

REFERENCIA DE PYTHON PARA EMPEZAR

Un buen sitio para aprender sobre estos temas: <http://www.mclibre.org/consultar/python/>

Asignación de Variables

<code>variable = expresión</code>	Las variables son nombres que almacenan valores. Los valores pueden ser de cualquier tipo. Ej: <code>x=1</code> <code>edad=32</code> <code>nombre='Sebastian'</code>
-----------------------------------	---

Expresiones (pueden ser valores de diferentes tipos y operaciones)

<code>int</code>	Números enteros. Ej. 5, 10, 0, -7, 20, etc.
<code>float</code>	Números con decimal. Ej. 1.0, 5.5, 7.8, etc.
<code>str</code>	Cadenas de letras. Ej. "hola", "chau", "sol"
<code>bool</code>	<code>True</code> o <code>False</code> (Verdadero o Falso)
<code>5 + 5</code>	10
<code>4.0 + 1</code>	5.0
<code>int(4.0+1)</code>	5
<code>str(4.0+1)</code>	'5'
<code>bool(0)</code>	False
<code>'5' + 'hola'</code>	'5hola'
<code>5 + 'hola'</code>	error (no se puede sumar int con str)
<code>+ - * /</code>	Suma, resta, multiplicación, división
<code>and, or</code>	y, o (sirven para unir valores tipo bool)

Los bloques en Python siempre llevan una sangría de 4 espacios.

<pre>if [expresión tipo bool]: [bloque de código] elif [expresión tipo bool]: [bloque de código] else: [bloque de código]</pre> <p>Ejecuta el bloque de código si y solo si la expresión es True. Si no, ejecuta el bloque "else".</p>	<pre>for [variable] in [lista]: [bloque de código] continue #(opcional) break #(opcional)</pre> <p>Asigna los valores de la lista consecutivamente a la variable y repite hasta acabar la lista.</p>	<pre>while [expresión tipo bool]: [bloque de código] continue #(opcional) break #(opcional)</pre> <p>Repite el bloque de código mientras la expresión sea True.</p>
---	---	---



Ejemplo 2.0 – colgadito.py

```
# -*- coding: utf-8 -*-
# ** COLGADITO **
#

import os
import random

colgadito = '''
+-----+
|               |
|               |
|       [#]      |
|       {        |
|       }        |
|     /\ . \    |
|   ( ^ )       |
| ~\ - /~       |
|   |           | % |
| // \| --- \| \\ \|
| " ! ^ ! "    |
| || || ||     |
| V  V         |
|             |
|             |
'''


def limpiar():
    os.system("clear")


def ojos_locos(linea, intentos):
    if intentos:
        return linea

    linea = linea.replace(".", "+ +")
    linea = linea.replace("\\-/\"", "\\v/")
    return linea


def pintar(intentos):
    lineas = colgadito.splitlines()

    # Poste
    for linea in lineas[:7]:
        print linea

    # Cabeza
    for linea in lineas[7:11]:
        print ojos_locos(linea, intentos) if intentos < 5 else "|"

    # Cuerpo
    for linea in lineas[11:13]:
        print linea if intentos < 4 else "|"

    # Piernas
    for linea in lineas[13:15]:
        print linea if intentos < 3 else "|"

    # Pies
    for linea in lineas[15:]:
        print linea if intentos < 2 else "|"


archivo = open("frases.txt", "r")
todas = unicode(archivo.read().decode("utf-8"))
# todas es un str con todas las frases que se lee del archivo frases.txt
posibles = todas.splitlines()

# elegimos una frase al azar
```

Funciones

También es posible definir bloques de código para ser usados muchas veces.

Estos se llaman funciones.
Para definirlas usamos **def**

```
def [funcion] ([parametros]):  
    [Bloque de código]  
    return [resultado]
```

Luego podemos invocar (llamar) a la función por su nombre.

Ej: `funcion()`



```

frase = posibles[random.randint(0, len(posibles)-1)]
acertadas = " "
# definimos que los espacios en las frases equivalen a letras acertadas

erradas = ""
intentos = 5
ganaste = False

total_letras = len(frase) - frase.count(" ")

while intentos>0:
    limpiar()

    pintar(intentos)

    total = 0
    for letra in frase:
        if letra=="á":
            letra = "a"
        if letra=="é":
            letra = "e"
        if letra=="í":
            letra = "i"
        if letra=="ó":
            letra = "o"
        if letra=="ú":
            letra = "u"
        if letra in acertadas:
            print letra,
            total = total + 1
        else:
            print "_",

    if total==len(frase):
        ganaste = True
        break

    print "\n\nDescubre la frase secreta. Tiene", total_letras , "letras!"
    print "Te quedan", intentos, "intentos."

    if erradas:
        print "Ya has descartado las siguientes:", erradas

    respuesta = unicode(raw_input().decode('utf-8'))

    if len(respuesta)!=1:
        continue

    if respuesta in frase:
        acertadas = acertadas + respuesta
    else:
        erradas = erradas + respuesta
        intentos = intentos - 1

    limpiar()
    if ganaste:
        pintar(5)
        os.system("aplay applause.wav")
        print "\n\nAcertaste!"
    else:
        pintar(0)
        os.system("aplay punch.wav")
        print "\n\nIntenta otra vez!"

    print "La respuesta era", frase

```

