

Facultad de ingeniería

**Manual de instrucciones para utilizar expresiones y definiciones regulares en python**

Autores

Fernando José Fiorillo Hermosa

Luis Fontalvo Ramos

Albín Ernesto Núñez Pérez

Luis Ángel Urueta Duarte

Tutor

Darwin Ramiro Mercado Polo

Barranquilla - Atlántico, abril 5 de 2022

**Instalación de python**

Para instalar python debemos escoger cual será nuestro editor de código, en este caso se escogió visual studio code porque es rápido ligero y muy sencillo al utilizar, a continuación, te explicaremos como usar visual studio code para poder usar python.

1. Nos vamos a la página oficial de visual studio code

<https://code.visualstudio.com/>

1. Luego nos aparecerá una ventana principal la cual nos indicara en la pestaña download donde escogeremos nuestro sistema operativo y su características ya sea de x32 o x64.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Al escoger nuestro sistema operativo le damos descargar y después de allí le damos siguiente siguiente siguiente aceptamos términos y ya después abrimos visual studio code.
2. Estando allí vemos una gran ventana luego de allí nos vamos a extensiones en donde se instalaran las extensiones que nos ayudaran a que nuestro código en python corra sin ningún problema.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

**Nota: la pestaña de extensiones se encuentra en el cuarto item a la parte izquierda donde está la lupa.**

1. Buscamos en la extensiones que se llama python y listo empezamos a correr nuestro programa.

**Uso de python para las expresiones regulares**

Nosotros para utilizar python en la evaluación de cadenas para las expresiones regulares se empleo el uso de la biblioteca re.

re: Biblioteca que proporciona operaciones con expresiones regulares como la búsqueda con el search y el fullmatch que nos sirve para retornar un objeto de coincidencia si y solo si toda la cadena coincide con el patrón. De lo contrario, devolverá ninguno.

Ejemplo: Dada la expresión regular 𝑐∗(𝑎|𝑏|𝑎𝑐?|𝑐𝑏|𝑏𝑐?)∗ una de las siguientes cadenas no es aceptada.

a) cabcabab

b) ccbccbba

c) baccbacc

d) baaabcac

e) Ninguna de las anteriores

Nota: Digitar manualmente en el teclado la cadena a evaluar para que el código funcione.

* Opción a: cabcabab
* Ingresamos la expresión y la cadena de manera manual.

Interfaz de usuario gráfica

Descripción generada automáticamente

* Como resultado obtenemos:

Texto

Descripción generada automáticamente

* Opción b: ccbccbba
* Ingresamos la expresión y la cadena de manera manual.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

* Como resultado obtenemos:

Texto

Descripción generada automáticamente

* Opción c: baccbacc
* Ingresamos la expresión y la cadena de manera manual.

Interfaz de usuario gráfica

Descripción generada automáticamente

* Como resultado obtenemos:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Opción d: baaabcac
* Ingresamos la expresión y la cadena de manera manual.

Interfaz de usuario gráfica

Descripción generada automáticamente

* Como resultado obtenemos:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

En conclusión la cadena que no es aceptada es baccbacc para la expresión regular formada por 𝑐∗(𝑎|𝑏|𝑎𝑐?|𝑐𝑏|𝑏𝑐?)∗ .

**Uso de python para las definiciones regulares**

Antes de iniciar con las definiciones regulares debemos resaltar que en la terminal se debe importar mediante el siguiente comando para que permita correr la interfaz.

pip install pysimplegui

Ahora sigamos con las definiciones regulares, donde se empleo el compile que es una función la cual crea un objeto de expresión regular compilando un patrón de expresión regular, objeto que puede ser usado como patrón de coincidencias en las funciones re.match, re.search, etc.

Ejemplo 1:

#Instrucciones:

#1) Un identificador en un lenguaje comienza con una letra mayúscula

#2) Luego, continúa con letras mayúsculas o minúsculas

#3) La longitud debe ser de 5

#4) No puede terminar en las subcadenas CIA o CIO

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Ejemplo 2:

#Instrucciones:

#Para este caso la longitud se limita a minimo 3 y máximo 5 digitos.

#Los 3 primeros símbolos deben ser letras mayúsculas sin incluir la Ñ y la O

#Luego de letras, deben seguir dígitos

#además de hacer distinción entre par e impar, por tanto se descompone la expresión así:

#Cadenas de longitud 6 pares

RE6par = re.compile('[A-N,P-Z][A-N,P-Z][A-N,P-Z][0-9][0-9][0,2,4,6,8]')

#Cadenas de longitud 7 pares

RE7par = re.compile('[A-N,P-Z][A-N,P-Z][A-N,P-Z][0-9][0-9][0,2,4,6,8]')

#Cadenas de longitud 8 pares

RE8par=re.compile('[A-N,P-Z][A-N,P-Z][A-N,P-Z][0-9][0-9][09][0,2,4,6,8]')

#Cadenas de longitud 6 impares

RE6impar = re.compile('[A-N,P-Z][A-N,P-Z][A-N,P-Z][0-9][0-9][1,3,5,7,9]')

#Cadenas de longitud 7 impares

RE7impar=re.compile('[A-N,P-Z][A-N,P-Z][A-N,P-Z][0-9][0-9][0-9][1,3,5,7,9]')

#Cadenas de longitud 8 impares

RE8impar=re.compile('[A-N,P-Z][A-N,P-Z][A-N,P-Z][0-9][0-9][0-9][0-9][1,3,5,7,9]')

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente