```
import requests
import md_to_html
import webbrowser

#Prends un id en paramètre et récupère toutes les infos du pokémon dans un dictionnaire en json :



def get_dataset(id:int)->dict:
    response = requests.get("https://pokeapi.co/api/v2/pokemon/" + str(id))
    data = response.json()
    nom_response = requests.get("https://pokeapi.co/api/v2/pokemon-species/" + str(id))
    donnees = nom_response.json()

    for nom_pokemon in donnees["names"]:
        if nom_pokemon ["language"]["name"] == "fr":
            data ["nom_pokemon_traduit_fr"] = nom_pokemon["name"]
    return data


# Le but de cette fonction de tri est de trier dans l'ordre croissant la stat de poid, mais de conserver la liaison avec le poid pour l'id et la vitesse
def tri(L, L2, L3):
    N = len(L)
    for n in range(1, N):
        cle = L[n]
        cle2 = L2[n]
        cle3 = L3[n]
        j = n - 1
        while j >= 0 and L[j] > cle:
            L[j + 1] = L[j]   # Tri du poid dans l'ordre croissant
            L2[j + 1] = L2[j] # Même echange pour l'id que pour le poid
            L3[j +1] = L3[j]  # Même echange pour la vitesse que pour le poid
            j = j - 1
        L[j + 1] = cle
        L2[j + 1] = cle2
        L3[j + 1] = cle3

liste_poid=[]
liste_id=[]
liste_vitesse=[]

def poke_stat(data: dict, filename:str)->None:
    dico={}
    liste = ""
    dico["weight"]=data["weight"]
    dico["name"] = data.get("nom_pokemon_traduit_fr", data["name"])
    #Pour les éléments dans types , la liste prend la valeur name qui est dans type qui est dans le dictionnaire data
    #Et on rajoute une espace pour les cas où y'a 2types sur 1 pokémon
    for t in data["types"]:
        liste +=(t["type"]["name"])+("  ")
    dico["stats"]=data["stats"]

    statisques = data["stats"]
    stat_speed = "speed"
    with open(filename, "w") as f:
        for stats in statisques:
            if stats["stat"]["name"] == stat_speed:
                dico[stat_speed] = stats["base_stat"]
    liste_poid.append(dico["weight"]/10)
    liste_vitesse.append(dico["speed"])
```

```
def fiche_pokemon(id: int)->None:
    poke_stat(get_dataset(id), "Fiche_pokemon.md")

def compute_statistics(liste_poid_pokemon, liste_id_pokemon, liste_vitesse_pokemon):
    global moyenne_poid_leger, moyenne_vitesse_leger
    global moyenne_poid_moyen, moyenne_vitesse_moyen
    global moyenne_poid_lourd, moyenne_vitesse_lourd
    num_pokemon = 1
    pokemon_leger_poid = []
    pokemon_leger_id = []
    pokemon_leger_vitesse = []
    pokemon_moyen_poid = []
    pokemon_moyen_id = []
    pokemon_moyen_vitesse = []
    pokemon_lourd_poid = []
    pokemon_lourd_id = []
    pokemon_lourd_vitesse = []

    for i in range(151):
        fiche_pokemon(num_pokemon)
        liste_id.append(num_pokemon)
        num_pokemon+=1

    # Tri des poids, id et vitesses des pokemons --> Le tri de poids se fait par ordre croissant, les id et vitesses suivent
juste les mêmes échanges que le poids
    tri(liste_poid_pokemon,liste_id_pokemon, liste_vitesse_pokemon)

    # Remplace les listes vides avec l'ajout des poids, id et vitesses pour les pokemon de poids leger :
    for index_leger in range(51):
        pokemon_leger_poid.append(liste_poid_pokemon[index_leger])
        pokemon_leger_id.append(liste_id_pokemon[index_leger])
        pokemon_leger_vitesse.append(liste_vitesse_pokemon[index_leger])

