

INF-253 Lenguajes de Programación

Tarea 1: Python

Profesor: Roberto Díaz

Ayudante de Cátedra: Anastasiia Fedorova

Ayudantes de Tareas: Sebastián Campos, Héctor Larrañaga, Axel Reyes

17 de septiembre de 2020

1. Lore

En el año MMXX, en una galaxia distinta a la nuestra ocurre la historia de una lucha épica entre el totalitario Imperio Javáctico y la democrática Alianza de Planetas Libres de Contexto. La coalición HAAS, en un acto heroico, comenzó a salvar a miles de refugiados pertenecientes a los planetas de la Alianza más afectados en este conflicto. Tú eres parte de esos refugiados, el cual logró escapar del planeta L4-T1NO4M3R-IK justo antes de ser destruido por un aliado del Imperio Javáctico, la Federación de Pythonia.

Mientras viajabas a un lugar seguro junto a cientos de refugiados, ocurrió una catástrofe. Una famosísima sicaria llamada Anne von Fohengramm se infiltró en la nave con la misión de destruirla junto a todos sus tripulantes. Para lograr su objetivo, tomó su forma ASCII y se escabulló dentro del sistema, encriptando todos los tesseractos de información necesarios y dejando la nave inutilizable.

Buscando una solución al problema, descubres en un post de StackOverflow que las llaves capaces de desencriptar los tesseractos tienen formatos aleatorios no relacionados entre sí con 7 capas de protección. Después de una ardua búsqueda sin buenos resultados y al borde de perder toda esperanza, un recurso narrativo llegó en tu ayuda entregándote unos documentos filtrados de la sicaria, donde te señala que las 7 capas tienen sus llaves en los siguientes lenguajes: colores, archivos de directorio, correos, el triángulo de Pascal, fechas y el reconocidísimo lenguaje de datos llamado XJSON. Decidido, comienzas una larga tarea para descubrir las posibles llaves revisando los documentos para poder salvar la nave y tu vida.

2. Reconocedor de lenguajes

En términos generales, se debe realizar un programa en Python que sea capaz de poder reconocer siete lenguajes utilizando expresiones regulares para apoyarse. Para esto, se debe utilizar Python 3 y la librería `re` para las expresiones regulares de la tarea, en caso de que alguna de estas dos condiciones no se cumplan no se revisará la tarea.

Su programa debe recibir un archivo de texto con palabras, las cuales deben ser clasificadas en uno de los lenguajes o rechazarla como una palabra inválida.

3. Lenguajes

A continuación, se listarán los lenguajes que el programa debe lograr reconocer:

3.1. Colores

Los colores en informática se pueden codificar utilizando canales de colores o definiéndolos en sus componentes constituyentes. Este lenguaje estará compuesto por dos formas de codificar colores, RGB Y HSL.

3.1.1. RGB

Codificación que utiliza tres canales de colores para formar un color, estos canales representan la componente roja, verde y azul del color. La forma de codificar estos canales es utilizando notación hexadecimal junto a un signo gato, donde cada canal está compuesto por dos dígitos hexadecimales que pueden tomar un valor desde 00 (0 en decimal) hasta FF (255 en decimal), siendo el primer color de la codificación el #000000 (que representa el color negro) y el último es #FFFFFF (que representa el color blanco).

3.1.2. HSL

Codificación que representa un color en tres componentes los cuales son: su ángulo en la rueda cromática, su porcentaje de saturación y su porcentaje de luminosidad. La forma de codificarlo es escribiéndolo en forma de 'función', donde el primer parámetro es el ángulo, el segundo la saturación y el tercero la luminosidad. En esta codificación, el color `hsl(0, 0%,0%)` representa el color negro y el `hsl(360 , 100% , 100%)` representa el color blanco.

3.1.3. Ejemplos:

- `#C4F3AA` Pertenece al lenguaje
- `#F1F0F3` Pertenece al lenguaje
- `hsl(361 , 55%, 60%)` No pertenece al lenguaje
- `#F30` No pertenece al lenguaje
- `{hsl(4 , 10% , 5%)}` No pertenece al lenguaje
- `hsl(15 , 10% , 12%)` Pertenece al lenguaje

3.2. Archivos en directorios

Se quieren reconocer direcciones a archivos en un sistema operativo. Para que una dirección se considere válida, puede o no partir por un disco, seguido de múltiples o ninguna carpeta y finalmente con un archivo, todo separados por \.

