**APRENDIENDO POR EXPERIENCIA**

import sys

argumentos = sys.argv[1:]

sys.argv[1:] : devuelve los arguetsmn qoue le pasemos al programa al momento de ejecutarlo

-----------------------------------

import getopt

opciones , argumentos = detopt.getopt(sys.arg[1:] ,

----------------------------

import time

tiempo = time.time()

#imprime en formato (1592361315.4571266) son los segundos transcurridos desde el 1 de enero de 1970 a las 00:00:00

tiempo2 = time.ctime(tiempo)

#convierte en formato legible de humano (Tue Jun 16 22:35:15 2020)

import datetime

tiempo = datetime.datetime.now()

#imprime en este formato : 2020-06-16 22:59:50.131098

#se puede restar entre dos datetime

hora = tiempo.hour #guarda solo la hora como numero

minuto = tiempo.minute

segundo = tiempo.second

anho = tiempo.year

DIFERENCIAS ENTRE PIP Y PIP3

for i in pip pip3 python python3;do type $i;done

“esto nos muestra las direcciones de cada cosa

**CONECTAR PYTHON CON HTML**

**-**string.Template(‘cadena a sustituir’) : a esta funcion se le pasa un string en este caso un script .html

-en el **html** tenemos que poner el **identificador** a sustituir precedido por ‘$’

$identificador

-asi tenemos que declarar la palabra a sustituir

-

import os

import sys

from string import Template #esto te permite **obtener** un objeto donde sustituiremos

stream = open(‘path\_html’)

src = Template(stream.read())

nuevo = src.sustitute(identificador = valor\_nuevo , identificador2 = valor\_nuevo2)

diccionario = {‘identificador1’ : valor1 , ‘identificador2’ : valor2}

nuevo2 = src.sustitute(diccionario)

**MODULO REQUESTS**

import requests

r = requests.get(‘<https://www.youtube.com/>’) #asi haces una peticiony toda la imformacion de la peticion esta guardada en la variable ‘r’

print(r.encoding) #nos devuelve el tipo de codificacion de la respuesta

print(r.status\_code) #nos devuelve el estatus de l peticion, si fue exitosa (200,404)

print(r.headers) # devuelve un diccionario de encabezados de respuesta

print(r.is\_redirect) #devuelve True o False si la peticion fue redireccionada

print(r.elapsed) #dice cuanto tardo la respuesta en llegar desde

-para acceder al contenido de la respuesta

print(r.text) #esto devuelve todo el codigo HTML de la pagina solitadaen forma de string

pritn(r.content) #te devuelve el contenido en forma de bytes

pritn(r.json) #te devuelve el contenido en formato de Json

-optener la respuesta cruda del servidor

r2 = requests.get('<https://www.youtube.com/>’, stream =True) #asi optendras la respuesta en crudo del servidor

print(r2.raw) #mostrara el contenido de la respuesta en bruto

-**ve ejemplo /programas\_4/descargar\_imagen.py**

PASAR PARAMETROS EN LA URL

query = {‘q’ : ‘Forest’ , ‘order’ : ‘popular’ , ‘min\_width’ : ‘800’, ‘min\_heigth’ : ‘600’} #estos son los parametros que pasaremos a la url

req = requests.get(‘<https://pixabay.com/en/photos/>’ , params = query)

req.url #aqui vemos la url

HACER UN POST REQUESTS

-esto sirve cuando queremos manda cualquier tipo de dato el servidor

**QUE ES DJANGO**

-es un framework web

-sirve para desarrollo web

-realizar aplicaciones de cualquier complejidad

**QUE ES UNA API?**

**-**API= Aplication programming interface

**-**es una interfaz entre dos sistemas

-te permite comunciar con otras aplicaciones

-pueden ser publicas o privadas

-necesitas una autenticacion

-el servidor te devolvera un ‘token’

-token es un objeto con todos los datos de la autenticcacion

-se pueden vencer los ‘token’

-las consultas a la api tiene un identiifcador unico (URI)

-al hacer una peticion a un servidor a travez de una api el servidor te responde con codigos

-2xx peticion exitosa

-3xx redireccion

-4xx peticion fallida

-5xx error ene l servidor

PRACTICAS PARA DESARROLLAR UNA **API REST**

-hateoas: la api se autodescribe

-cada recurso tiene informacion del recurso siguiente o de la cantidad de recurso que hay

-seguridad : si la api es privada tienes que proteger los datos

-testear : probar para que no falle y va de la mano con la documentacion

-documentacion : documentar el funcionamiento de la api

TIPOS DE APIS

-pueden ser locales o remotas

-locales se corren en el mismo entorno

-remotas pueden usar servicios web

-pueden usar REST

QUE ES REST

**-**RESP = Representational State Transfer

-es una api muy utilizada atualmente

**FRAMEWORK FLASK**

esto te sirve para poder hacer aplicaciones

sudo pip3 install Flask

sudo pip3 install Flask-SocketIO #esto te sirve para poder inviar datos con flask