

INF-253 Lenguajes de Programación

Tarea 1: Python

Profesor: José Luis Martí – Esteban Daines
Ayudante Cátedras: Juan Pablo Escalona
Ayudante Tareas: Javier Echiburú

3 de Agosto de 2016

1. SuerTex

SuerTex es una nueva idea para un editor de texto vía código. Se le ha pedido a los alumnos de Lenguajes de Programación que programen dicho editor.

El alumno aplicará conceptos de **Expresiones regulares** para realizar dicho programa, dado que deberá analizar gramaticalmente el código y ejecutar lo que se solicita debidamente.

2. Requerimientos

- El alumno debe utilizar Python 2.7 para el desarrollo de esta tarea.
- El programa debe utilizar expresiones regulares para el parseo del código.
- El archivo de texto a utilizar se llamará suertex.txt.
- El programa debe presentar el output en un nuevo archivo llamado output.html, el cual se ejecutara en un navegador web.
- El programa no debe crear archivos temporales.
- Debido a que el archivo output esta en html es necesario que el alumno domine algunos comandos de este lenguaje.

3. ¿Que debe poder hacer SuerTex?

SuerTex se compone de funciones y comandos, cabe destacar que los caracteres ‘\’, ‘{’ y ‘}’ solo se permite usarlos como parte del código, en caso de existir de manera singular dentro del texto (que no formen parte de un comando) debe arrojar error.

3.1. Funciones

Las funciones deben ir al inicio del texto, antes que cualquier otra cosa, si la función existe al inicio del código significa que debe ejecutarse, en el caso que no exista se ignora. Si se escribe en otra parte del código debe arrojar error.

- `\separamiles{}`: Cambia el formato de todos los números separándolos con un punto por miles.

Ejemplo: $1003240 = 1.003.240$

PD: Notar que si un numero pertenece a un nombre, este no debe ser cambiado.

- `\ofecha{}`: Cambia el formato de todas las fechas presentes a un único formato (dd/mm/aaaa).

Ejemplo: $24-07-2015 = 24/07/2015$

PD: Notar que la fecha siempre estará ordenada de esa manera, pero la separación puede ser un carácter distinto, sin embargo el separador no puede ser una letra, numero o espacio.

3.2. Comandos

Los comandos pueden ir en cualquier parte del código y pueden estar anidados, es decir, dentro de *Texto* puede haber mas comandos.

- `\fn{Texto}`: Debe colocar *Texto* en negrita.
- `\fc{Texto}`: Debe colocar *Texto* en itálica (cursiva).
- `\nproj{Texto}`: Debe escribir *Texto* como nombre del proyecto, en este caso sera el nombre de la pestaña al abrirlo por navegador. Este comando es obligatorio y debe venir inmediatamente después de las funciones, en caso de estar escrito mas de una vez o no existir debe arrojar error.
- `\titulo{Texto}`: Debe escribir *Texto* como título.
- `\inicio{Accion}`: Sirve para indicar el inicio de una acción, una acción puede ser lista_punteada o lista_enumerada, en caso de tener algo distinto a las 2 acciones mencionadas debe arrojar error.
- `\fin{Accion}`: Sirve para indicar el fin de una acción, una acción puede ser lista_punteada o lista_enumerada, en caso de tener algo distinto a las 2 acciones mencionadas debe arrojar error. En caso de que la acción no este cerrada debe arrojar error.
- `\item{Texto}`: Se utiliza entre `\inicio{Accion}` y `\fin{Accion}`, sirve para indicar donde comienza el nuevo ítem, dependiendo de la acción este será punteado o enumerado.

4. Ejemplo

Para visualizar el archivo HTML deben abrirlo en el navegador.

Consideraciones útiles:

- Es posible anidar comandos.
- Solo se ejecuta la función `\separamiles{}` cuando el numero no pertenece a un string con caracteres (ver caso del usuario lala37284 en el ejemplo).
- En este ejemplo no se modifica el formato de las fechas debido a que no se ejecuta la función `\ofecha{}` al inicio del código.
- El programa debe iniciar analizando gramaticalmente del código, si este llega a ser incorrecto se arroja error y no se crea el archivo HTML.
- En caso de error, debe ser informado por consola.

