprete de Python o cierras el lienzo, perderás todo ***¡Oh, pero no queremos perder nada!***

Una rápida solución es guardar los comandos en un archivo, así podrás dar órdenes a Turtle desde el archivo y abrirlo cuantas veces quieras, podrás modificarlo y ejecutarlo siempre.



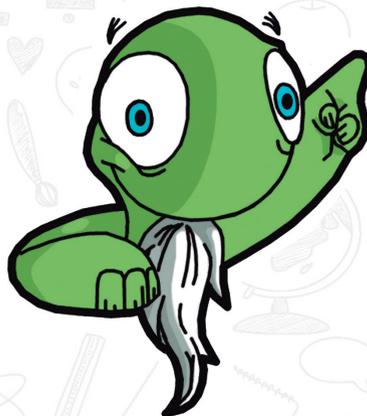
Pero **¿cómo ejecutarlo?** En programación llamamos **ejecutar o correr un archivo** cuando creamos un documento con un algoritmo (**serie de comandos u órdenes**) debemos prestar mucha atención al escribir dentro de ese archivo porque Turtle realizará solo lo que esté escrito en él.

Para guardar los comandos en un archivo, usaremos un **editor de textos**.

Pero, ¿Qué es un editor de textos? Es un programa que se encuentra en tu canaimita o tu computadora con el que podrás escribir lo que desees, los editores de textos te permiten crear archivos de solo texto , es decir, **sin imágenes**, y son tus aliados para la creación de programas.

Aunque existen muchos editores de textos, en tu Canaima cuentas con el editor **Gedit**.

¿Cómo llegamos a él? Primero hacemos clic en **Aplicaciones**, luego en **Accesorios** y finalmente en **Editor de textos gedit**.

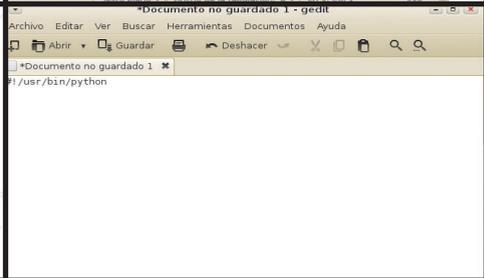




Se abrirá nuestro editor de textos, es una ventana como la de la imagen.



Ahora escribimos la primera línea en nuestro archivo, escribimos lo siguiente:
#!/usr/bin/python



Esta línea le indica a la máquina con que programa va a ejecutar este archivo.

Antes de seguir escribiendo es importante **guardar nuestro archivo**. Para ello haz clic en el botón guardar, que se encuentra en la barra de herramientas.



se abrirá una nueva ventana.

Lo siguiente es **colocar un nombre**, asegúrate que tenga la extensión **.py** al final del nombre. ejemplo: **MiTortuga.py**



En este caso **MiTortuga** corresponde al nombre y **.py** corresponde a la extensión o formato, sin el formato nuestro archivo no podrá ser ejecutado por python.

Ahora veremos que nuestra línea cambió a color azul, con un nombre y está guardado en nuestro escritorio.



Nuestra segunda línea será importar el módulo de Turtle, tal como lo hacemos desde el intérprete de python. Escribimos: **import turtle**.





Ahora escribimos los comandos para darle nombre, tamaño y color a Turtle.

jose=turtle.Turtle()// le damos un nombre a la tortuga, coloca el que desees.

maria=turtle.Turtle()// la damos un nombre a la tortuga, coloca el que desees

```
#!/usr/bin/python
import turtle

jose = turtle.Turtle()
```

¡Muy bien! lo has logrado

El último paso será ejecutar el archivo.

Para el siguiente ejemplo tomaremos los comandos para dibujar un cuadrado, los colocaremos en el editor y lo ejecutaremos, entonces aparecerá Turtle en el lienzo y comenzará a dibujar lo que le ordenaste.

Usa el # para escribir o explicar lo que le estás ordenando a Turtle que haga, a esto se le llama **Comentar un código**.

El último comando que colocarás siempre al final de tu archivo es: **raw_input()**

Éste comando le dirá a python que no cierre el lienzo para que puedas ver lo que Turtle ha dibujado.

```
#el nombre de la tortuga sera jose
ose = turtle.Turtle() #le digo a la tortuga que aparezca
ose.shape("turtle") #color de la tortuga
ose.color("blue") #color de lapiz sera purpura
ose.pencolor("purple")

ose.forward(100) #tamano del lado delcuadrado
ose.left(90) #gira 90 grados a la izquierda
ose.forward(100) #tamano del segundo lado delcuadrado
ose.left(90) #gira 90 grados a la izquierda
ose.forward(100) #tamano del tercer lado delcuadrado
ose.left(90) #gira 90 grados a la izquierda
ose.forward(100) #tamano del cuarto lado delcuadrado

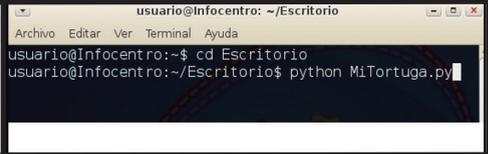
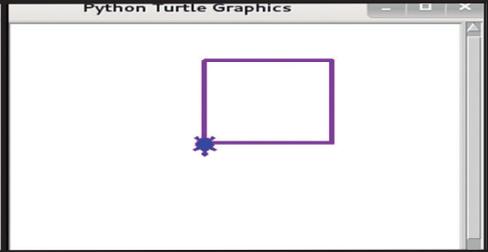
raw_input()
```

si has realizado todos los pasos anteriores, te estás convirtiendo en uno de los mejores encantadores, ¡Ya has creado tu primer programa en Python con Turtle a esto le llamamos **programar!**

¿Cómo ejecutamos o corremos nuestro archivo?

