



FACULTAD
CIENCIAS DEL HOMBRE Y LA NATURALEZA
LICENCIATURA CIENCIAS DE LA COMPUTACIÓN

DOCENTE: *Pedro Antonio Trejo Noble*

ASIGNATURA: *Algoritmo II*

PROYECTO: *Sopa de Letras en Python*

INTEGRANTES: *Diego Rolando Olmedo Melgar*

Luis Miguel Martir Martínez

Elvis Henry Fabián Martínez

INTRODUCCIÓN

*En el presente trabajo de la materia de Algoritmo II consistirá en llevar a cabo un proyecto el cual sera desarrollado en el Lenguaje de Programación Python. Como grupo se ha tomado la decisión de desarrollar una “**Sopa de Letras**”, esta consistirá en hallar la cantidad de palabras que ordene el usuario siguiendo las instrucciones, cada una de estas palabras llevara un orden aleatorio de fila y columna (horizontal, vertical, rectas o cruzadas, etc).*

OBJETIVOS

GENERAL:

Aprender a usar el lenguaje de programación Python elaborando el juego de una sopa de letras.

ESPECÍFICOS:

- × Identificar los códigos que se utilizaran en dicho juego*
- × Utilizar un método para hacer funcionar el juego*

MARCO CONCEPTUAL

Ya se declararon las variables y se aplicaron algunas estructuras de control y algunas condiciones.

Para intentar correr el código nos da una serie de errores en los cuales estamos trabajando.

HISTORIA Y ANTECEDENTES

Historia de un joven lenguaje de programación: Python Por: Reclu IT

9 de junio de 2014

Notas relacionadas

Comunidad e innovación al crear PHP

Creciendo en compañía de Ruby on Rails

Historia de un joven lenguaje de programación: Python

Datos que probablemente desconocías del lenguaje C

El tiempo a través de HTML

Las actualizaciones y cambios en la industria de la tecnología sucede de una manera vertiginosa, por ello surgen de forma constante nuevas tendencias, modelos de negocio y lenguajes de programación, por ello hoy nos adentraremos en la historia de Python, que es algo joven.

La razón por la que se denomine como un lenguaje joven, se debe a que sus inicios se remontan a principios de los 90s, de manera específica la primera versión vio la luz en 1991, pero fue hasta después de tres años cuando se publica la versión 1.0.

Python fue desarrollado por el investigador holandés Guido van Rossum, quien en aquellas épocas trabajaba en el centro de investigación CWI (Centrum Wiskunde & Informatica) de Ámsterdam.

A van Rossum le fue asignado un proyecto en el que tenía que desarrollar un sistema operativo distribuido llamado Amoeba, las condiciones en que se encontraba el CWI, es que tenía implementado un lenguaje de programación llamado ABC.

Guido decidió crear un nuevo lenguaje para el proyecto Amoeba, para superar las limitaciones y problemas con los que se encontró. Una vez presentado Python, CWI decidió liberar el intérprete del lenguaje bajo una licencia open source propia.

En tanto, para el inicio del siglo XXI y coincidiendo con la publicación de la versión 1.6, se tomó la decisión de cambiar la licencia por una que sea compatible con GPL (General Public Licence). Esta nueva licencia se denominará Python Software Foundation Licence.

El anterior hecho implica que es posible modificar el código fuente y desarrollar código derivado, sin la necesidad de hacerlo open source. Hasta el momento sólo han sido liberadas tres versiones principales, teniendo cada una de ellas diversas

actualizaciones.

Algunas curiosidades acerca de Python son:

Nombre. Guido lo bautizó Python en honor a la serie televisiva Monty Python's Flying Circus, de la cual, como se supondrá, era seguidor. Esto fue debido al hecho de que Guido van Rossum visualizó que el principio del diseño del lenguaje fuera divertido de utilizar, de ahí que el nombre fuera ese.

Variables y conceptos en honor al programa. También resulta curioso que, tanto en tutoriales, como en ejemplos de código, se suelen utilizar referencias a los Monty Python. Por ejemplo, en lugar de emplear los tradicionales nombres de variables foo y bar, se suele utilizar spam y eggs, en referencia a sketches de este grupo de cómicos

Zen del Python. Se trata de los principios básicos de la filosofía con que se maneja Python, los cuales son tan prácticos y han hecho de este lenguaje algo tan sencillo y agradable que incluso podemos encontrar libros de Python para niños. Además tenemos el caso de que las comunidades de desarrolladores son tan abiertas que a veces abarcan campos que parecieran no tener mucho en común con la programación, como lingüistas o entomólogos. Desde la versión 2.1.2, Python incluye estos puntos (en su versión original en inglés) como un "easter egg" que se muestra al ejecutar `import this`.

Este lenguaje tiene un claro carácter open source, por esa razón la Python Software Foundation invita a los desarrolladores a contribuir en la mejor y promoción de Python.

¿Qué son los paradigmas de programación?



*A lo largo de la historia, el término «paradigma» fue objeto de muchas interpretaciones. En su origen griego, significaba «modelo», «ejemplo» o «patrón». Sobre este punto de partida, podemos hablar de un paradigma como un **conjunto de creencias, prácticas y conocimientos que guían el desarrollo de una disciplina durante un período de tiempo**. En diversas ramas de la ciencia, un conjunto de ideas en vigencia puede ser reemplazado drásticamente por otro que entre en conflicto con él y se demuestre más acertado. La programación tiene sus propios paradigmas, pero el término «paradigma de programación» no necesariamente representa un modelo único que deba ser respetado hasta que aparezca otro mejor. De hecho, actualmente muchos paradigmas coexisten en armonía.*

Un paradigma de programación es un estilo de desarrollo de programas. Es decir, un modelo para resolver problemas computacionales. Los lenguajes de programación, necesariamente, se encuadran en uno o varios paradigmas a la vez a partir del tipo de órdenes que permiten implementar, algo que tiene una relación directa con su sintaxis. ¿Cuáles son los principales paradigmas de programación?

