

Insegnamento di Tecnologie Web

CdS In Informatica

(A.A. 2021-22)

Esame scritto del 16/09/2022

Nome:

Cognome:

Matricola:

Corso di Studi

Solo se diverso da Informatica Triennale

Anno di frequenza

Come specificato nel piano di studi: o "2021-22" oppure "precedente".

Attenzione:

- Questi computer sono limitati ad accedere solo ad alcuni siti: `eo1.unibo.it`, `virtuale.unibo.it`, `developer.mozilla.org` e `getbootstrap.com`. Non funzionano Google, stack overflow, etc.
- *Rispondete solo negli spazi delimitati dai blocchi ```` qui la risposta ````, senza modificarli o eliminarli.*
- *Consegnate solo questo file. Copiate ed incollate dentro agli appositi spazi la vostra risposta per intero.*
- *Potete decidere se inserire il CSS inline nel file HTML o metterlo in un file esterno. Nel secondo caso inserite l'elemento nella posizione corretta e mettete il CSS in un blocco separato.*
- *You can use either English or Italian for your answers.*
- *Per favore, per favore, per favore: nessun errore di ortografia. Questa è un università e non la scuola elementare.*

Domanda #1 - Domande di base (4 punti)

a) HTTP

Che differenza c'è in HTTP tra status code di tipo 4xx e status code di tipo 5xx?

b) Codifica caratteri

Occupa più byte un testo in inglese di 1000 caratteri, un testo in italiano di 1000 caratteri o un testo in cinese di 500 caratteri, se tutti e tre vengono espressi in UTF-8? Perché?

c) Javascript

Cos'è una promessa in Javascript?

d) CSS

Che differenza c'è in CSS tra il selettore “.foo .bar” e il selettore “.foo > .bar”?

Domanda #2 - HTML + CSS (10 punti)

Scrivere il codice HTML e CSS (bootstrap è ammesso solo se importato correttamente nella pagina) della seguente pagina web.

JSON-LD Playground Documentation Developers Specifications Branding

JSON for Linking Data

Data is messy and disconnected. JSON-LD organizes and connects it, creating a better Web.

Linked Data

Linked Data empowers people that publish and use information on the Web. It is a way to create a network of standards-based, machine-readable data across Web sites. It allows an application to start at one piece of Linked Data, and follow embedded links to other places of Linked Data that are hosted on different sites across the Web.

A Simple Example

```
{
  "@context": "https://json-ld.org/contexts/person.jsonld",
  "@id": "http://dbpedia.org/resource/John_Lennon",
  "name": "John Lennon",
  "born": "1940-10-09",
  "spouse": "http://dbpedia.org/resource/Cynthia_Lennon"
}
```

JSON-LD

JSON-LD is a lightweight Linked Data format. It is easy for humans to read and write. It is based on the already successful JSON format and provides a way to help JSON data interoperate at Web-scale. JSON-LD is an ideal data format for programming environments, REST Web services, and unstructured databases such as Apache CouchDB and MongoDB.

Playground

The **JSON-LD Playground** is a web-based JSON-LD viewer and debugger. If you are interested in learning JSON-LD, this tool will be of great help to you. Developers may also use the tool to debug, visualize, and share their JSON-LD markup.

[Launch the JSON-LD Playground](#)

Developers

JSON-LD is available in a number of popular programming environments. Each implementation of JSON-LD listed below is fully conforming to the official JSON-LD specifications.

Javascript jsonld.js 1.1 jsonld-streaming-parser.js 1.1 jsonld-streaming-serializer.js 1.1 rdf-parse.js 1.1	Python PyLD 1.1 RDFLib 1.0	Ruby JSON-LD for RDF.rb 1.1	Go JSON-gold 1.1
Java Titanium 1.1 JSONLD-JAVA 1.0	C# dotheRDF 1.1 json-ld-net 1.0	Erlang / Elixir jsonld-ex 1.0	PHP php-json-ld 1.0 jsonld 1.0
Rust json-ld 1.1 (WIP)	TypeScript jsonld-ldt 1.1 (WIP)		

The **JSON-LD Test Suite** (and **Framing Test Suite**) is useful for validating JSON-LD Processors. Conformance of various processors is documented in the official **implementation report**.

