



Repaso Sábado 10 - Septiembre

El profesor Oak de Pueblo Paleta está trabajando en ciertas investigaciones clasificadas con datos de algunos pokemones, es por eso que te pide ayuda para crear un programa que le permita obtener datos valiosos.

Tu tarea como developer será la de crear un programa que, según la opción elegida haga las tareas debajo mencionadas.

Te compartimos el dataset sobre el cual quiere trabajar:

```
pokemones = [  
  {  
    "id": 1,  
    "nombre": "bulbasaur",  
    "tipo": ["planta"],  
    "evoluciones": ["ivysaur", "venusaur"],  
    "poder": 4,  
    "fortaleza": ["agua"],  
    "debilidad": ["fuego"]  
  },  
  {  
    "id": 4,  
    "nombre": "charmander",  
    "tipo": ["fuego"],  
    "evoluciones": ["charmeleon", "charizard"],  
    "poder": 5,  
    "fortaleza": ["planta"],  
    "debilidad": ["agua"]  
  },  
  {  
    "id": 5,  
    "nombre": "charmeleon",
```

```
        "tipo": ["fuego"],
        "evoluciones": ["charizard"],
        "poder": 9,
        "fortaleza":["planta"],
        "debilidad":["agua"]
    },
    {
        "id": 6,
        "nombre": "charizard",
        "tipo": ["fuego", "volador"],
        "evoluciones": ["mega-charizard x", "mega-charizard y",
"charizard gigamax"],
        "poder": 18,
        "fortaleza":["planta"],
        "debilidad":["agua"]
    },
    {
        "id": 9,
        "nombre": "blastoise",
        "tipo": ["agua"],
        "evoluciones": [],
        "poder": 18,
        "fortaleza":["fuego", "piedra"],
        "debilidad":["planta", "electrico"]
    },
    {
        "id": 172,
        "nombre": "pichu",
        "tipo": ["electrico"],
        "evoluciones": ["pikachu", "raichu", "raichu de alola"],
        "poder": 3,
        "fortaleza":["agua"],
        "debilidad":["planta", "piedra"]
    },
    {
        "id": 25,
        "nombre": "pikachu",
        "tipo": ["electrico"],
        "evoluciones": ["raichu", "raichu de alola"],
        "poder": 5,
```

```
    "fortaleza":["agua", "electrico"],
    "debilidad":["planta", "piedra"]
  },
  {
    "id": 26,
    "nombre": "raichu",
    "tipo": ["electrico"],
    "evoluciones": [],
    "poder": 20,
    "fortaleza":["agua", "electrico"],
    "debilidad":["planta", "piedra"]
  },
  {
    "id": 94,
    "nombre": "gengar",
    "tipo": ["fantasma", "veneno"],
    "evoluciones": [],
    "poder": 30,
    "fortaleza":["psiquico", "volador"],
    "debilidad":["sinistro"]
  },
  {
    "id": 144,
    "nombre": "articuno",
    "tipo": ["hielo", "volador"],
    "evoluciones": [],
    "poder": 30,
    "fortaleza":["volador", "planta"],
    "debilidad":["fuego", "agua", "acero"]
  },
  {
    "id": 145,
    "nombre": "zapdos",
    "tipo": ["electrico", "volador"],
    "evoluciones": [],
    "poder": 30,
    "fortaleza":["agua"],
    "debilidad":["planta", "piedra"]
  },
  {
```

```
    "id": 146,
    "nombre": "moltres",
    "tipo": ["fuego", "volador"],
    "evoluciones": [],
    "poder": 30,
    "fortaleza": ["planta", "electrico"],
    "debilidad": ["agua", "piedra"]
  },
  {
    "id": 150,
    "nombre": "mewtwo",
    "tipo": ["psiquico"],
    "evoluciones": ["mega-mewtwo x", "mega-mewtwo y"],
    "poder": 25,
    "fortaleza": ["agua", "planta", "fuego", "electrico"],
    "debilidad": ["fantasma", "oscuridad"]
  },
  {
    "id": 151,
    "nombre": "mew",
    "tipo": ["psiquico"],
    "evoluciones": [],
    "poder": 24,
    "fortaleza": ["agua", "planta", "fuego", "electrico"],
    "debilidad": ["fantasma", "oscuridad"]
  },
  {
    "id": 448,
    "nombre": "lucario",
    "tipo": ["lucha", "acero"],
    "evoluciones": ["mega lucario"],
    "poder": 24,
    "fortaleza": ["piedra", "agua", "hielo", "planta"],
    "debilidad": ["electrico", "agua", "fuego", "psiquico",
"volador"]
  }
]
```

Nivel 1 - [En clase de repaso]

