**Lenguaje de Programación Python**

Iniciamos este nuevo recorrido lleno de aventuras, con un lenguaje de programación de uso multiple. El cual voy a enfocar en el Análisis de Datos.

iniciamos utilizando la versión 3.6 de Python. recuerda que si quieres cambiar la versión por otra que tengas instalada. Python y sus versiones se instalan en el directorio **usr/bin**

Para inilciar vamos a conocer su funcionalidad básica (Sintaxis, definición de Clases, manejo de objetos e implementación de librerías que tiene por defecto. Esto, en programas de Consola), despúes pasaremos a la implementación de Puthon para interfaces gráficas, Proyectos Web y manejo de API’s. Posteriormente implementaremos python para El BigData, implementando las librerías Pandas, NumPy y Matplotlip, posiblemente con la ayuda de MongoDB y apache spark.

como pequeña intro, mencionar que Python es un lenguaje de programación interpretado, salvo por algunas metodologías o plataformas que compilan Python a lenguaje C.

Python es un lenguaje de Programación de alto Nivel, lo que facilita su aprendizaje porque está en capas superiores al lenguaje de máquina. se lanzó en el año 1991 y su creador es Guido Van Rossum, un informático de los países Bajos. Su comunidad ha creado poderosas librerías para el análisis de datos, programas que facilitan su uso como Jupyter notebook, y el plugin Eclipse PyDev para el todo poderoso Eclipse IDE, librerías para implementar Machine Learnig como scikit-learn y TensorFlow, entre otras muchas librerías. Existen también poderosos Frameworks para el desarrollo Web como Flask y Django. Y para el desarrollo de Aplicaciones de escritorio tenenmos a la librería Tkinter o a PyQT, dos poderosas librerías gráficas.

Python tambien se utiliza para Procesos automatizados o Scripting y para robotica con Rasberry Pi.

ahora vamos a “Picar Código”...

Importante mencionar que Python es un lenguaje Fuertemente tipado y dinámico, con lo cual no hay necesidad de indicar el tipo de variable antes de implementarla y hace que distinga entre tipos de variables. Es un Lenguaje de programación que permite varios paradigmas de Programación, pero la que nos interesa y que vamos a implementar es la Programación Orientada a Objetos. Su licencia es compatible conb la GNU/GPL aunque No es totalmente software Libre.

La instalación de Python y de Django está muy detallado en mi documento “Configuración de mi GNU Linux para desarrollo

Para iniciar entonces, creamos el repositorio local y en GirHub mediante GitKraken en el directorio Developer, con el nombre Python\_init

posteriormente creamos el proyecto de Python con Eclipse PyDev en el directorio Python\_init y lo llamamos de la misma manera... “Python\_init”.

Ahora creamos un nuevo archivo de python que llamaremos “Main”

**Bloques de Código en Python:..**

Los bloques de código en Python se ordenan e identifican con Identación, de manera que en este lenguaje de programación no se delimita un segmento de código con el clásico “**;**” sino que se ubica el contenido, identandolo a partir de su contenedor. Así, el contenido de una FX sería implementado como sigue...

**def nombre\_fx()**

**contenido**

**def bombre\_fx2()**

**contenido**

**Imprimir texto en pantalla...**

el “Hello Demonscript” lo implementamos de la sig manera...

**print(“Hello Demonscript”)**

**Comentarios...**

Los comentarios en Python sirven para informar al programador en cuestión sobre cómo funciona nuestro código.

Los comentarios, segun el pep8(guía de estilo adoptada ampliamente por la comunidad de Python), deben tener max 72 caracteres y si necesitamos comentarios mas extensos, se debe utilizar dos o más lineas de comentarios...

# Primera línea de comentario

#segunda línea de Comentario

#Tercera línea de comentario

**NOTA:** recordemos que la diferencia crucial entre comentarios y documentar código, es que los comentarios son utilizados por el mismo programador o por otro programador que va a manipular directamente nuestro código. Mientras que la documentación es mucho mas generalizada y explica en detalle el programa, no solo a programadores, sino a personas que van a implementar o utilizar nuestro programa

**Documentación de nuestro Código en Python o Docstring...**

Un apartado muy importante es el de la documentación es el de los Docstring o cadenas de Documentación, que sirven para documentar nuestro código, de manera que se pueda explicar el funcionamiento general de nuestro código y poder exportar dicha Documentación a archivos externos y separados del mismo código, como también sirven mucho en el atocompletado o Intellisense de nuesto IDE.

