Universidad Nacional del centro de la Provincia de Buenos Aires

Tecnología de Información en Organizaciones

Trabajo Práctico Especial

Alumnos:

Fecha de Entrega:

###Informe sobre Python:

Python es un lenguaje de programación esencial para los informáticos en desarrollo web. Es un lenguaje dinámico que puede ser interpretado en varias plataformas, esto permite la creación de sitios web y aplicaciones multiplataforma como: \*\*iOS, Android, Windows o Mac\*\*. Python cuenta con una sintaxis \*\*limpia, clara y altamente legible\*\*.

Es conocido por su sencillez, este con pocas líneas de código se podrá programar algoritmos complejos con resultados satisfactorios. Es un lenguaje interpretado, lo cual da la posibilidad de ahorrar al informático mucho tiempo durante el desarrollo ya que no es necesario compilar ni enlazar.

El intérprete puede usarse interactivamente, lo que facilita experimentar con características del lenguaje, escribir programas descartables o probar funciones cuando se hace desarrollo de programas de abajo hacia arriba.

Como se menciona anteriormente, el intérprete es utilizable de forma interactiva y permite crear y borrar con facilidad; Esto permite escribir programas compactos y legibles. Python es recomendable ya que los programas son mucho más cortos que sus programas equivalentes en C, C++ o Java por varios motivos:

* Los tipos de datos de alto nivel permiten expresar operaciones complejas en una sola instrucción
* La agrupación de instrucciones se hace por sangría en vez de llaves de apertura y cierre
* No es necesario declarar variables ni argumentos.

El intérprete de Python y su biblioteca estándar, están a disposición de quien lo necesite de forma binaria y de código fuente para las principales plataformas desde el sitio web, \*\*htpp://[www.python.org/](http://www.python.org/)\*\*.

Es un lenguaje de programación el cual ofrece una fuerte estructura y soporte a programas grandes. Ofrece un chequeo de error más grande que \*c\*. Permite separar un programa en módulos los cuales pueden “reusarse” en otros programas; Posee una colección considerable de módulos en forma estándar, los cuales se pueden usar como base de programas para desarrolladores. Algunas de las cosas que estos módulos proveen son: Entrada/salida a archivos, llamadas al sistema, sockets, interfaz a sistemas de interfaces gráfica de usuario.

## Iteraciones:

Se denomina iteración a aquel conjunto de instrucciones que se encuentran acotadas por un bloque repetivo de acciones indicadas a través de comandos indicadas en el lenguaje Python. Las instrucciones por excelencia en este lenguaje pertenecen a los bloques \*for\* y \*while\*. Estos bloques poseen identacion para reconocer el bloque como tal.

Cuando iteramos sobre diccionarios, se pueden obtener al mismo tiempo la clave y su valor correspondiente usando el método \*iteritems()\*

.

Cuando se itera sobre una secuencia, se puede obtener el índice de posición junto a su valor correspondiente usando la función \*enumerate()\*

.

Para iterar sobre dos o más secuencias al mismo tiempo, los valores pueden emparejarse con la función \*zip()\*

Para iterar sobre una secuencia en orden inverso, se especifica primero la secuencia al derecho y luego se llama a la función \*reversed()\*

Para iterar sobre una secuencia ordenada, se utiliza la función \*sorted()\* la cual devuelve una nueva lista ordenada dejando a la original intacta.

##Funciones, definicion:

La ejecución de una función introduce una nueva tabla de símbolos usada para las variables locales de la función. Más precisamente, todas las asignaciones de variables en la función almacenan el valor en la tabla de símbolos local; así mismo la referencia a variables primero mira la tabla de símbolos local, luego en la tabla de símbolos local de las funciones externas, luego la tabla de símbolos global, y finalmente la tabla de nombres predefinidos.

Así, no se les puede asignar directamente un valor a las variables globales dentro de una función (a menos se las nombre en la sentencia global), aunque si pueden ser referenciadas.

La primer sentencia del cuerpo de la función puede ser opcionalmente una cadena de texto literal; esta es la cadena de texto de ocumentación de la \*función\* o \*docstring\* (cadena de documentaciòn).

Los parámetros reales (argumentos) de una función se introducen en la tabla de símbolos local de la función llamada cuando esta es ejecutada; así, los argumentos son pasados por valor (dónde el valor es siempre una referencia a un objeto, no el valor del objeto).

Cuando una función llama a otra función, una nueva tabla de símbolos local es creada para esa llamada.

La definición de una función introduce el nombre de la función en la tabla de símbolos actual.

La palabra reservada \*def\* se usa para \*definir funciones\*. Debe seguirle el nombre de la función y la lista de parámetros formales entre paréntesis. Las sentencias que forman el cuerpo de la función empiezan en la línea siguiente, y deben estar con sangría.

También es posible definir funciones con un número variable de argumentos.

Hay tres formas que pueden ser combinada:

* \*Argumentos con valores por omiciòn\*

* \*Palabras claves como argumento\*
* \*Lista de argumentos arbitrarios\*

#Ejemplo

Podemos crear una función que escriba la serie de Fibonacci hasta un límite determinado:

>>> def fib(n): # escribe la serie de Fibonacci hasta n

... """Escribe la serie de Fibonacci hasta n."""

... a, b = 0, 1

... while b < n:

... print b,

... a, b = b, a+b

...

>>> # Ahora llamamos a la funcion que acabamos de definir:

... fib(2000)

1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597

Bibliografía:

<http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>