**Python - attività 7: pubblicazione dati sensore su Web**

*Lo scopo di questa attività è rendere disponibili su Web i dati rilevati dal sensore.*

**Parte 1: HTML**

**Fase 1.1:** analizzare le prime parti del corso HTML a [questo link](https://www.w3schools.com/html/html_intro.asp)

**Fase 1.2:** consultare[questo link](https://www.w3schools.com/html/html_tables.asp) per inserire tabelle nella pagina HTML

**Fase 1.3:** svolgere il seguente esercizio:

Scrivere il codice HTML per visualizzare la tabella seguente. Salvare il file con estensione *html* ed aprirlo con un browser.

|  |  |
| --- | --- |
| **Data e ora** | **Valore** |
| 2021-11-17 17.53.10 | 125 |
| 2021-11-17 17.54.10 | 127 |
| 2021-11-17 17.55.10 | 131 |

**Parte 2: Creazione HTML da Python**

**Fase 2.1:** consultare[questo link](https://www.programmareinpython.it/gestire-file-cartelle-e-file-system-con-python/creare-leggere-e-scrivere-file-testuali-in-python/) per scrivere un file di testo da Python

**Fase 2.2:** svolgere il seguente esercizio:

Considerare il file JSON creato nell’attività precedente e scrivere un file HTML che contenga i dati letti dal file JSON, creando una pagina simile a quella dell’esercizio precedente.

Converrà creare una funzione che trasformi la lista in stringa con gli opportuni tag <tr> e <td> e poi scrivere la stringa sul file HTML. Tale funzione potrà essere riutilizzata successivamente.

**Parte 3: Flask – Web Server**

**Fase 3.1:** analizzare in cosa consiste il protocollo HTTP in [questo video](https://www.youtube.com/watch?v=RsQ1tFLwldY) (7:46)

**Fase 3.2:** consultare[questo link](https://flask.palletsprojects.com/en/2.0.x/quickstart/) per la prima applicazione Flask

**Fase 3.3:** svolgere il seguente esercizio:

Considerare l’esercizio precedente ed in particolare la funzione che trasforma la lista e costruisce la stringa contenente il codice HTML.

Modificare l’applicazione Flask dell’esempio in modo che restituisca il codice HTML generato dalla suddetta funzione.

Provare l’applicazione accedendo da un browser all’indirizzo:

*http://127.0.0.1:5000*

**Parte 4: ngrok – rendere visibile il server su Internet**

**Fase 4.1:** scaricare ***ngrok*** a [questo link](https://ngrok.com/download)

**Fase 4.2:** consultare[questo link](https://ngrok.com/docs) per l’uso di ngrok

**Fase 4.3:** impostare il proxy, se necessario

set http\_proxy=http://proxy.intranet:3128

set https\_proxy=http://proxy.intranet:3128

**Fase 4.4:** rendere visibile su Internet l’applicazione Flask dell’esercizio precedente e verificarne il funzionamento

**Parte 5: test d’insieme**

Verificare il funzionamento del sistema in tutte le sue parti:

1. Programma sensore attivo su Arduino
2. Programma Python che riceve i dati del sensore e scrive, in continuazione, il file JSON
3. Applicazione Flask che legge il file JSON e produce il codice HTML
4. Tunnel per esporre il Web Server su Internet

**Parte 6: approfondimenti**

1. Migliorare la produzione di codice HTML utilizzando i [*template*](https://flask.palletsprojects.com/en/2.0.x/quickstart/#rendering-templates)
2. Aggiungere grafici (vedi [*Plotly*](https://towardsdatascience.com/web-visualization-with-plotly-and-flask-3660abf9c946))
3. Studiare il funzionamento dei [*decorators*](https://www.youtube.com/watch?v=FsAPt_9Bf3U)