

# Soluzione dell'esame scritto di Tecnologie web del 13/01/2022 (prova online)

## Domanda #1 - Domande di base

### a) HTTP

In HTTP, la “catena di montaggio” (o *pipelining*) è una tecnica volta a migliorare le prestazioni della comunicazione in termini di tempo. Essa consiste in una rudimentale forma di parallelismo: il cliente che deve effettuare più richieste HTTP, non aspetta la risposta a ciascuna di queste prima di mandare la successiva; invece, sceglie di mandarne diverse in rapida successione. Questa tecnica garantisce ad ogni modo un’invariante: se una richiesta A è effettuata prima di una richiesta B, allora la risposta ad A viene data prima della risposta a B.

### b) HTML

In un’immagine:

- l’attributo *alt* indica una descrizione dell’immagine da usare in alternativa all’immagine stessa, per esempio quando l’immagine non è disponibile o si fa uso di tecnologia assistiva per ipovedenti o non vedenti;
- l’attributo *title* indica un titolo che integra l’immagine stessa anziché sostituirla, come i crediti, e che può potenzialmente comunque essere usato ai fini di accessibilità.

### c) CSS

L’unica regola non valida è `p @ a`: `@` è valido solo nelle “regole-a” (*at-rules*). Tutte le altre vanno bene: si noti che `p` e `a` non sono nomi riservati né per le classi né per gli identificatori.

### d) JavaScript

In JavaScript, un prototipo è un’istanza di base per un oggetto che funge da fornitore di metodi predefiniti quando questi non sono contenuti nell’oggetto stesso. Un prototipo è potenzialmente condiviso da più oggetti e i metodi definiti per esse sono modificabili in fase di esecuzione senza restrizioni. Nell’ottica della programmazione orientata agli oggetti, quello dei prototipi è da considerarsi un approccio alternativo alle classi (che in JavaScript sono ridotte a mero zucchero sintattico).

## Domanda #2 - HTML + CSS

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```

<meta charset="utf-8" />
<title>Hello world</title>
<link rel="stylesheet"
      href="https://www.fabiovitali.it/TW/lib/bootstrap.min.css" />
<script src="https://www.fabiovitali.it/TW/lib/jquery-3.6.0.js"></script>
<script src="https://www.fabiovitali.it/TW/lib/bootstrap.js"></script>
</head>

<body>
  <div class="d-flex justify-content-around pb-5">
    
    </div>
  <nav class="navbar navbar-expand-lg pb-5">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">
        
        </a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
        data-bs-target="#navbarNav" aria-controls="navbarNav"
        aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse flex-row-reverse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link text-dark" href="#">
              Review</a>
            </li>
          <li class="nav-item">
            <a class="nav-link text-dark" href="#">
              Trips</a>
            </li>
          <li class="nav-item">
            <a class="nav-link text-dark">
              Alerts</a>
            </li>
          <li class="nav-item">
            <a class="nav-link">
              <button type="button" class="nav-link btn btn-dark text-white">

```

```

        Sign in</a>
    </li>
</ul>
</div>
</div>
</nav>
<div class="container-fluid pb-5">
    <div class="row">
        <button type="button"
            class="d-flex justify-content-between col-2 btn btn-white border">
            <span class="p-2">Hotels</span>
            
        </button>
        <button type="button"
            class="d-flex justify-content-between col-2 btn btn-white border">
            <span class="p-2">Vacation Rentals</span>
            
        </button>
        <button type="button"
            class="d-flex justify-content-between col-2 btn btn-white border">
            <span class="p-2">Things to Do</span>
            
        </button>
        <button type="button"
            class="d-flex justify-content-between col-2 btn btn-white border">
            <span class="p-2">Restaurants</span>
            
        </button>
        <button type="button"
            class="d-flex justify-content-between col-2 btn btn-white border">
            <span class="p-2">Travel Forums</span>
            
        </button>
        <button type="button"
            class="d-flex justify-content-between col-2 btn btn-white border">
            <span class="p-2">More</span>

```

