Desarrollo de aplicaciones web Python y Web2py

Evento Patrocinado LATINUX

Python

- Lenguaje creado por Guido Van Rossum
- Es un lenguaje de alto nivel, intepretado/pseudocompilado
- Se enfoca en tener una sintaxis muy limpia
- Multiparadigma
- Es de tipado dinámico
- Multiplataforma
- Incluye baterías ;)

Python

- Python fue creado en el Instituto Nacional de Investigaciones Matemáticas y de Ciencias Informáticas de Holanda para ser un sucesor del lenguaje ABC
- Guido es el Benevolente Dictador Vitalicio (BDFL)
- Python tiene varias implementaciones en distintas plataformas de lenguajes (www.python.org)
  - Jython - JythonDroid
  - IronPython

Python – Características y Paradigmas

- Al ser multiparadigma, no se obliga al programador a adoptar un estilo particular
- Python trata todo como un objeto
- Python puede ser empleado en
  - Programación estructurada
  - Programación Orientada a Objetos
  - Programación Funcional
  - Programación Imperativa

Python y la filosofía SL

- Diseñado para su fácil extensión
- Al ser un proyecto libre, toma y entrega características de/para otros lenguajes y plataformas
  - Perl: regex
  - Java: logging
  - Lisp: lambda(), reduce(), map(), filter()

Zen de Python

- Hermoso es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Complejo es mejor que complicado.

  … etc etc etc …

  - Lo podemos ver en el interprete al ejecutar import this

Intérprete y el Modo Interactivo

- Python incluye un modo interactivo donde podemos ejecutar cualquier sentencia
- Las instrucciones recibirán su resultado de forma inmediata
- Esto resulta útil tanto para las personas que se están familiarizando con el lenguaje como también para los programadores más avanzados

El Lenguaje y su sintaxis

- Los bloques de código están definidos por los dos puntos “:”

```
def mifuncion(x):
    print “hola curso”, x
```

- Utiliza espacios para definir el código perteneciente a un bloque
  - Espacios o Tabuladores. La consistencia es obligatoria
- Es una sintaxis limpia, clara y muy especifica
  - `If x is not y: print(“X no es Y”)`

Operadores Básicos

- + Suma y concatenación de cadenas de caracteres
- - Resta y números negativos
- * Multiplicacion
- ** Exponente
- / Division
- % Modulus

Tipos de datos

- Enteros (Int)
- Long
- String/Unicode
- Complejos (complex)
- Tupla (tuple)
- Lista
- Conjunto (set)
- Diccionario
- Float
- Booleans
- Complejo

Baterías Incluidas??

- raw_input()
- len()
- max()
- in
  - Ej: 'ton' in 'Python' evaluará False

Formatos de Strings

- %s = Strings
- %d = Números

ej: “Se llama %s y tiene %d hijos” %('Pedro', 2)

Métodos de Strings

- split()
- join()
- find()
- strip()
- lower()/upper()
- replace()
- ljust()
- center()

Métodos de Strings

- rjust()
- swapcase()
- Isalpha()
- endswith()

Listas y Tuplas

- Listas son mutables, las tuplas no
- Para declarar listas, se usan corchetes
- Para declarar tuplas, se usan paréntesis
- Tanto las listas como las tuplas, pueden tener elementos diversos
- Las listas se utilizan generalmente para grupos de elementos variables en cantidad
- Las tuplas se utilizan para grupos de elementos estáticos o de cantidad fija

Métodos de Listas

- append()
- insert()
- remove()
- pop()
- reverse()
- sort()
- sorted()
- extend()

Indices Python y Slicing

- Característica muy poderosa de manejo de cadenas, listas y tuplas
- Permite una precisión única, sólo obtenible en otros lenguajes empleando mucho código
- Permite anidar sobre sentencias previamente anidadas de forma ilimitada
- Se realiza con indices positivos y negativos entre corchetes
- Para obtener un pedazo (slice) de mas de un caracter, se utilizan los dos puntos “:”

Indices Python y Slicing

```
>>> x = 'Esta es mi cadena de caracteres'
>>> x[0]
'E'
>>> x[1]
's'
>>> x[-1]
's'
>>> x[-2]
'e'
>>> x[0:4]
'Esta'
>>> x[10:]
' cadena de caracteres'
```


Indices Python y Slicing

```
>>> x = ['abc', 'def']
>>> x[0]
'abc'
>>> x[-1]
'def'
>>> x[0][1]
'b'
>>> x[0][0:2]
'ab'
```


Diccionarios

- Conocidos tambien como matrices/arrays asociativos
  - Son componentes basados en llave:valor
- Son modificables y pueden tener cualquier tipo de valor
- Son colecciones de datos llave/valor
- Son declarados con llaves “{}”
  - Ej: `xdict = {'mistring': 'python', 'miint': 12345}`

Métodos de Diccionarios

- get()
- update()
- pop()
- clear()
- items()
- fromkeys()
  - Ej: for k, v in x.items(): print("%s: %s" k, v)