Lavorare con i dati in python: JSON

Docente: Tommaso Muraca

Python e JSON

JSON, acronimo di **JavaScript Object Notation**, è un **formato di scambio dati** basato su un sottoinsieme del linguaggio di programmazione JavaScript. È diventato uno **standard per la trasmissione di dati strutturati** tra un server e un client e viene utilizzato in molti contesti, inclusi lo sviluppo web e le **API** (Application Programming Interface) per lo **scambio di dati.**

Un file json si presenta così:

{ ‘proprieta1’: 'Valore',

‘proprieta2’: 'Valore',

‘proprietaN’: 'Valore’ }

Come potete notare **la sintassi è identica ai dizionari in python**, per evitare di fare confusione python non fornisce supporto nativo ma ci basta aggiungere il **modulo della standard library json** per lavorarci.

Python e JSON

Per **leggere** un elemento **json in python** ci basta usare il metodo **json.load():**

```
import json
# elemento JSON:
x =  '{ "name":"John", "age":30, "city":"New York"}'
# leggi x:
y = json.loads(x)
# il risultato è un dizionario Python quindi per avere il valore basta scrivere:
print(y["age"])
```

Python e JSON

Invece per **convertire un elemento python in json** dobbiamo usare il metodo **json.dumps():**

```
import json
# dizionario python:
x = { "name": "John",
  "age": 30,
  "city": "New York"}
# conversione in JSON:
y = json.dumps(x)
# risultato stringa
print(y)
```

Python e JSON

Il metodo **dumps()** ha **diversi parametri opzionali**:

- **json.dumps(x, indent=4)**

  Ci permette di aggiungere degli spazi in modo da rendere il file più leggibile;

- **json.dumps(x, indent=4, separators=(". ", " = "))**

  Ci permette di definire i separatori tra le chiavi : valore del json, il valore predefinito è (", ", ": ");

- **json.dumps(x, indent=4, sort_keys=True)**

  Ci permette di ordinare le chiavi nel risultato.

Python e JSON e API

Ma cosa sono le API? **API è l'acronimo di interfaccia di programmazione delle applicazioni.**

Possiamo dire che sono un **intermediario** grazie al quale **due applicazioni** distinte e separate possono **comunicare tra loro**.

Lo **scopo delle API è integrare dati**, applicazioni e dispositivi create da aziende diverse e magari programmatori diversi.

Uno dei **formati più utilizzato** nello scambio di dati con le **API** è appunto il **JSON**, per capire come funzionano possiamo affidarci a **2 strumenti**:

- https://open-meteo.com/ per studiare **API gratuite sul meteo**;
- https://web.postman.co/ strumento per il **testing delle api**.

Andiamo subito a vederne il funzionamento.

Python e JSON e API

Come visto in precedenza e rivisto ora grazie alla **funzione «code» di Postman**, per poter **interagire con il web python** ci fornisce la **libreria requests**.

Per utilizzarla con i json abbiamo **2 modi**:

```
import requests
response = requests.get("https://www.google.it")
#response = requests.request("GET", "https://www.google.it)
print(response.json())
```

```
import requests
Import json
response = requests.get("https://www.google.it")
#response = requests.request("GET", "https://www.google.it)
Response_text=response.text()
Response_json = json.loads(Response_text)
```

Primo Esercizio

Create un programma python che permette tramite le api https://open-meteo.com/en/docs (per le previsioni metereologiche) e https://www.bigdatacloud.com/free-api/free-reverse-geocode-to-city-api#getStarted (per l'ottenimento in automatico della propria langitudine e latitudine tramite l'indirizzo ip), di vedere le previsione metereologiche.

L'utente potrà scegliere se visionarle dei prossimi 1, 3 o 7 giorni e se visionare oltre che le temperature anche la velocità del vento e le probabili precipitazioni.

Secondo Esercizio

Create un programma python utilizzando le api https://pokeapi.co/api/v2/pokemon/ {numero} che simula un pokedex, quando troverete un pokemon in maniera randomica verificherà se è presente nel vostro pokedex (pokedex.json), in caso non fosse presente vi permetterà di catturarlo salvando il numero identificativo, nome, abilità, xp(punti esperienza),peso e altezza.

(Sul sistema API sono presenti poco più di 1000 pokemon)

Terzo Esercizio

Create un programma python utilizzando le api https://opentdb.com/api_config.php per creare un quiz per 2 squadre.

Si avranno 10 domande in totale, ogni squadra riceverà una domanda facile sul cinema con risposta vero o falso, se la squadra indovina la risposta riceve 3 punti, se sbaglia perde 1 punto, chi ha il punteggio più alto alla fine delle domande vince la partita.

Bonus: Potete tradure le domande con https://api.mymemory.translated.net/get?q=***frase da inserire***&langpair=en|it ,per farlo ricordatevi di selezionare url3986 su opentdb

Quarto Esercizio

Modificate il programma creato nell'esercizio precedente.

A inizio gioco si potrà scegliere:

- Numero squadre;
- Numero domande totali;
- Difficolta domande;
- Categoria domande;
- Se avere Difficoltà e Categorie casuali per ogni domanda.