```

        
    </button>
</div>
</div>
<div class="card text-center bg-dark text-white mb-5">
    
    <div class="card-img-overlay align-items-center">
        <input class="form-control mt-5 ml-5 mr-5" placeholder="Where to?">
    </div>
</div>
<div>
<h2 class="text-body mb-2">Explore Experiences</h2>
<div class="container-flex mb-5">
    <div class="row d-flex">
        <div class="col-3 align-self-stretch">
            <div class="card" style="width: 18rem;">
                
                <div class="card-body">
                    <p class="card-text">
                        The Lava Tunnell - Standard Lava Tunnel Tour - Raufarhólshellir
                    </p>
                </div>
                
            </div>
        </div>
        <div class="col-3 align-self-stretch">
            <div class="card" style="width: 18rem;">
                
                <div class="card-body">
                    <p class="card-text">
                        The Lava Tunnell - Standard Lava Tunnel Tour - Raufarhólshellir
                    </p>
                </div>
                
            </div>
        </div>
    </div>
</div>

```

```

<div class="col-3 align-self-stretch">
  <div class="card" style="width: 18rem;">
    
    <div class="card-body">
      <p class="card-text">
        The Lava Tunnell - Standard Lava Tunnel Tour - Raufarhólshellir
      </p>
    </div>
    
  </div>
</div>
<div class="col-3 align-self-stretch">
  <div class="card" style="width: 18rem;">
    
    <div class="card-body">
      <p class="card-text">
        The Lava Tunnell - Standard Lava Tunnel Tour - Raufarhólshellir
      </p>
    </div>
    
    
  </div>
</div>
</div>
</body>

</html>

```

## Domanda #3 - Javascript

### Parte I

Il protocollo è `http` o `https`. Il formato potrebbe essere `/ideas/places/{place}/activities/{activity}`. Siccome sono interessato a leggere risorse dal senza modificare alcunché, il metodo è `GET`. I parametri della richiesta sono l'identificativo del luogo e quello dell'attività. In caso di successo (200), la risposta ha nel corpo le quattro proposte pertinenti, le due pertinenti solo per luogo e le due pertinenti solo

per attività. Ciascuna proposta contiene titolo, descrizione, immagine, prezzo, collegamento. In caso di almeno un identificativo fra i parametri della richiesta non riconosciuto, si risponde con un errore di risorsa non trovata (404): il corpo contiene due booleani, ciascuno dei quali indicante se l'identificativo corrispondente sia stato riconosciuto o meno. In caso per almeno una delle tre categorie di idee non ci siano sufficienti proposte, si risponde con un errore generico (400): il corpo contiene, per ciascuna delle tre categorie, un contatore intero non negativo delle idee effettivamente disponibili per quella categoria. È anche disponibile una risorsa `/ideas/` che restituisce sempre, con codice 200, due liste (una per i luoghi e una per le attività). Ciascuna associa a ciascun identificativo il corrispettivo nome.

## Parte 2

```
<html>

<head>
  <script type="text/javascript" src="script.js"></script>
</head>

<body>
  <h1>Pagina dei suggerimenti</h1>
  <h2>Suggerimenti pertinenti<h2>
    <article>
      <h3 id="title1"></h3>
      <img id="picture1"></img>
      <p id="description1">
      <p>
      <p id="price1"></p>
    </article>
    <article>
      <h3 id="title2"></h3>
      <img id="picture2"></img>
      <p id="description2">
      <p>
      <p id="price2"></p>
    </article>
    <article>
      <h3 id="title3"></h3>
      <img id="picture3"></img>
      <p id="description3">
      <p>
      <p id="price3"></p>
    </article>
    <article>
      <h3 id="title4"></h3>
```

```

<img id="picture4"></img>
<p id="description4">
<p>
<p id="price4"></p>
</article>
<h2>Suggerimenti meno pertinenti</h2>
  <!-- Ma che è un blocco esplorabile -->
  <article>
    <h3 id="title5"></h3>
    <img id="picture5"></img>
    <p id="description5">
    <p>
    <p id="price5"></p>
  </article>
  <article>
    <h3 id="title6"></h3>
    <img id="picture6"></img>
    <p id="description6">
    <p>
    <p id="price6"></p>
  </article>
  <article>
    <h3 id="title7"></h3>
    <img id="picture7"></img>
    <p id="description7">
    <p>
    <p id="price7"></p>
  </article>
  <article>
    <h3 id="title8"></h3>
    <img id="picture8"></img>
    <p id="description8">
    <p>
    <p id="price8"></p>
  </article>
<h2>Nuova ricerca</h2>
<form>
  <label>Nuovo luogo
    <select name="place" id="newPlace">
    </select>
  </label>
  <label>Nuova attività
    <select name="activity" id="newActivity">
    </select>
  </label>
  <input type="submit" value="Cerca" id="newSearch"/>

```

