

Python - attività 5: memorizzazione in formato JSON

Lo scopo di questa attività è memorizzare i dati ricevuti dal sensore in modo da poterli rendere disponibili successivamente.

Parte 1: Dizionari - attività preliminare (senza hardware)

Fase 1.1: studiare come funzionano i dizionari in [questo video](#) (7:16)

Fase 1.2: studiare come funzionano le liste di dizionari a [questo link](#)

Fase 1.3: svolgere il seguente esercizio:

```
Scrivere un programma in Python che chieda all'utente di inserire delle coppie
DataOra-Valore e li inserisca in una lista di dizionari.
Le coppie saranno inserite con formato seguente, dove la virgola è il separatore:

2021-11-23 17:50:34,134

e il dizionario creato sarà:

{'DataOra': '2021-11-23 17:50:34', 'Valore': 134}

Quando l'utente ha terminato l'inserimento, lo segnalerà con una riga vuota.
Successivamente si stamperanno le ultime 5 coppie inserite.
```

Parte 2: JSON - attività preliminare (senza hardware)

Fase 2.1: studiare cos'è JSON in [questo video](#) (6:52)

Fase 2.2: studiare come trasferire una lista di dizionari su di un file JSON a [questo link](#)

Fase 2.3: svolgere il seguente esercizio:

```
Modificare il programma precedente in modo che, quando l'utente ha terminato
l'inserimento, si scriva la lista di dizionari su file in formato JSON.
Successivamente si analizzi il contenuto del file scritto con un editor di testo.
Si scriva poi un programma che legga il file JSON e stampi le ultime 5 coppie
presenti nella lista.
```

Parte 3: scrittura su file JSON dei dati del sensore

Fase 3.1: sviluppare il seguente programma:

```
Scrivere un programma che riceva i dati del sensore da Arduino, crei un dizionario
con le chiavi DataOra e Valore.

Il dizionario creato venga poi aggiunto in coda ad una lista, tenendo solo le
ultime 10 rilevazioni.

La lista venga poi scritta su di un file JSON.

Converrà probabilmente, dopo aver effettuato i test, ridurre la frequenza di invio
da parte di Arduino, ad esempio ogni 30 secondi.
```