- 01 - Imprimir nombres de pokemones
- 02 - Imprimir pokemones que tenga un ID par
- 03 - Imprimir pokemones que tenga un ID múltiplo de 25
- 04 - Imprimir nombre de pokemones con su ID de prefijo.
- 05 - Imprimir los pokemones con más poder y cuánto poder tienen (misma fuerza)
- 06 - Imprimir los pokemones con menos poder y cuánto poder tienen (misma fuerza)

Nivel 2 - [En clase de repaso]

- 07 - Imprimir el promedio de poder de pokemones que entre sus tipos tenga 'psíquico'
- 08 - Imprimir el promedio de poder de pokemones que entre sus tipos tenga 'fuego'
- 09 - Imprimir el promedio de poder de pokemones que entre sus tipos tenga 'eléctrico'

Nivel 3 - [En clase de repaso]

- 10 - Imprimir pokemones que posean más de un tipo
- 11 - Imprimir pokemones que posean más de un tipo y su cantidad
- 12 - Imprimir pokemones que posean más de una evolución
- 13 - Imprimir pokemones que posean más de una evolución y su cantidad
- 14 - Imprimir pokemones que posean más de una fortaleza
- 15 - Imprimir pokemones que posean más de una fortaleza y su cantidad
- 16 - Imprimir pokemones que posean más de una debilidad
- 17 - Imprimir pokemones que posean más de una debilidad y su cantidad

Te dejamos algunos tips para que te guies:

01.1

Crear la función "obtener_nombre_pokemon" la cual recibirá por parámetro un diccionario que representará al pokemon, la función deberá obtener el nombre y retornarlo como un string.

01.2

Crear la función "imprimir_pokemones" la cual recibirá por parámetro una lista de pokemones e imprimirá sus nombres. Reutilizar 'obtener_nombre_pokemon'

02.1

Crear la función "tiene_id_par" la cual recibirá por parámetro un diccionario que representará al pokémon y verificará que su id sea par, en caso de que sea par retornará True, caso contrario retornará False.

02.2

Crear la función "obtener_id_pokemon" la cual recibirá por parámetro un diccionario que representará al pokemon, la función deberá obtener el id y retornarlo como un string.

02.3

Crear la función "pokedex_imprimir_pokemon_id_par" la cual recibirá por parámetro la lista de pokemones y deberá imprimir solo los que cumplan con la condición de tener un ID par. Reutilizar las funciones:

'tiene_id_par', 'obtener_nombre_pokemon'

03.1

Crear la función "id_multiplo_25" la cual recibe por parámetro un diccionario que representará al pokémon y verificará que su id múltiplo de 25, en caso de que lo sea retornará True, caso contrario retornará False.

03.2

Crear la función "pokedex_imprimir_pokemon_id_mul_25" la cual recibirá por parámetro la lista de pokemones y deberá imprimir solo los que cumplan con la condición de tener un ID múltiplo de 25. Reutilizar las funciones:

'id_multiplo_25', 'obtener_nombre_pokemon'

04.1

Crear la función "nombre_format_pokemon" la cual recibirá por parámetro un diccionario que representará al pokemon, la función deberá obtener el nombre y su número y retornarlo como un string formateado respetando el estilo: #006 - charizard

Reutilizar las funciones:

'obtener_id_pokemon', 'obtener_nombre_pokemon'

04.2

Crear la función "pokedex_imprimir_nombres_poke_fmt" la cual recibirá una lista de pokemones

y deberá imprimirlos formateados respetando el estilo: #006 - charizard

Reutilizar las funciones:

"nombre_format_pokemon"

05.1

Crear la función "calcular_max_dato" la cual recibirá por parámetro:

La lista de pokemones

Un string que representará el máximo

Un string que representará el dato/key a calcular

La función retornará el valor máximo de la key calculada.

05.2

Crear la función "obtener_lista_pokemones" la cual recibirá por parámetro:

- La lista de pokemones

- La key la cual deberá evaluar cada pokémon que coincida su valor

- El valor el cual debe evaluar que sea igual.

La función devolverá una lista con todos los pokemones que coincidan

Ejemplo:

```
obtener_lista_pokemones(pokemones, 'fuerza', 5)
>> Devolvera una lista con todos los pokemones que tenga fuerza 5
```

05.3

Crear la función "string_max_dato" la cual recibirá por parámetro:

- La lista de pokemones

- Un string que representará el máximo

- Un string que representará el dato/key a calcular

La función retornará un string con el valor máximo del dato calculado

y todos los pokemones que cumplan dicha condición

Ejemplo:

```
calcular_max_dato(pokemones, 'máximo', 'poder')
>> "poder máximo: 30 | pokemones: gengar - articuno - zapdos - moltres"
```

05.4

Crear la función "imprimir_pokemones_fuertes" la cual recibirá por parámetro el string formateado con los pokemones más fuertes y deberá imprimirlo

06.1

Crear la función "calcular_min_dato" la cual recibirá por parámetro:

- La lista de pokemones

- Un string que representará el mínimo

- Un string que representará el dato/key a calcular

La función retornará el valor mínimo de la key calculada.