- El disco se indica con una única letra mayúscula seguida de dos puntos por ejemplo `E:\`
- Las carpetas están compuestas por cualquier carácter ASCII que no sean

`\ / : * ? " > < |`

y deben comenzar por un carácter alfanumérico.

- Los archivos están conformados por caracteres alfanuméricos y puntos. Un archivo no puede comenzar con punto.

Además, en caso de no indicarse un disco, se puede comenzar por el directorio *home* `~\` o por el directorio actual `.\` y, como opción adicional, se puede hacer referencia al directorio padre `..\`

3.2.1. Ejemplos

- `E:\Carpeta\archivo.txt` Pertenece al lenguaje
- `.\Desktop\USM\LP\Tarea 1\programa.py` Pertenece al lenguaje
- `..\..\..\Minecraft.exe` Pertenece al lenguaje
- `uwu\ esto no se puede\ .py` No pertenece al lenguaje

3.3. Correos

Un correo electrónico está constituido por una **parte-local** y un **dominio** ambos separados por un `@`.

3.3.1. local-part

Puede contener cualquiera de los siguientes caracteres ASCII

- Letras mayúsculas y minúsculas `A` a la `Z` y `a` a la `z`
- Dígitos del 0 al 9
- Caracteres imprimibles

`! # $ % & ' * + - / = ? ^ _ ' { | } ~`

- Puntos `.` siempre y cuando no sea ni el primer ni el último carácter, además no pueden aparecer dos seguidos.

3.3.2. dominio

Los dominios deben seguir reglas más estrictas. Están conformados por múltiples etiquetas DNS separadas por punto. Cada etiqueta DNS puede estar conformado solo por caracteres alfanuméricos y guiones.

- Letras mayúsculas y minúsculas `A` a la `Z` y `a` a la `z`
- Dígitos del 0 al 9
- Guiones `-`

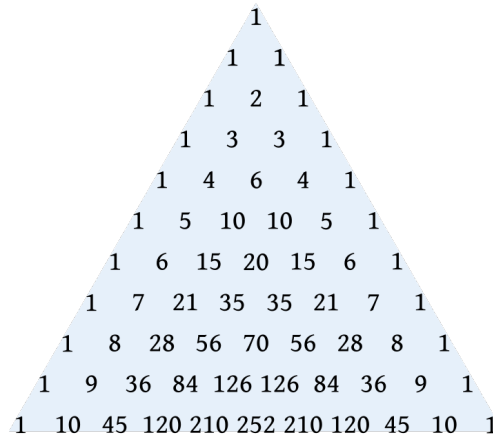
3.3.3. Ejemplos:

- `trivialito@ejemplo.correcto.com` Pertenece al lenguaje
- `un.nombre+y+algunos+tags@uwu-correcto.net` Pertenece al lenguaje
- `ejemplo.malo.com` No pertenece al lenguaje
- `otro@ejemplo@malo.cl` No pertenece al lenguaje
- `aqui_si@aqui_no.malo.com` No pertenece al lenguaje

3.4. Triángulo de Pascal

El Triángulo de Pascal se puede utilizar como una representación de los coeficientes binomiales, donde los niveles del triángulo indican el grado del polinomio aumentando de arriba hacia abajo.

Triángulo de Pascal



Debe determinar si la secuencia pertenece a un nivel del Triángulo de Pascal. La imagen es solo de referencia, el Triángulo de Pascal no posee límite de altura.

3.4.1. Ejemplos:

- {1 4 6 4 1} Pertenece al lenguaje
- {1 8 28 56 70 56 28 8 1} Pertenece al lenguaje
- {1 3 3 6 2 2 1} No pertenece al lenguaje

3.5. Fecha

Las fechas poseen distintos formatos posibles para representarlas. En este caso usaremos el formato **año-mes-día**.

Debe determinar si la secuencia corresponde a una fecha válida, considerando la cantidad de días de cada mes y años bisiestos.