Es muy fácil, lo que tienes que hacer es abrir un terminal, buscar dónde has guardado el archivo y luego correrlo. Son solo 3 pasos.

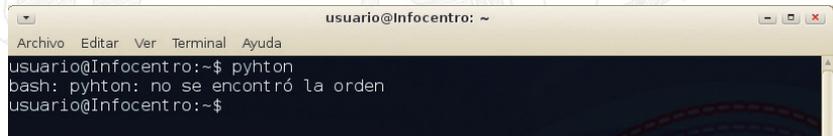


<p>Paso 1: abrimos un terminal. Hacemos clic en aplicaciones, luego en la nueva ventana hacemos clic en accesorios y por último en terminal.</p> <p>Aplicaciones->Accesorios->terminal</p>	
<p>Paso 2: buscar donde has guardado el archivo ¿Recuerdas cuando le diste nombre al archivo?, pues, en esa ventana decimos dónde se guardaría, y eso fue en el Escritorio. En esta ventana escribes cd Escritorio y presiona Enter</p>	
<p>Paso 3: correr el archivo con el siguiente comando python dejas un espacio y escribes el nombre del archivo luego presionas Enter. python MiTortuga.py</p>	
<p>Cuando presiones la tecla enter, podrás ver que aparecerá Turtle dibujando un cuadrado.</p>	

¡Felicidades, desde ahora podrás reconstruir Anuk'pura más fácil!

Instalación de python

Si has llegado a este punto, obtendrás el poder del instalador.





Lo primero que debes tener es la llave del instalador, esta llave es la **clave del administrador** también llamada **clave root** de tu computadora donde instalarás **Python**.

Le pediremos ayuda al **Gestor de paquetes Synaptic**, (también lo puedes encontrar como **Centro de Software**).

Para llegar hasta el Gestor, debes hacer clic en la imagen Aplicaciones



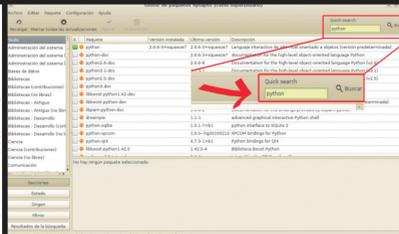
Ahora haces clic en:
Sistema -> Administrador -> Gestor de Paquetes Synaptic

¡Es hora de utilizar la llave del instalador!

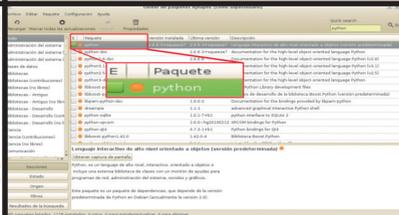
Copia la contraseña del administrador de tu computador. (clave de root)



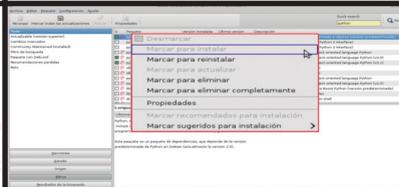
Una vez abierto el gestor de paquetes synaptic usa el buscador y escribe la palabra **python** en él



Si ves que se despliega una lista y entre ellos una que dice python quiere decir que tu poder de instalador esta funcionando. lo próximo será seleccionar el paquete python



Para ello, deberás hacer clic derecho con el ratón y hacer clic en la opción **Marcar para Instalar**





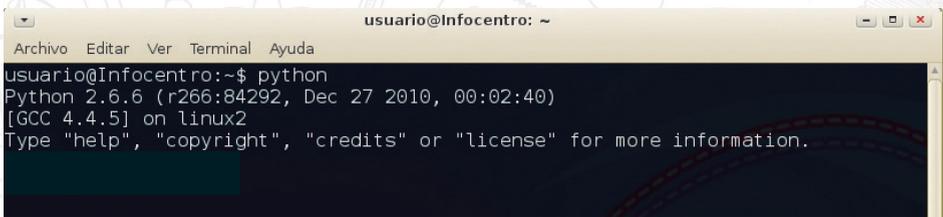
Luego hacemos clic en el botón Aplicar



¡Ya está! Ahora tienes acceso a Python.