La diferrencia Con PHP es que los Docstring el Python se hacen dentro del contenedor del códgo(dentro de la clase, del Método, e imagíno que al inicio de cada paquete o Módulo).

Otra diferencia con PHP son los caracteres que se utilizan para implementar Docstring, que son comilla simple y Triple comillas, donde...

**Docstring simple:...**

Si queremos implementar un Doctring de una sola línea, lo implementamos entre Comilla Simple...

**‘Comentario de Una sola Línea’**

**Docstring multi-líneas:...**

Según el PEP257(guía de estilo adoptada por la comunidad Python), la primera línea de este tipo de Docstring dbe ser a modo de título o resumen general y termina con un punto. Importante que esté separada del resto del Docstrong por un salto de línea. Esta línea de resumen puede ser utilizada por los motores de Indexación.

El resto de la cadena de documentación debe describir el comportamiento del código documentado, los valores devueltos, las excepciones que arroja y detalles relevantes.

se recomienda dejar un salto de línea antes de cerrar las tríples comillas.

Se puede utilizar docstring en cualquier fragmento de código, tanto en Clases, métodos, paquetes, módulos, etc.

**Inspección de nuestro Docstring:...**

* Podemos imprimir la documentación de nuestro código, el cual siempre dispone de un método asignado \_doc\_ o con la FX help() pasandole como argumento el nombre de la Clase, del metodo etc...

**NOTA:..**.

Por algúna razón que desconozco, al utilizar **help(nombre\_método)** el mensaje es que No hay Documentación para dicho método, aunque si imprime el contenido del Docstring

Aquí un breve ejemplo de cómo implementar Docstrging y de cómo imprimir su contenido...

**class Master:**

**"""**

**Clase contendora.**

**Esta clase es para ensayar**

**No tiene atributos**

**tiene un método**

**"""**

**def suma(self,a,b):**

**"""**

**Metodo suma.**

**Esta función recibe como parámetros...**

**@param a: Int**

**@param b: Int**

**@return: None**

**"""**

**print(a+b)**

**print("Hello Demon!!!")**

**obj = Master**

**print(obj.suma.\_\_doc\_\_)**

**print("Ahora imprimimos el Docstring de la Clase...")**

**print(Master.\_\_doc\_\_)**

**print("Ahora mostramos lo mismo con help...")**

**help(obj.suma.\_\_doc\_\_)**

* La otra manera de inspeccionar el Docstring de nuestro código es Exportar el Docstring de nuestro proyecto, que generalmente se exporta en formato HTML, lo que hacemos es mostrar la documentación completa del proyecto. Creo que la manera más profesional de generar documentación de es Doxygen, sphinx y Pydoc que en escencia hacen lo mismo... exportar la documentación de un archivo .py.

Pero yo me quedo con Doxyen qye es un programa externo, pero que me funciona muy bien para generar la documentación, no solo de Python, pero también de otros poderosos como PHP, C++ y Java. En otra ocasión tal vez veamos cómo funciona sphinx y pydoc.

**Sintasis Básica de nuestro lenguaje de Programación Python:...**

ya vimos parte de la escritura en Python(cómo imprimir texto en la consola, cómo comentar código y algo muy importante.. Cómo Documentar nuestro código de acuerdo a las convenciones de la comunidad mundial de programadores Python).

**tipos, operadores y variables en Python:...**

**NOTA:...** En programación, **una sentencia** es la mínima unidad o fragmento de código que el interprete en el caso de python o el compilador en el caso de lengauejs compilados, pueden ejecutar. Por ejemplo...

**xn = 4**

la anterior es una sentencia que no presenta resultados.

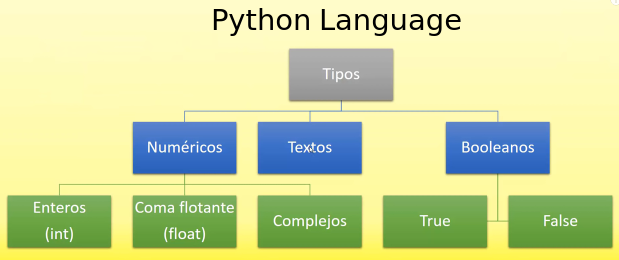
**print(“Hola”)**

la anterior es una expresión que presenta un mensaje

Una **Expresión** es la convinación de variables, valores y operadores, aunque una **sentencia** también la podemos definir como Expresión, puntualmente una expresión sería...

**var1 = 4 + 23**

**Los tipos** básicos en Python son los mismos que en otros lenguajes de programación comúnes, esos tipod se agrupan de la siguiente manera...