- **Imperativo.** *Los programas se componen de un conjunto de sentencias que cambian su estado. Son secuencias de comandos que ordenan acciones a la computadora.*
- **Declarativo.** *Opuesto al imperativo. Los programas describen los resultados esperados sin listar explícitamente los pasos a llevar a cabo para alcanzarlos.*
- **Lógico.** *El problema se modela con enunciados de lógica de primer orden.*
- **Funcional.** *Los programas se componen de funciones, es decir, implementaciones de comportamiento que reciben un conjunto de datos de entrada y devuelven un valor de salida.*

- **Orientado a objetos.** El comportamiento del programa es llevado a cabo por objetos, entidades que representan elementos del problema a resolver y tienen atributos y comportamiento.

Otros son de aparición relativamente reciente y no forman parte del grupo principal:

- **Dirigido por eventos.** El flujo del programa está determinado por sucesos externos (por ejemplo, una acción del usuario).
- **Orientado a aspectos.** Apunta a dividir el programa en módulos independientes, cada uno con un comportamiento bien definido.

Cada paradigma es ideal para la resolución de un conjunto de problemas particular, por lo que no puede decirse que uno sea necesariamente mejor que otro.

JUSTIFICACIÓN

El siguiente proyecto se justificara de la siguiente manera:

¿Por que es novedoso este proyecto?

Es novedoso porque se esta poniendo en practica el lenguaje de programación Python para poder crear un juego.

Factibilidad

Es accesible ya que se estaría utilizando Python el lenguaje de programación mas común.

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

En el siguiente programa se pretende desarrollar la aplicación y los usos de los algoritmos aplicados en python para desarrollar un programa innovador y práctico para el usuario.

CRONOGRAMA DE ACTIVIDADES

Paso 1 . *Primeramente nos reunimos en grupo para elegir que tipo de proyecto se iba a llevar a cabo ; luego como grupo decidimos crear un juego de sopa de letras en python.*

Paso 2 . *Planteamos un objetivo general y dos específicos con el fin de establecer las metas que queremos alcanzar en este proyecto.*

Paso 3 . *Redactamos la introducción de este proyecto en el cual describimos en que consistirá en este proyecto y cada una de sus partes*

Paso 4 . *Elaboramos una justificación en donde nos hacemos la interrogante ¿Por que es novedoso este proyecto ? Y ahí nosotros justificamos y damos solución a la pregunta y la factibilidad de hacer una sopa de letras en python.*

Paso 5 . *En este paso concluimos que en este proyecto debemos aplicar de una forma lógica y ordenadas las estructuras de control; también concluimos que debemos poner en practica el ejemplo de programación para contribuir al entretenimiento del usuario.*

Paso 6 . *Agregamos el marco conceptual al documento en donde pusimos la descripción de la solución propuesta y también la historia y antecedentes del lenguaje de programación de uso de este proyecto.*

DIAGRAMA DE ACTIVIDADES

CONCLUSIÓN

En conclusión este proyecto no esta haciendo desarrollar mas los conocimientos sobre el uso y aplicación de las estructuras de control ya que deben ser aplicados de una forma lógica y ordenada; este proyecto tiene como fin ademas de poner en practica un ejemplo mas de programación, contribuir al entretenimiento ya que dicho código al ejecutarse puede adaptarse a las exigencias del usuario.

ANEXOS

```
sopa.py (-/Escritorio)
Archivo Editar Ver Buscar Herramientas Documentos Ayuda
#'''TP1, sopa de letras'''
print "Bienvenido a mi juego de sopa de letras, espero q te diviertas."
print "El juego funciona de la siguiente manera, primero un usuario elige la longitud de la"
print "tabla y las palabras que tendra la sopa de letras, las palabras no pueden tener ,"
print "acentos, apostrofes, o cosas que se la parezcan."
print "Una vez ingresadas las palabras, un usuario diferente debe ingresar las coordenadas"
print "que conncuerden con los numeros y letras al costado de la tabla."
print "Las coordenadas deben seguir una estructura parecida a esta 3A 3G. solo puede utilizar letras mayusculas."
print "Hora de jugar"
print "\n"
import random
import string

def letra_aleatoria():
    """Genera una letra aleatoria del abecedario"""
    return random.choice(string.ascii_lowercase)

def entero(minimo,maximo,mensaje):
    """Recibe un numero, si este esta entre el maximo y el minimo
    indicado lo guardara en la funcion"""
    cantidad = raw_input(mensaje)
    while not (cantidad.isdigit() and int(cantidad)>=minimo and int(cantidad)<=maximo):
        cantidad = raw_input("su numero fue incorrecto,ingreselo nuevamente: ")
    return int(cantidad)

def ingresar_a_lista_de_palabras(minim,maxim,cantidad_p):
    """Recibe una cantidad equivalentes a un valor ingrsado anteriormente
    en la variable cant_p ingresada anteriormente, guarda las palabras
    en una lista llamada lista de palabras"""
    palabra = ""
    lista_palabras = []
```

BIBLIO GRAFÍA

*http://ww
w.recluit.
com/histo
ria-de-
un-joven-
lenguaje-
de-
programa*

cion-python/#.Wp4At2rOW1s