

Esercitazione di TPSIT su Flask/Python/JSON

1. Decomprimete la cartella Bethany
2. Fate una copia della pagina index.html (index2.html per esempio o qualsiasi altro nome)
3. Rimuovete il codice HTML che non serve, cioè lasciate le zone Header e Portfolio (tenete il resto del codice HTML)
4. Create le cartelle *static* e *templates*
5. Copiate tutti file HTML in *templates* (copiate eventualmente i file di testo ma in realtà questi non servono ai fini dell'esercitazione)
6. Copiate tutti i file statici nella cartella *static*
7. Modificate la pagina index1.html in questo modo: aggiungete ai tag script, img, link, a-href prima di asset la cartella static
8. Creare il file images.json all'interno della cartella *static/assets/json*. Il file dovrebbe somigliare al seguente:

```
[
  {
    "title": "ABC",
    "subtitle": "abc",
    "filename": "portfolio-1.jpg"
  },
  {
    "title": "EFG",
    "subtitle": "efg",
    "filename": "portfolio-2.jpg"
  },
  {
    "title": "HIJ",
    "subtitle": "hij",
    "filename": "portfolio-3.jpg"
  }
  ...
]
```

9. Occorre modificare il file Python main.py nel modo seguente:

```
from flask import Flask, render_template
import json

app = Flask(__name__)

images = None # il riferimento alla lista di oggetti di tipo dizionario. Ogni
              # dizionario contiene le 3 chiavi title, subtitle e image

# caricamento degli oggetti json nella lista images
with open('static/assets/json/images.json') as f:
    images = json.load(f)

print(images) # stampa di controllo

@app.route('/')
def index():
    return render_template('index1.html', images=images)

@app.route('/portfolio_details')
def portfolio_details():
    return render_template('portfolio-details.html')

app.run(host='0.0.0.0', port=8080, debug=True)
```

10. Fate la stessa cosa descritta nel punto 7 al file portfolio-details.html

11. All'interno della pagina index1.html, cancellate i 9 tag div che contengono i riferimenti alle 9 immagini (portfolio-1.jpg, portfolio-2.jpg ecc fino a portfolio-9.jpg) sostituendoli con un solo tag div. Usate ora un ciclo for, modificando i riferimenti alle immagini e alla pagina portfolio-details.html in questo modo:

```
{% for image in images %}
<div class="col-lg-4 col-md-6 portfolio-item filter-app">
    ...
    

    <div class="portfolio-info">
        <h4>{{ image.title }}</h4>
        <p>{{ image.subtitle }}</p>
        ...
        <a href="static/assets/img/portfolio/{{ image.filename }}" data-
gall="portfolioGallery" class="venobox" title="{{ image.title }}"><i class="bx bx-
plus"></i></a>
        <a href="{{ url_for('portfolio_details') }}" title="More Details"><i
class="bx bx-link"></i></a>
    </div>
</div>
</div>
{% endfor %}
...
```

12. Ora attivate il server. Se c'è qualcosa che non funziona, provate a riavviare il server e ad aggiornare la pagina di repl.it. Se le cose non vanno ed ottenete messaggi di errore da parte del server dittico 500 (malfunzionamento interno del server o errore 404 ecc), significa che avete sbagliato qualcosa e occorrerà cercare meglio. Il debug è complicato se usate repl.it ma è il modo più rapido per evitare di installare tutto sulla vostra macchina. Fate comunque riferimento ai video di YouTube che sto inserendo come attività asincrona all'ottava ora del lunedì o al PDF di Miguel Grinberg sullo sviluppo web con Flask che ho postato in Teams. E' tutto.