4.1. SuerTex.txt

```
\separamiles{}
\nproy{Nombre Proyecto}
\titulo{A que temer}

Todo hombre sabio teme \fn{tres} cosas:
\inicio{lista_enumerada}
\item{La tormenta en el mar.}
\item{La noche sin luna.}
\item{La ira de un hombre amable.}
\fin{lista_enumerada}

\titulo{Como saber quien eres}
Si quieres saber quien eres, camina hasta que no haya nadie que sepa tu nombre.

\titulo{Taborlin el Grande}
Pero Taborlin conocia el nombre de todas las cosas y todas las cosas estaban a sus ordenes.
Le dijo a la piedra: \fc{Rompete!}, y la piedra se rompio.

\fn{\fc{En resumen} Patrick Rothfuss} ha escrito 2 libros de la cronica de asesino de reyes:
\inicio{lista_punteada}
\item{El nombre del viento. 27-03-2007}
\item{El temor de un hombre sabio. 01,03,2011}
\fin{lista_punteada}

Existen 2938194 posibilidades de que el usuario \fc{lala37284} exista en un juego.
```

4.2. Output.html

```
<!DOCTYPE HTML>
<head>
  <title>Nombre Proyecto</title>
</head>
<body>
  <h1>A que temer</h1>

  <p>Todo hombre sabio teme <strong>tres</strong> cosas:</p>
  <ol>
    <li>La tormenta en el mar.</li>
    <li>La noche sin luna.</li>
    <li>La ira de un hombre amable.</li>
  </ol>

  <h1>Como saber quien eres</h1>
  <p>Si quieres saber quien eres, camina hasta que no haya nadie que sepa tu nombre.</p>

  <h1>Taborlin el Grande</h1>
  <p>Pero Taborlin conocia el nombre de todas las cosas y todas las cosas estaban a sus ordenes.
    Le dijo a la piedra: <em>Rompete!</em>, y la piedra se rompio.</p>
```

```

<p><strong><em>En resumen</em> Patrick Rothfuss</strong> ha escrito 2 libros de la
    cronica de asesino de reyes:</p>
<ul>
    <li>El nombre del viento. 27-03-2007</li>
    <li>El temor de un hombre sabio. 01,03,2011</li>
</ul>

<p>Existen 2.938.194 posibilidades de que el usuario <em>lala37284</em> exista en un
    juego.</p>
</body>

```

5. Archivos a entregar

- suertex.py
- readme.txt

6. Sobre Entrega

- Los comentarios deben ir al inicio de cada función y deben llevar el siguiente formato:

```

"""
funcion(parametros)
    Descripcion breve de que hace.
    Que retorna.
"""

```

- La revisión se efectuará sobre el sistema operativo CentOS presente en los computadores del laboratorio de la Universidad.
- La entrega debe realizarse en tarball (tar.gz) y debe llevar el nombre: **Tarea1LP_RolIntegrante-1_RolIntegrante-2.tar.gz**
- El archivo README.txt debe contener nombre y rol de los integrantes del grupo e instrucciones detalladas para la correcta utilización de su programa.
- El no cumplir con las reglas de entrega conllevará **-30 puntos** en su tarea.
- La entrega será vía moodle y el plazo máximo de entrega es hasta el **lunes 22 de agosto a las 23:55 hora moodle**.
- Por cada día de atraso se descontarán 30 puntos.
- Las copias serán evaluadas con nota 0 y se informarán a las respectivas autoridades.

7. Calificación

Todos comienzan con nota máxima y se les irá descontando por cada punto omitido (el puntaje que aparece al lado es el máximo de puntos que se les descontara en caso de incumplimiento):

- Código no ordenado (-5 puntos)
- Código no comentado (-6 puntos)
- Por cada comando que no realice su cometido (-7 puntos)
- Por cada función que no realice su cometido (-10 puntos)
- No se revisa la gramática del código (-20 puntos)
- No respetar Reglas de entrega (-30 puntos)
- No utiliza expresiones regulares (-100 puntos)

En caso de existir nota negativa esta será reemplazada por un 0.