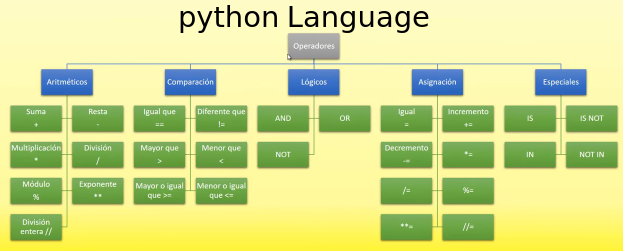


Podemos ver que los tipos numéricos se organizan en Números enteros, en Tipo Float(decimales o de coma flotante) y Números complejos(que son un tipo de números que complementan a los números reales, son números “imaginarios” utilizados ára realizar operaciones algebraicas)

Los datos de tipo Texto son cadenas alfanuméricas que están contenidas entre comillas.

Los datos Booleanos con utilizados para toma de deciciones, comparaciones, etc.

**Operadores en Python**. disponemos de diversos operadores para determionadas operaciones que seguramente realizaremos en lo corrido de nuestro aprendizaje...



Existen otros operadores en Python, pero son muy poco utilizados, de manera que nos enfatizamos en los más comunes.

Podemos ver que los aritmeticos los hemos implementado en PHP, al igual que los de comparación, los lógicos(que se utilizan en toma de decisiones como booleanos) y los de asignación también nos son familiares gracias a C++ y PHP de manera que a este nivel son muy entendibles. Los Operadores especiales se utilizan en la inspeccipon de cadenas de texto u otros valores, en Bucles y en consultas BD.

De los operadores aritméticos vamos a implementar un código para utilizar el módulo, división entera y exponente, que son los que difieren en sintaxis de otros lenguajes de prog..

**#el módulo de una división...**

**print(7 % 3)**

**#división entera...**

**print(7 // 3)**

**#operador de exponente...**

**print(7 \*\* 3)**

**Las Variables en Python:..**

las variables(espacios de la memoria al que le asignamos una clave o identificador y un valor contenido en ella, valor que puede variar en la trayectoria del programa) son manejadas de una manera muy similar a PHP, donde, para implementarlas no hay que definir su tipo, Python lo asigna de manera dinámica, de acuerdo al valor que asignemos dentro de la variable.

Se recomeinda utilizar nombres semánticos en las variables, clases y Métodos(nombres que reflejan el contenido y funcionalidad del código o valor contenido).

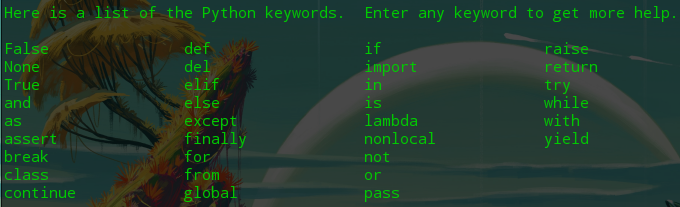
Los nombres de Variables NO deben comenzar con Números, puntos o caracteres especiales y tampoco debe contener puntos o caracteres especiales.

se aceptan números y guión bajo como parte de su nombre, pero NO al inicio del nombre de dicha Variable, Método o Clase.

Tampoco podemos declarar variables, dándoles nombre de palabras “clave” de Python, las conocemos como “palabras reservadas del lenguaje”. Hasta la versión 3.6 Python tenía 33 palabras reservadas. Podemos ejecutar la ayuda para ver en cada versión las palabras reservadas, con el comando...

**help(‘keywords’)**

con lo que obtenemos en pantalla...



En Python Todo es un objeto. Y lo podemos comprobar al mostrar el tipo de una variable cualqiuera...

**num1 = 39**

**print(type(num1))**

en pantalla podemos ver el resultado...

**<class 'int'>**

Para concatenar fragmentos de texto utilizamos el operador **+**

**text1 = "Hello Demonscript "**

**text2 = "Estas aprendiendo Python"**

**union\_text = text1 + text2**

**print(union\_text)**

**#otra manera de concatenar es en el moento de imprimir..**

**print(text1 + text2)**

En ambos casos obtenemos como resultado...

**Hello Demonscript Estas aprendiendo Python**

lo cual nos indica que la variale es un tipo de objeto que pertenece a la clase “entero”

Para concatenar texto con tipos que no son texto, como números u otro tipo, utilizamos la conversión a texto del valor a concatenar por medio del método **str()...**

**num1 = 39**

**print("El número "+str(num1)+" es de tipo "+str(type(num1)))**

y como resultado obtenemos...

**El número 39 es de tipo <class 'int'>**