

Python - attività 8: controllo dell'attuatore da browser

Lo scopo di questa attività è inviare dati da Web all'applicazione Flask, in modo da poter controllare l'attuatore.

Considerate il seguente esempio (mostrato in due versioni, la seconda con l'uso dei templates)

Versione 1:

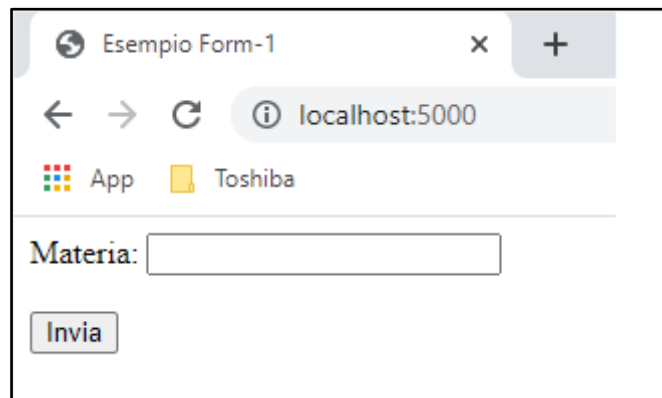
```
1  from flask import Flask,request
2
3  app = Flask(__name__)
4  #-----
5  # invia il form vuoto
6  #-----
7  @app.route("/")
8  def inviaFormVuoto():
9      f='''
10     <html>
11         <head>
12             <title>Esempio Form</title>
13         </head>
14         <body>
15             <form action="/ricevi">
16                 <label>Materia:</label>
17                 <input type="text" name="materia"><br><br>
18                 <input type="submit" value="Invia">
19             </form>
20         </body>
21     </html>
22     '''
23     return(f)
24 #-----
25 # riceve i dati del form e li visualizza
26 #-----
27 @app.route("/ricevi")
28 def riceviForm():
29     return(request.args["materia"])
```

Versione 2:

```
1  from flask import Flask,request,render_template
2
3
4  app = Flask(__name__)
5
6  #-----
7  # invia il form vuoto
8  #-----
9  @app.route("/")
10 def inviaFormVuoto():
11     return(render_template("form1.html"))
12
13 #-----
14 # riceve i dati del form e li visualizza
15 #-----
16 @app.route("/ricevi")
17 def riceviForm():
18     return(request.args["materia"])
```

```
1  <html>
2      <head>
3          <title>Esempio Form-1</title>
4      </head>
5      <body>
6          <form action="/ricevi">
7              <label>Materia:</label>
8              <input type="text" name="materia"><br><br>
9              <input type="submit" value="Invia">
10          </form>
11      </body>
12  </html>
```

Entrambe le versioni producono la seguente pagina:



Esempio Form-1

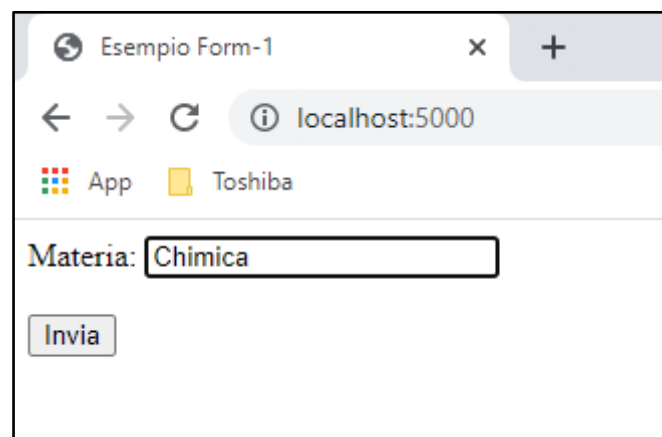
localhost:5000

App Toshiba

Materia:

Invia

Si scrive poi nel campo di input:



Esempio Form-1

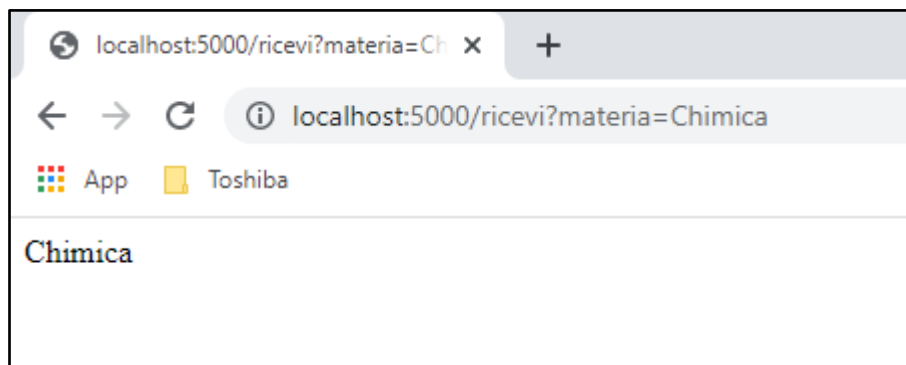
localhost:5000

App Toshiba

Materia:

Invia

E si preme invio ottenendo:



localhost:5000/ricevi?materia=Chimica

App Toshiba

Chimica

Parte 1: Form di invio comandi all'attuatore

Lo scopo di questa parte è realizzare un form con un campo di input in cui scrivere la velocità, due radio buttons per selezionare la direzione e il tasto di invio

Fase 1.1: analizzare l'uso dei radio buttons [questo link](#);

Fase 1.2: verificare, stampando `request.args`, cosa si riceve in Flask quando un radio button è selezionato

Fase 1.3: svolgere il seguente esercizio:

Realizzare un'applicazione Flask che invii un form composto da un capo di input in cui scrivere la velocità (0-255) da inviare al motore, due radio button per la direzione (uno selezionato di default) e il bottone di invio. L'applicazione Flask riceve il form e prepara la stringa da inviare all'attuatore su Arduino per il controllo del motore.

Parte 2: approfondimento

Unificare in un'unica pagina web la visualizzazione dei risultati del sensore e il controllo del motore