Tecnologie Web

C.d.L. in Informatica e Informatica per il Management

Compito del 23 Giugno 2016

Nome:

Cognome:

Matricola:

C.d.L.:

Team:

Corso seguito nell'A.A.:

☐ Non è la prima volta. Data indicativa dell'ultimo appello provato: _____

☐ Ho già consegnato il progetto. Data indicativa: _____

Importante:

- Indicare ciascun esercizio risolto con una croce sul numero relativo. Saranno corretti solo gli esercizi così segnati.
- Se si consegna un foglio protocollo, scrivere in maniera chiara il numero di ogni esercizio presso la sua soluzione.
- Sul foglio protocollo, indicare inoltre il modo chiaro: nome, cognome e numero di matricola.

Esercizio		**Punti**	**Voto**
1	Domande di base	12	
2	HTML	6	
3	Javascript	6	
4	Semantic Web	6	
5	Teoria	4	
Totale		**34**	

1. Domande di base (12 punti)

Rispondere correttamente a tre delle seguenti domande:

A. Scrivere una stringa a piacere che occupa 4 byte sia nella codifica UTF-8 che in ASCII.

B. Indicare se e per quale motivo la seguente affermazione è vera o falsa.

"*Il metodo PUT di HTTP è sicuro e idempotente*."

C. Qual'è l'output dell'esecuzione del seguente script PHP?

```
$days = array();

$days["january"] = 31;
$days["february"] = 28;
$days["march"] = 31;
$days["april"] = 30;
$days["may"] = 31;
$days["june"] = 31;

foreach ($days as $m => $d) {
        if ($d < 30) echo $m;
}
```

D. Scrivere un espressione XPath che applicata al seguente documento XML restituisca l'elemento `<month name="june" days="30"/>`.

```
<months>
  <month name="january" days="31"/>
  <month name="february" days="28"/>
  <month name="march" days="31"/>
  <month name="april" days="30"/>
  <month name="may" days="31"/>
  <month name="june" days="30"/>
<months>
```

2. HTML (6 punti)

Dato il codice HTML mostrato di seguito:

1. Individuare e descrivere almeno 3 errori contenuti nel codice HTML fornito.

2. Scrivere il codice CSS per ottenere la visualizzazione mostrata in Figura 1. Si tenga presente che:

    - il testo delle voci del menu di navigazione è bianco; lo sfondo è arancione (per le voci non selezionate) e grigio (per la voce selezionata); la voce selezionata ha il bordo nero;
    - il bordo esterno dell'immagine non appartiene alla pagina e non va creato;
    - il blocco con il contenuto principale ha il bordo nero ed è posizionato al centro della pagina;
    - il contenuto della prima cella di ogni colonna è al centro in grassetto;
    - i luoghi nella descrizione sono in corsivo;
    - le immagini sono alte 150px e larghe 200px;
    - le immagini sono separate da un margine, e sono allineate al centro della pagina.
    - dove non specificato, le dimensioni esatte di margini e padding non sono rilevanti

3. Si scriva/modifichi il codice HTML e CSS per aggiungere l'immagine di una freccia verso sinistra prima della prima fotografia (file `images/left-arrow.gif`), e l'immagine di una freccia verso destra dopo l'ultima fotografia (file `images/right-arrow.gif`). La dimensione delle immagine è 100x50 px (larghezza per altezza). Tali immagini devono essere allineate sia verticalmente che orizzontalmente.

Si tengano in considerazione questi vincoli:

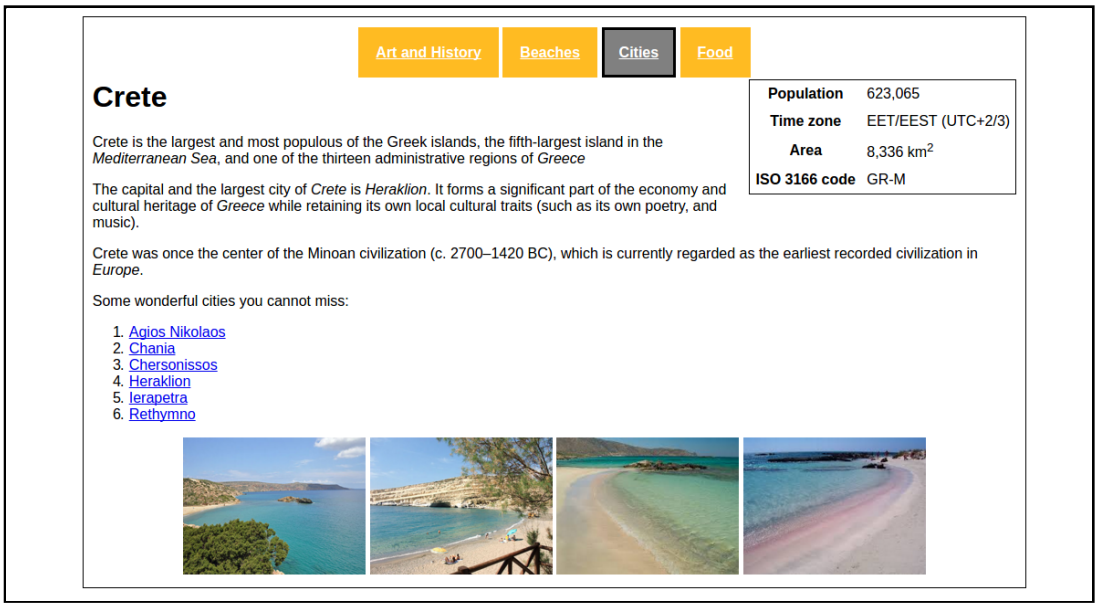
- non è possibile aggiungere nè fare riferimento ad altri attributi `id` o `class` tranne quelli presenti nel testo dell'esercizio;
- i contenuti della pagina sono 'statici' (eventuali comportamenti dinamici vannno nell'esercizio Javascript).

Sorgente HTML:

```
<!DOCTYPE html>
<html>
  <head>
    ...
  </head>
  <body>
    <div id="main">
      <nav id="nav">
        <a href="/art">Art and History</a>
        <a href="/beach">Beaches</a>
        <a href="/cities">Cities</a>
        <a href="/food">Food</a>
      </nav>
      <table id='info'>
            <tr><td>Population</td><th>623,065</th></tr>
            <tr><td>Time zone</td><th>EET/EEST (UTC+2/3)</th></tr>
            <tr>
                <td>Area</td>
```

```
            <th>8,336 km<superscript>2</superscript></th>
          </tr>
          <tr><td>ISO 3166 code</td><th>GR-M</th></tr>
        </table>
        <div id="content">
          <h1>Crete</h1>
          <span>
                  Crete is the largest and most populous of the Greek islands,
                  the fifth-largest island in the
                  <span class="place">Mediterranean Sea</span>, and one of the
                  thirteen administrative regions of
                  <span class="place">Greece</span>.<br/>
          <span>
                  The capital and the largest city of
                  <span class="place">Crete</span> is
                  <span class="place">Heraklion</span>. It forms a significant
                  part of the economy and cultural heritage of
                  <span class="place">Greece</span> while retaining its own
                  local cultural traits (such as its own poetry, and music).
                  </span><br/>
          <span>
                  Crete was once the center of the Minoan civilization
                  (c. 2700–1420 BC), which is currently regarded as the
                  earliest recorded civilization in
                  <span class="place">Europe</span>.</span><br/>
          <span>Some wonderful cities you cannot miss:</span><br/>
          <ul>
            <li><a href='/cities/agios'>Agios Nikolaos</a></li>
            <li><a href='/cities/chania'>Chania</a></li>
            <li><a href='/cities/chersonissos'>Chersonissos</a></li>
            <li><a href='/cities/heraklion'>Heraklion</a></li>
            <li><a href='/cities/ierapetra'>Ierapetra</a></li>
            <li><a href='/cities/rethymno'>Rethymno</a></li>
          </ul>
          <div id='pictures'>
            <img src="images/crete01.jpg"/>
            <img src="images/crete02.jpg"/>
            <img src="images/crete03.jpg"/>
            <img src="images/crete04.jpg"/>
          </div>
        </div>
      </div>
    </body>
</html>
```

Figura 1 - Resa della pagina in un browser:



3. Javascript (6 punti)

Si riprenda in considerazione l'esercizio HTML della domanda precedente.