3.5.1. Ejemplos:

- 2020-02-29 Pertenece al lenguaje
- 2019-02-29 No pertenece al lenguaje
- 2020-2-2 Pertenece al lenguaje
- 2020-12-02 Pertenece al lenguaje
- 0001-12-3 Pertenece al lenguaje
- 1-12-3 No pertenece al lenguaje

3.6. XJSON

XJSON es un formato de texto sencillo, derivado de JSON, utilizado para el intercambio de datos. Cada elemento XJSON puede estar compuesto por 0 o más datos de la forma *NombreDeDato* = *Valor* encerrados en paréntesis angulares ('<', '>') y separados por comas. El tipo de valor de los datos pueden ser cuatro: enteros, strings, arreglos de enteros o strings, y XJSON.

Los strings deben estar entre comillas y sólo pueden poseer caracteres alfanuméricos. Los nombres de los datos sólo pueden tener caracteres alfabéticos. Los arreglos se encuentran entre corchetes ('[', ']') y cada elemento se encuentra separado por coma ('(',')')

3.6.1. Ejemplos

- < hola = 1 , joaoaoa = "XDDDDDDDD", tomc = 432, xdson = < holaestoydentro = 1 , yo = "igual">, M = [1,2,3,4]> Pertenece al lenguaje
- < holasoyelhector = 10 , holasoyelaxel = 11, holasoylaana = 12, holaestoymal : 13> No pertenece al lenguaje
- <jeje = ["xd", 3]> Pertenece al lenguaje
- <jeje = [":(", 3]> No pertenece al lenguaje
- <jeje = ["xd", 3, [2, 3]]> No pertenece al lenguaje

4. Input y Output

El programa recibirá un archivo llamado palabras.txt donde cada línea corresponderá a una palabra (secuencia) que puede o no pertenecer a los lenguajes descritos con anterioridad. Debe clasificar cada una de las palabras y mostrar por pantalla su clasificación como se muestra en el ejemplo:

4.1. Ejemplo:

4.1.1. Input:

```
1852-12-12
{1 2 1}
hsl(160, 50%, 40%)
<xd = "xd", xdd = 2>
no pertenezco aqui, no es tu familia
soyuncorreo@correo.com
#EFEXD1
#4F3C1A
ejemplo.malo.com
```

4.2. Output:

```
Fecha
Pascal
HSL
XJSON
No pertenece al lenguaje
```

Correo
No pertenece al lenguaje
RGB
No pertenece al lenguaje

5. Consideraciones adicionales

- En caso de no utilizar Re se evaluará con nota 0.
- Las expresiones regulares deben ser parte del reconocimiento de todos los lenguajes expuestos de forma obligatoria. Sino, no se considerará el reconocimiento de ese lenguaje como implementado.

6. Sobre Entrega

- Se deberá entregar un programa llamado `reconocedor.py`.
- Las funciones implementadas deben ser comentadas de la siguiente forma. **SE HARÁN DESCUENTOS POR CADA FUNCIÓN NO COMENTADA**

```
"""
Nombre de la función
-----
Parametro 1 : Tipo
Parametro 2 : Tipo
Parametro 3 : Tipo
.....
-----
Breve descripción de lo que realiza la función y lo que retorna
"""
```

- Se debe trabajar de forma individual obligatoriamente.
- La entrega debe realizarse en `tar.gz` y debe llevar el nombre: `Tarea1LP_RolAlumno.tar.gz`
- El archivo `README.txt` debe contener nombre y rol del alumno e instrucciones detalladas para la correcta utilización de su programa.
- El no cumplir con las reglas de entrega conllevará un descuento máximo de 30 puntos en su tarea.
- La entrega será vía aula y el plazo máximo de entrega es hasta el **Miércoles 30 de Septiembre a las 23:55 hora aula**.
- Por cada día de atraso se descontarán 20 puntos (10 puntos dentro la primera hora).
- Las copias serán evaluadas con nota 0 y se informarán a las respectivas autoridades.
- Se responderán sólo las consultas realizadas vía Aula.
- Sólo se responderán las consultas hasta 2 días antes de la fecha original de entrega.

7. Calificación

7.1. Entrega

- RGB (10 Puntos)
- HSL (10 Puntos)
- Directorios (10 puntos)
- Correos (10 puntos)
- Triángulo de pascal (15 puntos)
- Fecha (15 puntos)
- XDSON (30 puntos)

7.2. Descuentos

- Sin README (-15 puntos)
- Función sin comentar o no sigue formato (-5 puntos c/u)
- No seguir formato de entrega (-10 puntos)