Getting Help / Collaborating

JSON-LD 1.1 was being formally specified by the **W3C JSON-LD Working Group**. Although the Working Group has completed, it remains open as a **Maintenance Group** and may release revised specifications in the future. To participate in this work, please join the **W3C** and then join the **Working Group**. The **Working Group mailing list** is visible to the public and participation in the **specification repositories** is encouraged. Additionally, discussions and exploration continue in the **Community Group** spaces mentioned below.

JSON-LD has been developed by the **W3C JSON-LD Community Group**. It is a **W3C Recommendation** as of 10 January 2014. Participation is open to the public. There is a **JSON-LD GitHub repository**. If you need immediate help, we have a **#json-ld** IRC support channel on [irc.w3.org](#) and a **Gitter channel**. There is also a **JSON-LD mailing list**. All weekly meetings are open to the public, and are **minuted**, **recorded**, and **archived**. Listed below are all of the communication tools that the community uses on a weekly basis.

Code json-ld on GitHub	Meetings Text and audio minutes	Mailing List public-linked-json@w3.org	Chat #json-ld on irc.w3.org Gitter
----------------------------------	---	--	---

Learn More...

This website contains a number of curated video, presentations, tutorials, and documentation about JSON-LD. Assuming you are familiar with JSON, these training materials will help you quickly put the power of JSON-LD into your web development efforts.

[Learn more about JSON-LD](#)

Website content released under a **Creative Commons CC0 Public Domain Dedication** except where an alternate is specified. Part of the **PaySwam** standardization initiative.

Le immagini sono fornite in uno zip scaricabile da EOL. Il codice deve funzionare su Firefox. Può essere usato come base il documento `base.html`. Non è importante essere totalmente precisi con colori e misure, ma essere ragionevolmente attenti alle differenze tra elemento ed elemento. Se il file non si apre correttamente nel browser, il punteggio è 0.

Codice HTML (*ed eventualmente CSS interno*)

Codice CSS (*solo se esterno*)

Domanda #3 - Javascript (10 punti)

Il sito `MyLifeMyCouch.com` permette di creare divani personalizzati combinando forme, materiali e colori diversi in maniera libera. Ogni divano viene consegnato in 4-6 settimane a seconda del materiale e ad un costo di 100-300€ a seconda della dimensione. L'acquisto può avvenire o su quattro o cinque modelli predefiniti dai nomi evocativi, oppure attraverso l'opzione di crearsi un proprio modello. Ogni modello, sia predefinito sia personalizzato, si compone di una selezione di moduli diversi (telaio, chaise-longue, schienale, braccioli, opzione letto, poggiatesta, ecc.), una selezione di materiali (tela, pelle, altri materiali), e una ampia gamma di colori che dipendono dal materiale (non tutti i colori sono disponibili per tutti i materiali).

L'API del sito offre alcuni servizi (tutti i dati, sia in input che in output, sono in JSON):

- `PresetModels`: elenca i modelli predefiniti. Di ogni modello descrive i parametri di composizione, e fornisce anche un breve testo descrittivo (scritto da un redattore dell'azienda), un'immagine, un costo (leggermente più basso del costo ottenibile costruendosi il modello equivalente) e un tempo di spedizione (leggermente inferiore al tempo ottenibile costruendosi il modello equivalente).
- `MakeYourCouch`: crea e denomina un modello personalizzato sulla base di una selezione di componenti, coperture e colori.
- `Buy`: ordina un divano predefinito o personalizzato.

Usate il documento allegato, `base.html`, per fare le vostre prove. Il file contiene, già configurati, un accesso a jquery e a bootstrap. Potete usare i servizi `http://site202100.tw.cs.unibo.it/info` oppure `http://www.fabioitali.it/TW/test/2021/doResponse.php` per effettuare prove e debugging. Basandosi, dove si ritiene, su uno o più framework Javascript a piacere tra quelli illustrati a lezione, si realizzi quanto segue:

Parte I (2 punti) Realizzare il codice HTML per visualizzare la pagina di elenco dei modelli predefiniti. La pagina è inizialmente vuota e carica i dati dal servizio `PresetModels`, visualizza in maniera libera ma appropriata ed interessante ogni informazione presente nei dati ricevuti (includere le immagini ed i prezzi), ha un logo che riporta alla home, altri elementi a vostra discrezione per le varie caratteristiche del sito, e un pulsante "Crea il tuo modello". Inoltre ogni voce della lista di modelli avrà un pulsante "Acquista".

