Python y Expresiones Regulares

Lenguaje **multi-paradigma** de programación creado por Guido van Rossum a finales de los 80.

-Interpretado

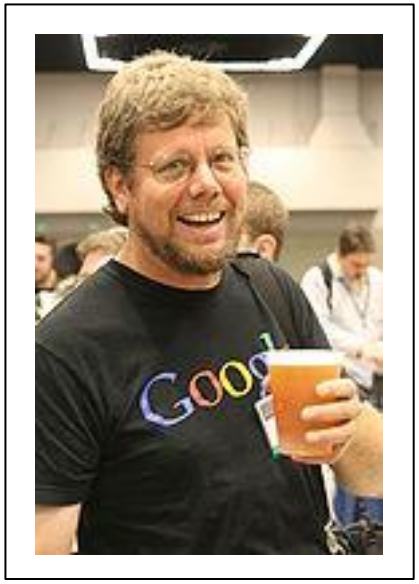
-Multi-plataforma

-Funcionalidades

-Frameworks

-Comunidad

-Investigación



Guido van Rossum en 2006, de https://es.wikipedia.org/wiki/Python

Hola mundo en python

holamundo.py

```
1 print("hola mundo")
```

Versión y Ejecución

- Vamos a usar **python 3.8**

- Ejecución

  python programa.py param1 param2 …

- Verificar versión

  python --version

Tipos de Datos

- **Bool** – True, False
- **Int** – 15 ,8, 0, -1, -5
- **Float** – 1.25, -7.49
- **Str** – "abc de", ‘hola!’, ‘hola’ + ‘!’
- **List** – [1,2,3], [‘a’,1], list((1,)) – l[0], l[1:3], l[::-1]
- **Tuple** – (1,2,3), tuple([1,2]) (listas inmutables)
- **Dict** – {‘a’: 0, ‘b’:1}, dict(a=0,b=1), v[‘a’]
- **Set** – set(...) , unión, intersección, etc.

Módulos

- Cada archivo **.py** define un módulo
  - Estos pueden ser agrupados en paquetes
  - Paquete: directorio que contiene archivo __init__.py
- Para importar
  - **import** m
  - **from** m **import** x, …
- En un módulo pueden haber **funciones**, **clases** y **variables**
- **módulo de expresiones regulares**

  import re

Expresiones Regulares

En python (en el módulo re) una expresión regular se denota como **r'...'**. Por ejemplo **r'.*'**

- **re.search(pattern, string, flags=0)**
  - Busca una ocurrencia del patrón en la string
  - Retorna el *match* o None si el patrón no ocurre en la string
  - En caso de ocurrencia, con group se accede a los grupos

    Ej. m.group(0) # el grupo de la er completa
  - El método groups retorna todos los grupos en la er

Expresiones Regulares

- **re.findall(pattern, string, flags=0)**
  - Retorna la lista de ocurrencias no solapadas del patrón en la string
  - Si hay grupos definidos en la er retorna una lista de tuplas

- **re.sub(pattern, repl, string, count=0, flags=0)**
  - Retorna la string resultado de reemplazar las ocurrencias del patrón por repl

Expresiones Regulares

Metacaracteres

. – cualquier caracter menos fin de línea

***** – 0 o más ocurrencias

{n,m**}** – de n a m ocurrencias

? – 0 o 1 ocurrencias

+ – 1 o más ocurrencias

| – unión

**** – caracter de escape

^, **$** – principio y fin de la entrada

(…**)** – forma grupos a ser extraídos

(?: …**)** – agrupa sin formar grupo

Expresiones Regulares

- **Clases de caracteres**

  […] – cualquiera de los caracteres

  Admite rangos, ej. [1-5]

  [^…] – clase negada (complemento)

- **Abreviaciones de clases**
  - **\d** es un dígito [0-9]
  - **\w** es un caracter alfanumérico [a-zA-Z0-9]
  - **\s** es un espacio, **\b** el comienzo de una palabra
  - Cualquiera de ellos, en mayúscula, refiere al complemento. Ej. **\D** es un no-dígito

Expresiones Regulares

- **Flags**
  - re.I (ignora mayúsculas y minúsculas)
  - re.MULTILINE (cambia el comportamiento de ^ y $)
  - re.DOTALL (. matchea fines de línea)

- **Greedy y non-greedy**
  - ? Indica a los operadores de repetición que abarquen lo menos posible

    *?, +?, ??

Ejemplo de entrada

```
<doc id="841352" title="Nucifraga" nonfiltered="1" processed="1"
dbindex="230002">

El género Nucifraga perteneciente a la familia Corvidae incluye
a dos especies de cascanueces: el cascanueces del Viejo Mundo y
el cascanueces norteamericano o de Clark.




ENDOFARTICLE.
</doc>
```

Ejemplos

- **Eliminar etiquetas de abrir**
  - re.sub(r"<doc[^>]*>", "", texto)
  - re.sub(r"<doc.*?>", "", texto)
- **Obtener palabras (delimitadas por espacios)**
  - re.findall(r"(\S+)", texto)
- **Cantidad de palabras**
  - len(re.findall(r"(\S+)", texto))