06.2

Crear la función "string_min_dato" la cual recibirá por parámetro:

- La lista de pokemones

- Un string que representará el mínimo

- Un string que representará el dato/key a calcular

La función retornará un string con el valor mínimo del dato calculado

y todos los pokemones que cumplan dicha condición

Ejemplo:

```
calcular_max_dato(pokemones, 'mínimo', 'poder')
>> "poder mínimo: 3 | pokemones: pichu"
```

06.3

Crear la función "imprimir_pokemones_debiles" la cual recibirá por parámetro el string formateado con los pokemones más debiles y deberá imprimirlo

6.3.1 (opcional)

Crear la función "calcular_min_max_dato" la cual recibirá por parámetro:

- La lista de pokemones

- Un string que representará el mínimo o máximo

- Un string que representará el dato/key a calcular

La función imprimirá todos los pokemones que cumplan dicha condición

07 a 09

Crear la función 'calcular_promedio_dato' el cual recibirá por parámetro:

- Una lista de pokemones (no necesariamente la original)

- Un string que representará el dato/key a calcular

- Un string que representará el tipo de pokemon

La función deberá calcular el promedio de dicha key SOLO de los pokemones que tenga ese tipo

(fuego, agua, etc) y retornarlo como un float

Crear la función 'pokedex_imprimir_promedio_dato_tipo' el cual recibirá por parámetro:

- La lista de pokemones

- Un string que representará el dato/key a calcular

- Un string que representará el tipo de pokemon

La función deberá imprimir un mensaje que diga cual es el promedio de esa key y ese tipo de pokemon

Ejemplo:

```
pokedex_imprimir_promedio_dato_tipo(pokemones, 'fuerza', 'fuego')
>> "El promedio de fuerza de pokemones fuego es: N"
```

10.1

Crear la función "tiene_varios_tipos" la cual recibirá por parámetro un diccionario que representará al pokémon y verificará que el pokémon posea más de un tipo retornando True o sino False caso contrario.

10.2

Crear la función "obtener_pokemones_varios_tipos" la cual recibirá por parámetro una lista de pokemones y verificará los pokemones que posean más de un tipo, quienes cumplan con la condición, deberán ser guardados en una lista auxiliar. La función retornará dicha lista.

Reutilizar:

- 'tiene_varios_tipos'

10.3

Crear la función "pokedex_imprimir_pokemones_varios_tipos" la cual recibirá por parámetro:

La lista de pokemones

La función imprimirá los nombres formateados de todos los pokemones que posean más de un tipo.

Reutilizar:

'obtener_pokemones_varios_tipos', 'pokedex_imprimir_nombres_poke_fmt'

11.1

Crear la función "obtener_pokemones_cantidad_tipos" la cual recibirá por parámetro:

La lista de pokemones que tenga más de un tipo

La función deberá generar un diccionario solo con el nombre de cada pokémon como clave y la cantidad de tipos que tenga, como valor. Finalmente retornará dicho diccionario.

Reutilizar:

'obtener_pokemones_varios_tipos'

11.2

Crear la función 'pokedex_imprimir_pokemones_cantidad_tipos' la cual recibirá por parámetro:

La lista de pokemones

La función deberá iterar un diccionario que contendrá el nombre de cada pokemon y la cantidad de tipos que posea, imprimiendo cada clave y su valor

Ejemplo:

```
"Pokemon: articuno | Cantidad de tipos: 2"
```

12.1

Crear la función "tiene_varias_evos" la cual recibirá por parámetro un diccionario que representará al pokémon y verificará que el pokémon posea más de una evolución retornando True o sino False caso contrario.

12.2

Crear la función "obtener_pokemones_varias_evos" la cual recibirá por parámetro una lista de pokemones y verificará los pokemones que posean más de una evolución, quienes cumplan con la condición, deberán ser guardados en una lista auxiliar

La función retornará dicha lista.

Reutilizar:

'tiene_varias_evos'

12.3

Crear la función "pokedex_imprimir_pokemones_varias_evos" la cual recibirá por parámetro:
La lista de pokemones

La función imprimirá los nombres formateados de todos los pokemones que posean más de una evolución.

Reutilizar:

'obtener_pokemones_varias_evos', 'pokedex_imprimir_nombres_poke_fmt'

13.1

Crear la función "obtener_pokemones_cantidad_evos" la cual recibirá por parámetro:

La lista de pokemones que tenga más de un tipo

La función deberá generar un diccionario solo con el nombre de cada pokémon como clave y la cantidad de evoluciones que tenga, como valor. Finalmente retornará dicho diccionario.

