

Tecnologie Web (6 CFU)

C.d.L. in Informatica

Compito del 27 luglio 2011

Nome:

Cognome:

Matricola:

Team:

Non è la prima volta che provo questo esame

Ricapitolo: fare qui sotto una croce sul numero di ciascun esercizio risolto. Se lo si consegna su un foglio protocollo, scrivere in maniera chiara il numero dell'esercizio presso la sua soluzione.

Esercizio		Punti	Voto
1	Domande di base	12	
2	HTML	6	
3	XSLT	4	
4	Javascript	6	
5	Teoria	4	
Totale		32	

Domande di base (12 punti)

Rispondere correttamente ad almeno tre delle seguenti domande:

4) Indicare un metodo HTTP 'idempotente' ed uno 'non idempotente'.

HTML (6 punti)

Scrivere il codice XHTML 1.0 Strict (e CSS) di un documento che rappresenti il contenuto di questa immagine:

Dolci estivi

D'estate uno potrebbe vivere di soli dolci.

Dolce	origine	esempio
Gelato	Sicilia	Gelato al limone
Granita	Sicilia	Granita alla mandorla
Caffè shakerato	Lecce	Caffè al ghiaccio

Gelato

Il gelato viene comunemente definito come una "preparazione alimentare ottenuta con una miscela di ingredienti portata allo stato solido o pastoso mediante congelamento e contemporanea agitazione".

Granita

La granita è un dolce freddo al cucchiaio, tipico della Sicilia. Si tratta di un composto liquido semi-congelato preparato con acqua, zucchero e un succo di frutta o altro ingrediente (oltre alla frutta sono più diffusi mandorla, pistacchio, caffè o cacao).

Nello scrivere il codice si tengano in considerazione questi vincoli:

- i bordi delle celle della tabella sono di due colori e di due stili differenti;
- i link che partono dai nomi degli artisti sono collegamenti alle omonime sezioni;
- i link che partono dagli esempio sono collegamenti all'URI costruito partendo da `http://example.org/cerca-ricetta` e aggiungendo nel parametro `dolce` della query-string il nome dell'esempio stesso.

Inoltre,

- nessun elemento deve contenere l'attributo `class` (di conseguenza non si possono usare selettori classe nel codice CSS);
- nessun elemento deve contenere l'attributo `style`;

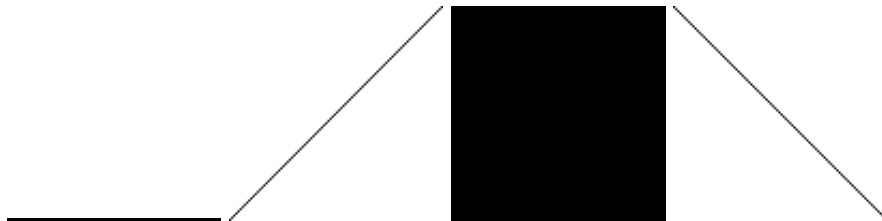
Si consiglia di scrivere tutto il codice CSS in un "file" separato, non in elementi `<style>`

Nota: il DocType di XHTML 1.0 Strict è `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">` e il namespace è `http://www.w3.org/1999/xhtml`.

XSLT (4 punti)

JavaScript (6 punti)

Siano date quattro immagini GIF come in figura:



che rappresentano quattro linee nere rispettivamente orizzontale (1000.gif), diagonale a 45 gradi (1045.gif), verticale (1090.gif) e diagonale a 135 gradi (1135.gif).

Parte 1

Si definisca in Javascript la classe "*segmento che usa immagini in GIF*", in breve `SCUIIG`, costituito dall'immagine di un segmento orizzontale, oppure verticale oppure diagonale dilatata secondo necessità e posizionata in maniera assoluta.

Se ne definisca un costruttore con quattro parametri interi x_1, y_1, x_2, y_2 , tale per cui: se $x_1 = x_2$, allora si dilata secondo necessità l'immagine orizzontale, se $y_1 = y_2$ allora si dilata secondo necessità l'immagine verticale, se $x_1 < x_2$ e $y_1 < y_2$ oppure $x_1 > x_2$ e $y_1 > y_2$ allora si dilata secondo necessità l'immagine diagonale a 45° , altrimenti si dilata secondo necessità l'immagine diagonale a 135° .

Si definisca il metodo `posiziona(x, y)`, che sposta il segmento in modo che l'angolo superiore sinistro dell'immagine sia posto alla posizione assoluta x, y

Si definiscano i metodi `getBoundingBox()`, che restituisce un array di quattro coordinate corrispondenti agli estremi superiore sinistro e inferiore destro del rettangolo che contiene il segmento, `getWidth()`, che restituisce la larghezza della bounding box del segmento (un intero), `getHeight()`, che restituisce l'altezza della bounding box del segmento (un intero), e `getLength()`, che restituisce la lunghezza del segmento (un intero).

Analogamente si definiscano i metodi `setBoundingBox(x1, x2, y1, y2)`, che dilata il segmento in modo che gli estremi superiore sinistro e inferiore destro della bounding box siano quelli specificati nei parametri (cambiando la natura dell'immagine da verticale a orizzontale a diagonale se necessario), `setWidth(w)`, che cambia la larghezza della bounding box del segmento, `setHeight(h)`, che cambia l'altezza della bounding box del segmento, e `setLength(l)`, che cambia la lunghezza del segmento.

Parte 2

Si definisca la classe delle "*figure disegnabili in SVG ma che usano segmenti che usano immagini in GIF*", in breve `FDISMCUSCUIIG`. Si definisca un costruttore che ha come parametro un array di segmenti o figure geometriche più semplici.

Si definiscano, analogamente al caso dei segmenti, i metodi `getBoundingBox()`, che restituisce un array di quattro coordinate corrispondenti agli estremi superiore sinistro e inferiore destro del rettangolo che contiene il segmento (corrispondenti ai min e max, asse per asse, dei bounding box

degli elementi contenuti nella figura), `getWidth()`, che restituisce la larghezza della bounding box della figura (un intero), e `getHeight()`, che restituisce l'altezza della bounding box della figura (un intero).

Si definisca la funzione `show()` che crea un `<div class="imgContainer">...</div>` e al suo interno pone tanti elementi `IMG` quanti sono i segmenti, tutti correttamente posizionati sulla base delle loro bounding box. Si ingrandisca il `DIV` in questione quanto basta per visualizzare l'intera figura (suggerimento.: si usino le dimensioni della bounding box dell'intera figura).

Parte 3

Si definisca infine la classe dei "documenti in JSON contenenti amene figure disegnabili in SVG ma che usano segmenti che usano immagini in GIF", in breve `DIJCAFDISMCUSCUIIG`. Si definisca un metodo `load(uri)` e un metodo `save(uri)` che usando Ajax con il framework preferito tra `ExtJS` e `jQuery` caricano (metodo `GET`) e salva (metodo `PUT`) un documento il cui formato è il seguente:

```
data: [{
  type: 'rect',
  data: [
    [ 0, 0, 0, 100],
    [ 0, 100, 100, 100],
    [ 100, 100, 100, 0],
    [ 100, 0, 0, 0]
  ]
}, {
  type: 'triangle',
  data: [
    [ 0, 100, 50, 0],
    [ 50, 0, 100, 100],
    [ 100, 100, 0, 100]
  ]
}]
```

Infine si scriva il metodo `show(i)` che visualizza ogni figura contenuta nel documento all'interno dell'elemento del documento `HTML` che ha `id` uguale ad `i`.

Si carichi e mostri il documento `casettaincanada.json`

Teoria (4 punti)