    # Retire la partie des pokemons leger de la liste initiale --> il ne restera plus que les plus lourds et les poids moyen :
    liste_poid_pokemon = liste_poid_pokemon[51:]    # On ne commence qu'a partir de 51 pour couper la liste
    liste_id_pokemon = liste_id_pokemon[51:]
    liste_vitesse_pokemon = liste_vitesse_pokemon[51:]

    # Remplace les listes vides avec l'ajout des poids, id et vitesses pour les pokemon de poids moyens :
    for index_moyen in range(50):
        pokemon_moyen_poid.append(liste_poid_pokemon[index_moyen])
        pokemon_moyen_id.append(liste_id_pokemon[index_moyen])
        pokemon_moyen_vitesse.append(liste_vitesse_pokemon[index_moyen])

    # Retire la partie des pokemons moyen de la liste initiale --> il ne restera plus que les plus lourds :
    liste_poid_pokemon = liste_poid_pokemon[51:]
    liste_id_pokemon = liste_id_pokemon[51:]
    liste_vitesse_pokemon = liste_vitesse_pokemon[51:]

    # Completion des listes pour les pokemons les plus lourds :
    pokemon_lourd_poid = liste_poid_pokemon
    pokemon_lourd_id = liste_id_pokemon
    pokemon_lourd_vitesse = liste_vitesse_pokemon

    # Création des variables de moyennes de poids des pokemons :
    moyenne_poid_leger = round ((sum(pokemon_leger_poid)/len(pokemon_leger_poid)), ndigits=2)
    moyenne_poid_moyen = round ((sum(pokemon_moyen_poid)/len(pokemon_moyen_poid)), ndigits=2)
    moyenne_poid_lourd = round ((sum(pokemon_lourd_poid)/len(pokemon_lourd_poid)), ndigits=2)

    # Création des variables de moyennes de vitesses des pokemons :
    moyenne_vitesse_leger = round((sum(pokemon_leger_vitesse)/len(pokemon_leger_vitesse)), ndigits=2)
```

```
    moyenne_vitesse_moyen = round((sum(pokemon_moyen_vitesse)/len(pokemon_moyen_vitesse)), ndigits=2)
    moyenne_vitesse_lourd = round((sum(pokemon_lourd_vitesse)/len(pokemon_lourd_vitesse)), ndigits=2)


compute_statistics(liste_poid,liste_id, liste_vitesse)


def dataset_to_md(dataset: dict, filename: str)-> None:
    global moyenne_poid_leger, moyenne_vitesse_leger
    global moyenne_poid_moyen, moyenne_vitesse_moyen
    global moyenne_poid_lourd, moyenne_vitesse_lourd
    #Pour les éléments dans types , la liste prend la valeur name qui est dans type qui est dans le dictionnaire data
    #Et on rajoute une espace pour les cas où y'a 2types sur 1 pokémon
    stat_speed = "speed"
    with open(filename, "w") as f:
        f.write("# Problematique : les pokemons les plus petits sont-ils les plus rapides ? \n")
        f.write("# I- Les petits pokemons \n")
        f.write("## Poids moyen : " + str(moyenne_poid_leger) + "kg \n")
        f.write("## Vitesse moyenne : " + str(moyenne_vitesse_leger) +  "\n")
        f.write("# II- Les pokemons moyens \n")
        f.write("## Poids moyens : " + str(moyenne_poid_moyen) + "kg \n" )
        f.write("## Vitesse moyenne : " + str(moyenne_vitesse_moyen) + "\n")
        f.write("# III- Les grands pokemons : \n")
        f.write("## Poids moyen : " + str(moyenne_poid_lourd) + "kg \n")
        f.write("## Vitesse moyenne : " + str(moyenne_vitesse_lourd) + "\n")
        f.write("# Conclusion : les pokemons les plus grands sont les plus rapides ! \n")


def infos_locales()->None:
    dataset_to_md(get_dataset(1), "Fiche_infos_locales.md")
    md_to_html.convert("Fiche_infos_locales.md","Fiche_infos_locales.html")
    webbrowser.open("Fiche_infos_locales.html")
    print("Tout s'est bien passé")

infos_locales()
```