```

        </form>
    </body>

</html>

```

### Parte 3

```

let place = 1,
    activity = 1;

function getSuggestions(placeId, activityId) {
    fetch(
        `http://www.fabioitali.it/TW/test/2021/doResponse.php/places/${placeId}/activities/${activityId}`
        { method: "GET" }
    )
    .then((response) => {
        switch (response.status) {
            case 400:
                throw new Error("Non ho trovato abbastanza suggerimenti!");
            case 404:
                throw new Error("Errore di identificativi.");
            default:
                if (!response.ok) throw new Error(`Errore HTTP ${response.status}.`);
                return response.json();
        }
    })
    .then((json) => {
        let ideas = json.ideas;
        for (let i = 0; i < ideas.length; ++i) {
            document.getElementById("title" + i).textContent = ideas[i].title;
            document.getElementById("img" + i).src = ideas[i].img;
            document.getElementById("description" + i).textContent =
                ideas[i].description;
            document.getElementById("price" + i).textContent =
                "$ " + ideas[i].price;
        }
    })
    .catch((error) => {
        throw new Error(error);
    });
}

function getOptions() {
    fetch(`http://www.fabioitali.it/TW/test/2021/doResponse.php/ideas/`, {
        method: "GET",
    })

```



```

.then((response) => {
  switch (response.status) {
    default:
      if (!response.ok) throw new Error(`Errore HTTP ${response.status}.`);
      return response.json();
  }
})
.then((json) => {
  for (let i = 0; i < json.places; ++i)
    document.getElementById("newPlace").innerHTML += `<option ${
      json.places[i].id == place ? "selected" : ""
    } value="${json.places[i].id}">${json.places[i].name}</option>`;
  for (let i = 0; i < json.activities; ++i)
    document.getElementById("newActivity").innerHTML += `<option ${
      json.activities[i].id == activity ? "selected" : ""
    } value="${json.activities[i].id}">${json.activities[i].name}</option>`;
})
.catch((error) => {
  throw new Error(error);
});
}

function newSearch() {
  getSuggestions(
    (place = document.getElementById("newPlace").value),
    (activity = document.getElementById("newActivity").value)
  );
  getOptions();
}

window.onload = () => {
  getSuggestions(place, activity);
  getOptions();
  document.getElementById("newSearch").onclick = newSearch();
};

```

## Domanda #4 - Semantic Web

@prefix my: <...>.

@prefix rdf: <...>.

```

my:bob-dylan
  my:ha-registrato [
    rdf:type my:registrazione;
    my:traccia my:every-grain-of-sand
    my:luogo: my:los-angeles;

```

```

    my:tempo [
      my:anno "1981";
      my:stagione my:primavera
    ];
  ];
my:ha:pubblicato: [
  rdf:type my:pubblicazione;
  my:album my:shot-fo-love;
  my:tempo [
    my:anno "1981";
    my:mese my:agosto
  ]
].

my:shot-of-love:
  rdf:type my:album;
  my:include my:every-grain-of-sand.

```

Il grafo, così come ho scelto di modellarlo, descrive sei entità, di cui quattro anonime.

## Domanda #5 - Domanda di accessibilità

```

<div
  id="rg1"
  class="radiogroup"
  role="radiogroup"
  arialabelledby="radiogroup1-title"
>
  <h3 id="radiogroup1-title">Genere:</h3>
  <span class="radio" role="radio" aria-checked="true" tabindex="0"
    >maschile
  </span>
  <span class="radio" role="radio" aria-checked="false" tabindex="-1">
    femminile
  </span>
  <span class="radio" role="radio" aria-checked="false" tabindex="-1">
    altro
  </span>
  <span class="radio" role="radio" aria-checked="false" tabindex="-1">
    preferisco non dirlo
  </span>
</div>

```