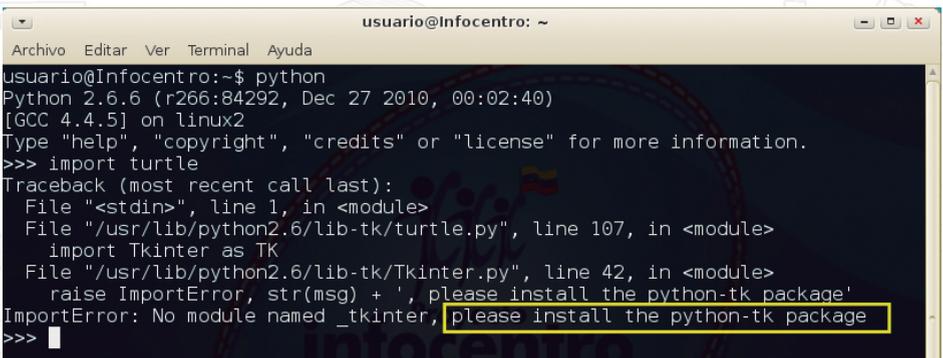
Para llegar a la serpiente, solo debes escribir Python en el Terminal o abrir la consola del programa desde el menú aplicaciones.

¡La aventura apenas inicia!



¡Vuelve al paso en el que quedaste antes de llegar a este punto para que empieces tu labor!

Instalación de Turtle

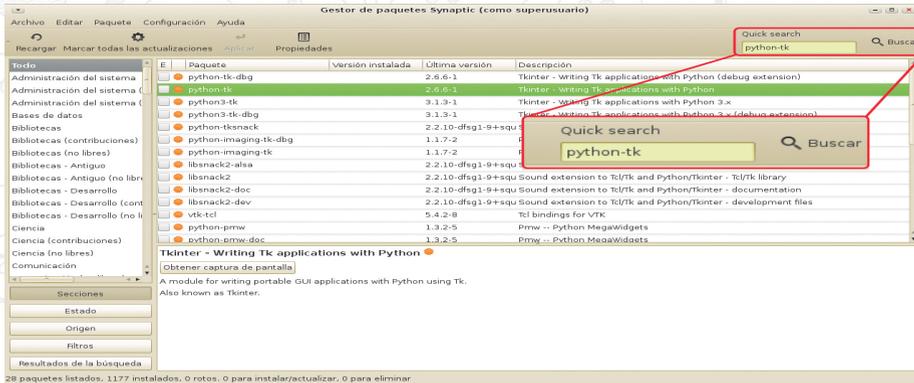


Si por alguna causa aparece la frase **please install the pythontk package**, debemos instalar el paquete indicado, en este caso **Python-tk**.



¡No te preocupes! Instalarlo es muy fácil!

Solo se debe abrir el gestor de paquetes Synaptic (como lo explicamos al principio) y colocar en el buscador **python-tk**.



Escribimos python-tk en el buscador

<p>Se desplegará una lista con todos los programas encontrados. Entonces seleccionamos python-tk y presionamos Aplicar</p>	
<p>Se abre entonces una ventana en la que debemos seleccionar Para ser instalado y luego presionar el botón Aplicar que se encuentra en la parte inferior de la ventana.</p>	
<p>Luego comenzará a instalarse y veremos el progreso.</p>	

Después de que lo instalamos, podemos volver al intérprete de Python y teclear **import turtle**, luego presionamos **Enter**. Si lo que se ve ahora son los símbolos **>>>** es porque ya puedes comenzar a emitir órdenes a la Tortuga.



Directorio

Ministerio del Poder Popular para Educación Universitaria, Ciencia y Tecnología

MANUEL FERNÁNDEZ

Ministro del Poder Popular para Educación Universitaria, Ciencia y Tecnología

JEHYSON GUZMÁN

Viceministro para Educación Universitaria

LILA CARRIZALES

Viceministra para la Investigación y Aplicación del Conocimiento.

CARLOS FIGUEIRA

Viceministro para el Fortalecimiento Institucional, la Conectividad y el Intercambio del Conocimiento.

Directorio

Infocentro

DAVID PARRA

Presidente

SANDINO MARCANO

Director Ejecutivo

JENNY MEJÍA

Consultora Jurídica

YORWEL PARADA

Coordinador General de la Red
Infocentro

VLADIMIR GUARACHE

Coordinador General de la Red
de Formación

PATRICIA ZAMBRANO

Gerente de Tecnología

JESÚS CASTRO

Gerente de Infraestructura

ALEJANDRO ARMAO

Director de Comunicación
Estratégica

JONÁS REYES

Director de Atención al
Soberano

NURY FASANELLA

Directora de Talento Humano (E)

JOSE LUIS HURTADO

Director de Planificación
Presupuesto y Control

Equipo de Publicación

EDGAR HEREDIA

Coordinador de
Publicación

ROGER ALTUVE

Diagramación e Ilustración

CARLOS MARRERO

Diseño

MIGDALIA MEDINA

CORINTIAN PALACIOS

LUZ GOMEZ

Desarrollo de Contenido
y Corrección

Agradecimiento

TRINO GOMEZ

Profesor de la UCV