Al caricamento della pagina, l'area delle fotografie è vuota. Viene creata una variabile *numImmagini* (ad esempio uguale a 4) e una variabile *timeInterval* (ad esempio uguale a 2000).

Esiste poi un servizio all'indirizzo *http://www.cretetourism.gr/getPictures.py*, con metodo *GET* e parametri *start* e *end*, che ritorna in oggetto JSON con i seguenti campi:

- *max*: numero massimo di immagini disponibili
- *start*: la posizione di partenza fornita in input
- *pictures*: URL e descrizione alternativa delle immagini richieste, dalla posizione *start* alla posizione *end*

Un esempio è mostrato in seguito:

```
{
        "max": 146,
        "start": 0,
        "pictures": [
                {"pos": 0, "url": "images/abc.jpg", "alt":"First image of Crete"},
                {"pos": 1, "url": "images/def.jpg", "alt":"Second image of Crete"},
                {"pos": 2, "url": "images/ghi.jpg", "alt":"Third image of Crete"},
                {"pos": 3, "url": "images/jkl.jpg", "alt":"Fourth image of Crete"}
        ]
}
```

Usando un framework Javascript a piacere, si realizzino gli script necessari per i seguenti comportamenti:

1. Al caricamento della pagina, si interroga il servizio *getPictures.py* caricando gli indirizzi delle prime *numImmagini* fotografie e le si visualizza nell'area corrispondente. Non è necessario realizzare lo script *getPictures.py*. In caso di errore di caricamento si visualizza ove necessario l'immagine *images/placeholder.gif*.
2. Agendo sul pulsante destro (freccia destra, aggiunta nel precedente esercizio), si caricano le *numImmagini* fotografie successive a quelle attualmente visualizzate. Si assuma per semplicità che *max* è sempre un multiplo intero di *numImmagini*.
3. Ogni *timeInterval* millisecondi, le fotografie scrollano verso sinistra di una posizione (cioè la prima fotografia a sinistra sparisce, la seconda diventa la prima, la terza diventa la seconda, ecc. e viene caricata attraverso il servizio *getPictures.py* una nuova fotografia da porre in ultima posizione. Per semplicità si assume che quando arriva all'ultima fotografia lo scroll si interrompe. NON è consentito usare librerie di carousel di JQuery, Twitter Bootstrap o altri framework esistenti.

4. Semantic Web (6 punti)

Si consideri il seguente contesto JSON-LD:

```
{
    "@context": {
        "dbr": "http://dbpedia.org/resource/",
        "xsd": "http://www.w3.org/2001/XMLSchema#",
        "italy": "dbr:Italy_national_football_team"
        "ita_player_base_url": "italy:/",
        "foaf": "http://xmlns.com/foaf/0.1/",
        "team": "foaf:Organization",
        "player": "foaf:Person",
        "player_of": {
                "@id": "foaf:member",
                "has_type": "@id"
        },
        "plays_with": {
                "@id": "foaf:knows",
                "@type": "@id"
        },
        "name": "foaf:name"^^xsd:string,
        "number": {
                "@id": "foaf:status",
                "@type": "xsd:positiveInteger"
        }
    }
}
```

Rispondere alle seguenti domande:

1. Correggere tutti gli errori sintattici presenti nel contesto JSON-LD.
2. Utilizzando **tutto** il contesto introdotto (e precedentemente corretto) messo a disposizione all'URL “http://www.twexams.com/jsonld/context.json”, e senza introdurre nessun'altra proprietà aggiuntiva rispetto a quelle già descritte, scrivere un documento JSON-LD che contenga le seguenti affermazioni:
    - Buffon (numero di maglia: 1) è un giocatore della squadra della Nazionale Italiana;
    - Insigne (numero di maglia: 20) è un giocatore della squadra della Nazionale Italiana e gioca con Buffon.
3. Qual è il numero di statement RDF creati mediante il documento JSON-LD appena definito?

5. Teoria (4 punti)

Descrivere i due principali modelli di progettazione di XSLT, iterativo e ricorsivo, e fornire esempi di entrambi.