

# Find\_the\_file

February 27, 2023

## 0.1 Instalando Git

En Windows, se puede descargar [Git for Windows](#) desde [este enlace](#), ejecutar el instalador y seguir las instrucciones. Viene con una terminal y una interfaz gráfica.

## 0.2 Buscando la clase

- Abrimos la terminal de GitBash
- Bajamos el repo con `git clone`:

```
git clone git@github.com:fcolavecchia/standard_library_python.git
```

Vamos a usar herramientas de la biblioteca standard de Python para poder buscar la clase. [Busquemos](#).

Importamos el módulo `os`.

```
[ ]: import os

[ ]: os.getcwd() #Current working directory

[ ]: standard_library_data_dir = '/Users/flavioc/Codes/Languages/Python/
    ↳standard_library_python/data'

os.chdir(standard_library_data_dir)
print(os.getcwd())
```

Primer intento, mostramos todos los directorios y los archivos:

```
[ ]: for root, dirs, files in os.walk('./.'):
    print(root,dirs,files)
```

Segundo intento, imprimo sólo los archivos:

```
[ ]: for root, dirs, files in os.walk('./.'):
    for file in files:
        file = root + "/" + file
        print(file)
```

Aparecen mal concatenados los directorios y los archivos, falta el / correspondiente. Para eso usamos `os.path.join`.

```
[ ]: for root, dirs, files in os.walk('.'):
      for file in files:
          file = os.path.join(root, file)
          print(file)
```

Ahora podemos definir entonces la lista de archivos.

```
[ ]: def file_list(directory):
      l = []
      for root, dirs, files in os.walk('.'):
          for file in files:
              file = os.path.join(root, file)
              l.append(file)
      return l
```

```
[ ]: files = file_list('.')
      print(len(files))
```

### 0.2.1 Módulo glob

El módulo `glob` encuentra nombres de archivos (o directorios) utilizando patrones similares a los de la consola. La función más utilizada es `glob.glob()` Veamos algunos ejemplos de uso:

```
[ ]: import glob
```

```
[ ]: help(glob)
```

```
[ ]: nb_clase7= glob.glob('07*.ipynb')
      nb_clase7
```

mmm, not recursive. Try recursive:

```
[ ]: nb_clase7= glob.glob('07*.ipynb', recursive=True)
      nb_clase7
```

Evidentemente el archivo no está con ese nombre. Entonces tendríamos que ver la manera de buscar algo de un archivo `.ipynb` en el directorio.

Los notebooks son archivos de texto con una estructura **JSON**, para más detalles, veamos [el formato JSON del notebook](#).

```
{
  "metadata" : {
    "kernel_info": {
      # if kernel_info is defined, its name field is required.
      "name" : "the name of the kernel"
    },
    "language_info": {
      # if language_info is defined, its name field is required.
      "name" : "the programming language of the kernel",
```

```

        "version": "the version of the language",
        "codemirror_mode": "The name of the codemirror mode to use [optional]"
    }
},
"nbformat": 4,
"nbformat_minor": 0,
"cells" : [
    # list of cell dictionaries, see below
],
}

```

Ok, entonces ahora tenemos que buscar algún archivo que contenga, por ejemplo, un *string* típico de estos archivos, por ejemplo `nbformat`.

```
[ ]: def find_text_in_file(text,file_name):
    with open(file_name, 'r') as fin:
        file_content = fin.read()
        return (text in file_content)
```

```
[ ]: clave = "nbformat"
```

```
[ ]: for f in files:
    if (find_text_in_file(clave,f)):
        file_found = f
        break

file_found
```

## 0.2.2 Módulo `shutil`

Nice.

```
nombre_clase = "07_modulos_biblioteca_bis.ipynb"
```

- Copiarlo al directorio de trabajo con `shutil`.

Python `dirname` and `basename`.

```
[ ]: import shutil
```

```
[ ]: help(shutil.copy)
```

```
[ ]: file_clase = "07_modulos_biblioteca_bis.ipynb"
shutil.copy(file_found,file_clase)
```

## 0.2.3 Ejercicio

Además de la clase, puede también que se hayan perdido algunos cuentos de Borges en el directorio. Sus objetivos son - Encontrar alguno de esos cuentos. - Leerlo, si no tuvo la chance alguna vez de hacerlo. - Reportar en el repositorio de GitHub que el nombre del cuento, el personaje principal, y cuál es su opinión sobre el mismo, a través de un *issue*. Para ello tendría que crearse una cuenta

en GitHub, ir al buscar el repositorio `fcollavecchia/standard_library_python.git`, seleccionar **Issues** y luego crear un nuevo issue con el botón **New issue**.

```
[ ]: borges = "Borges"

for f in files:
    if (find_text_in_file(borges,f)):
        file_found = f
        print(file_found)
```

## Misc

```
[ ]: import re
```

```
[ ]: busca = "[gat].*"
texto = "gato, gata, toga, perro"

for word in texto.split(','):
    m= re.search(busca, word)
    print('Para la línea:', word)
    if m is None:
        print('    No encontré nada!')
    else:
        print('    Encontré :', m.string)
```

```
[ ]: pol = "41x^4+3x^3-2222x+1"
busca = "(-?\d+)[x^|x]?(\d?)"
m = re.compile(busca)
q = m.findall(pol)
q
```

```
[ ]: s = re.findall(busca,pol)

s
```

Match los monomios por separado:

```
[ ]: otrobusca = "([+-]?[~--]+)"
t = re.findall(otrobusca,pol)
t
```

Match cada monomio

```
[ ]: mon = "([-+]?([0-9]*)?(x\^[0-9]+)?)+)"

for m in t :
    found = re.findall(mon,m)
    print(found)
```