Reutilizar:

'obtener_pokemones_varias_evos'

13.2

Crear la función 'pokedex_imprimir_pokemones_cantidad_evos' la cual recibirá por parámetro:

La lista de pokemones

La función deberá iterar un diccionario que contendrá el nombre de cada pokemon y la cantidad de evoluciones que posea, imprimiendo cada clave y su valor

Reutilizar:

'obtener_pokemones_cantidad_evos'

Ejemplo:

```
"Pokemon: mewtwo | Cantidad de evoluciones: 2"  
"Pokemon: charizard | Cantidad de evoluciones: 3"
```

14.1

Crear la función "tiene_varias_fortalezas" la cual recibirá por parámetro un diccionario que representará al pokémon y verificará que el pokémon posea más de una fortaleza retornando True o sino False caso contrario.

14.2

Crear la función "obtener_pokemones_varias_fortalezas" la cual recibirá por parámetro una lista de pokemones y verificará los pokemones que posean más de una fortaleza, quienes cumplan con la condición, deberán ser guardados en una lista auxiliar

La función retornará dicha lista.

Reutilizar:

'tiene_varias_fortalezas'

14.3

Crear la función "pokedex_imprimir_pokemones_varias_fortalezas" la cual recibirá por parámetro:

La lista de pokemones

La función imprimirá los nombres formateados de todos los pokemones que posean más de una fortaleza.

Reutilizar:

'obtener_pokemones_varias_fortalezas', 'pokedex_imprimir_nombres_poke_fmt'

15.1

Crear la función "obtener_pokemones_cantidad_fortalezas" la cual recibirá por parámetro:

La lista de pokemones que tenga más de una fortaleza

La función deberá generar un diccionario solo con el nombre de cada pokémon como clave y la cantidad de fortaleza que tenga, como valor. Finalmente retornará dicho diccionario.

Reutilizar:

'obtener_pokemones_varias_fortalezas'

15.2

Crear la función 'pokedex_imprimir_pokemones_cantidad_fortalezas' la cual recibirá por parámetro:

La lista de pokemones

La función deberá iterar un diccionario que contendrá el nombre de cada pokemon y la cantidad de fortaleza que posea, imprimiendo cada clave y su valor

Reutilizar:

'obtener_pokemones_cantidad_fortalezas'

Ejemplo:

```
"Pokemon: articuno | Cantidad de fortalezas: 2"
"Pokemon: mewtwo | Cantidad de fortalezas: 4"
```

16.1

Crear la función "tiene_varias_debilidades" la cual recibirá por parámetro un diccionario que representará al pokémon y verificará que el pokémon posea más de una debilidad retornando True o sino False caso contrario.

16.2

Crear la función "obtener_pokemones_varias_debilidades" la cual recibirá por parámetro una lista de pokemones y verificará los pokemones que posean más de una debilidad, quienes cumplan con la condición, deberán ser guardados en una lista auxiliar

La función retornará dicha lista.

Reutilizar:

'tiene_varias_debilidades'

16.3

Crear la función "pokedex_imprimir_pokemones_varias_debilidades" la cual recibirá por parámetro:

La lista de pokemones

La función imprimirá los nombres formateados de todos los pokemones que posean más de una debilidad.

Reutilizar:

'obtener_pokemones_varias_debilidades', 'pokedex_imprimir_nombres_poke_fmt'

17.1

Crear la función "obtener_pokemones_cantidad_debilidades" la cual recibirá por parámetro:

La lista de pokemones que tenga más de una debilidad

La función deberá generar un diccionario solo con el nombre de cada pokémon como clave y la cantidad de debilidades que tenga, como valor. Finalmente retornará dicho diccionario.

Reutilizar:

'obtener_pokemones_varias_debilidades'

17.2

Crear la función 'pokedex_imprimir_pokemones_cantidad_debilidades' la cual recibirá por parámetro:

La lista de pokemones

La función deberá iterar un diccionario que contendrá el nombre de cada pokemon y la cantidad de debilidades que posea, imprimiendo cada clave y su valor

Reutilizar:

'obtener_pokemones_cantidad_debilidades'

Ejemplo:

```
"Pokemon: articuno | Cantidad de debilidades: 3"
"Pokemon: mewtwo | Cantidad de debilidades: 2"
```

Tarea para casa:

Cómo harías la refactorización de estas últimas funciones, para crear funciones genéricas que hagan lo mismo pasándole como keys: (Ejercicios 10 a 17)

"tipo", "debilidad", "fortaleza" y "evoluciones" ?