Parte II (5 punti) Usando un framework Javascript a piacere tra quelli visti a lezione, si realizzi la parte client-side del servizio `MakeYourCouch`. Questa deve visualizzare in una pagina nuova un form per creare il proprio divano,

La prima parte della pagina permette di comporre la forma desiderata. Esistono selettori che permettono di crearsi un divano composto di cinque elementi: il telaio, lo schienale, i piedini, i braccioli, la chaise-longue, il poggiatesta, opzione letto. L'utente deve selezionare telaio, schienale e piedini e può scegliere una combinazione a piacere, o anche niente, degli altri componenti. Ad ogni selezione di elementi corrisponde un costo base di produzione (dato dalla somma pura degli elementi scelti) e un numero di mq di materiale di copertura. Si noti che opzione letto e piedini non modificano il numero di mq di copertura.

La seconda parte della pagina permette di scegliere il materiale della copertura, tra tessuto in cotone, lino, lana, seta, velluto, pelle ed eco-pelle. Ogni materiale ha un proprio costo al mq: il prezzo finale dunque deve considerare il prezzo al mq del materiale per il numero di mq richiesti (arrotondato all'intero superiore).

Sulla base del materiale specificato, infine, si potrà scegliere il colore. Ogni materiale porta con sé un elenco diverso di colori e stampe disponibili, mai meno di cinque, mai più di venti. Il sistema mostra sia il nome sia un'immagine associata al colore e/o alla stampa di volta in volta selezionata.

Ogni colore ha un prezzo diverso al mq, e contribuisce in maniera diversa al prezzo finale. Il sistema calcola automaticamente il prezzo del modello finito via via che l'utente esegue delle scelte, e permette di dare un nome creativo ed evocativo al modello appena realizzato. Un pulsante "Acquista" permette di passare al servizio di acquisto.

Parte III (3 punti) Usando un framework Javascript a piacere tra quelli visti a lezione, si realizzi la parte client-side del servizio Acquista. Questa visualizzerà in un pannello separato un form per acquistare uno o più divani di un modello predefinito o di uno personalizzato. La pagina presenta un riepilogo delle caratteristiche del divano che si sta per acquistare, un'immagine (se esiste) e il prezzo scomposto: costo dei componenti, costo della copertura, costo del colore/stampa. Inoltre aggiunge il tempo di spedizione (che dipende dal tessuto prescelto: cotone e lana 4 settimane, pelle e eco-pelle 6 settimane, mentre lino, velluto e seta richiedono 8 settimane) e il costo di spedizione (100€ per modelli a due e tre posti, 200€ per modelli con opzione letto o chaise-longue, 300€ per modelli con sia letto sia chaise-longue).

Il prezzo totale dell'acquisto viene calcolato e mostrato immediatamente. Si aggiunga un campo nome e un campo indirizzo come semplificazione di un form completo di identificazione del cliente, e un pulsante "Procedi" che invoca il servizio buy passandogli tutti i parametri appropriati.

Domanda #4 - Semantic Web (4 punti)

Scrivi in JSON-LD il grafo RDF della seguente frase, poi di' quante triple contiene: «Nadia Stepanova (Buriazia) è una religiosa russa, presidente degli sciamani della Repubblica di Buriazia, la quale è una repubblica della Federazione Russa, la cui popolazione, secondo l'ultimo censimento, è di poco inferiore a un milione di abitanti.»

Domanda #5 - Domanda di accessibilità (5 punti)

Si forniscano due implementazioni di un pulsante che, se premuto, invochi la funzione JavaScript “onButtonPress” definita di seguito. Un’implementazione dovrà usare un elemento HTML nativo, mentre l’altra dovrà utilizzare attributi della specifica WAI-ARIA applicati su un elemento “generico” (ovvero la cui semantica, priva di attributi della specifica WAI-ARIA, non rappresenti un pulsante). Entrambe le implementazioni dovranno essere pienamente accessibili; è possibile trascurare ogni aspetto legato alla visualizzazione grafica dei due elementi.

```
function onButtonPress(event) {  
    console.log(`Handling button press from ${event.target.tagName} element`)  
    alert('Button pressed, great!')  
}
```