

Tecnologie Web

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

Compito del 4 giugno 2014

Nome:

Cognome:

Matricola:

C.d.L.:

Team:

Corso seguito nell'A.A.:

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

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 e, in intestazione, indicare 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. Quali dei seguenti marcatori HTML deve essere usato per creare una lista ordinata?

- p
- table
- ol
- ul
- span
- list

B. Descrivere brevemente il concetto di "Cascata" in CSS.

C. Qual è il valore della variabile "t" alla fine dell'esecuzione di questo script Python?

```
t = "Questa è una stringa"  
t = t.split(" ")  
t = t[3]
```

Indicare un valore tra i seguenti:

- 3
- "stringa"
- "una"
- None
- "Questa è una stringa"

D. Scrivere un frammento JSON con tre campi ("nome", "cognome" e "matricola") contenenti i vostri dati.

2. HTML (6 punti)

Scrivere il codice HTML 5 (e CSS) di un documento che rappresenti il contenuto di questa immagine:

[Art and History](#) [Beaches](#) [Cities](#) [Food](#)

Crete


Crete is the largest and most populous of the Greek islands, the fifth-largest island in the Mediterranean Sea, and one of the thirteen administrative regions of Greece.

The capital and the largest city of Crete is Heraklion. It forms a significant part of the economy and cultural heritage of Greece while retaining its own local cultural traits (such as its own poetry, and music).

Crete was once the center of the Minoan civilization (c. 2700–1420 BC), which is currently regarded as the earliest recorded civilization in Europe.

Some wonderful cities you cannot miss:

- [Agios Nikolaos](#)
- [Chania](#)
- [Chersonissos](#)
- [Heraklion](#)
- [Ierapetra](#)
- [Rethymno](#)



Population	623,065
Time zone	EET/EEST (UTC+2/3)
Area	8,336 km ²
ISO 3166 code	GR-M

Nello scrivere il codice si tengano in considerazione questi vincoli:

- non è possibile usare tabelle di layout, ad eccezione della tabella con i dati riassuntivi mostrata in alto a destra
- non è possibile usare elementi `i`, `b`, `strong`, `em`
- nessun elemento deve contenere l'attributo `style`
- non è possibile usare l'elemento `STYLE` (usare un CSS "esterno")
- le dimensioni esatte di margini e padding non sono rilevanti
- usare URL a piacere per le immagini

3. Javascript (6 punti)

Si riprenda in considerazione l'esercizio HTML della domanda precedente. In una ulteriore evoluzione della pagina, si inseriscano 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*).

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 restituisce un oggetto JSON come segue:

```
{
  "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, si caricano le *numImmagini* fotografie successive a quelle attualmente visualizzate. Si gestisca correttamente il caso in cui *max* non è un multiplo intero di *numImmagini*. Si disattivi il pulsante destro se al momento è visualizzata l'ultima fotografia.
3. Agendo sul pulsante sinistro, si caricano le *numImmagini* fotografie precedenti a quelle attualmente visualizzate. Si gestisca correttamente il caso in cui la *pos* della prima fotografia visualizzata è inferiore a *numImmagini*. Si disattivi il pulsante sinistro se al momento è visualizzata la prima fotografia.
4. 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. Lo scroll è infinito, ovvero quando arriva all'ultima fotografia, viene mostrata in fondo all'elenco la prima fotografia, ri-iniziando lo scrolling da capo. Contemporaneamente vanno attivati/disattivati i pulsanti sinistro e destro secondo necessità. NON è consentito usare librerie di carousel di JQuery, Twitter Bootstrap o altri framework esistenti.

4. Semantic Web (6 punti)

Si consideri la seguente ontologia in Turtle:

```
@prefix : <http://www.unibo.it/ontology/> .

# La classe di tutte le persone
:Person a owl:Class ;
    rdfs:subClassOf owl:Thing .

# La classe di tutte le persone che hanno partecipato ad una qualche conferenza
:Attendant a owl:Class ;
    rdfs:subClassOf :Person .

# La classe di tutte le conferenze
:Conference a owl:Class ;
    rdfs:sbuClassOf owl:Thing .

# La proprietà che permette di specificare un nome a qualsiasi cosa.
:hasName a owl:DatatypeProperty ;
    rdfs:domain owl:Thing ;
    rdfs:range xsd:string .

# La proprietà che permette di collegare un partecipante
# ad una conferenza con la conferenza a cui ha partecipato
:attended a owl:ObjectProperty ;
    rdfs:domain :Attendant ;
    rdfs:range :Conference .
```

Rispondere alle seguenti domande:

1. Utilizzando il formato a scelta tra Turtle e RDF/XML, e senza specificare esplicitamente i prefissi usati, tradurre la seguente frase in RDF utilizzando tutte le classi e proprietà mostrate nell'ontologia: "Miriam Allocca e Francesco Fernandez hanno partecipato alla conferenza Extended Semantic Web Conference 2014".
2. Scrivere una query SPARQL che, se una qualunque istanza `i` della classe `:Attendant` ha partecipato ad una qualunque conferenza, costruisca uno statement RDF che specifichi quello stesso individuo `i` come istanza della classe `:Person`.
3. Supponete di aver caricato nella variabile "graph" l'intera ontologia indicata all'inizio. Utilizzando un linguaggio di programmazione a scelta tra Python e PHP, aggiungere all'ontologia la proprietà `:hasParticipant` come inversa della proprietà `:attended` e salvarla nel file "newgraph.rdf" in un formato RDF a scelta.

5. Teoria (4 punti)

Descrivere i seguenti ruoli delle applicazioni HTTP, evidenziandone le differenze:

1. origin-server
2. proxy
3. gateway
4. tunnel