

1. Posso utilizzare il CSS per raggruppare tutti gli elementi con una data etichetta all'inizio di un documento HTML? Quali altri strumenti posso utilizzare?

CSS è un linguaggio di presentazione non permette di modificare la struttura e la gerarchia di una DOM. Quindi non è possibile utilizzare CSS per modificare la struttura. In alternativa c'è possibile utilizzare JavaScript. Con JavaScript è possibile invece modificare la struttura del documento. Ad esempio potrei avere un document.getElementById("etichetta") e posso impostare che siano all'inizio. È possibile farlo anche con la CSP.

2. Con CSS, è possibile fare in modo che i caratteri in bold dentro una tabella siano resi in verde? Se sì, in che modo?

Con CSS c'è possibile effettuare un cambio di colore dentro una tabella effettuando:

Table b { color : green; }

3. Posso fare in modo che i primi elementi di ogni lista siano resi in bold? In che modo?

Potrei farlo usando CSS

:first-child { text : bold; }

Potendo trovare gli elementi di una lista che sia ordinata o non ordinata (ul, ol), e ponendo questo il loro testo in bold.

- 4.

```
1. function Employee() {
2.     this.name = "";
3.     this.dept = 'general';
4. }
5. var marc = new Employee()
6. alert(marc.name)

1) di che strumento si tratta? Da cosa si vede?
2) Cosa succede se viene eseguito?
```

1) Lo strumento utilizzato è Developer, ma nel caso del var che indica un solo debolmente tipato, dalla dichiarazione dell'oggetto Employee.

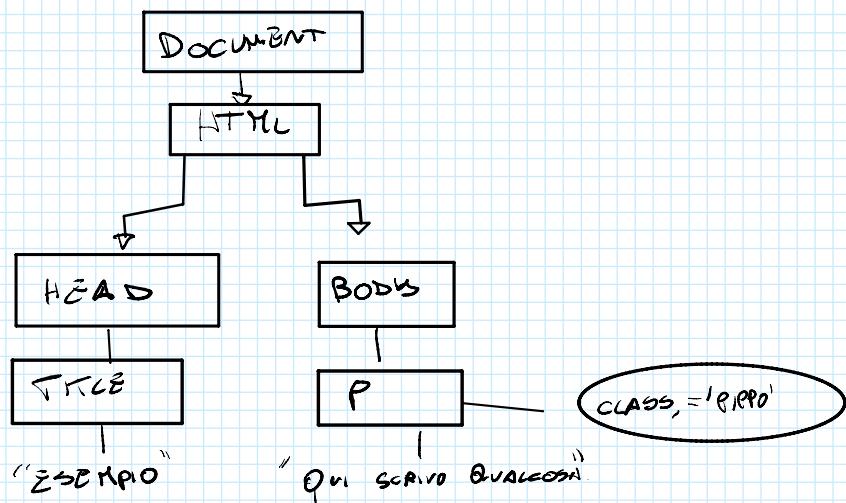
2) Viene comunque compilato ed eseguito e viene mostrato un popup vuoto, poiché nell'istanza

dichiarazione di pareri, delle variazioni dell'offerta e proposte.

2) Vene comitato completamente eseguito e viene mostrato un popup vuoto, poche nell'elenco viene fornita una singola voce.

5. Si disegni un grafico che rappresenti la DOM del seguente documento HTML

1. <html>
2. <head>
3. <title>esempio</title>
4. </head>
5. <body>
6. <p class='pippo'> qui scrivo qualcosa </p>
7. </body>
8. </html>



1. Si dia il codice per una servlet che raccolga in modo anonimo le preferenze rispetto ai tre partiti politici di Paperopoli (PaperoneForPresident, IDirittiDeiCittadini, MarmellataPerTutti), ma che eviti che lo stesso utente esprima più di una preferenza (si ricordi l'anonimia del rilevamento).

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Servlet extends HttpServlet {
    public void service(HttpServletRequest request, HttpServletResponse response) {
        response.setContentType("text/html");
        PrintWriter writer = response.getWriter();
        if ((request.getParameter("Pap") == null) &&
            (request.getParameter("Diritti") == null) &&
            (request.getParameter("Marmellata") == null)) {
            writer.println("No preferences selected");
        } else if ((request.getParameter("Pap") != null) &&
            (request.getParameter("Diritti") != null) &&
            (request.getParameter("Marmellata") != null)) {
            writer.println("Multiple preferences selected");
        } else if ((request.getParameter("Pap") != null) &&
            (request.getParameter("Diritti") == null) &&
            (request.getParameter("Marmellata") == null)) {
            writer.println("Selected PaperoneForPresident");
        } else if ((request.getParameter("Diritti") != null) &&
            (request.getParameter("Pap") == null) &&
            (request.getParameter("Marmellata") == null)) {
            writer.println("Selected IDirittiDeiCittadini");
        } else if ((request.getParameter("Marmellata") != null) &&
            (request.getParameter("Pap") == null) &&
            (request.getParameter("Diritti") == null)) {
            writer.println("Selected MarmellataPerTutti");
        }
    }
}

```

Per reagire in modo anonimo uso session così non ho dettagli dell'utente ma ho dettagli della sessione

Io posso utilizzare un metodo di session tracking, senza però riconoscere le persone, quindi dobbiamo evitare l'autenticazione, ma posso riconoscerlo con esempio dell'IP.

```
String id = request.getParameter("id");
```

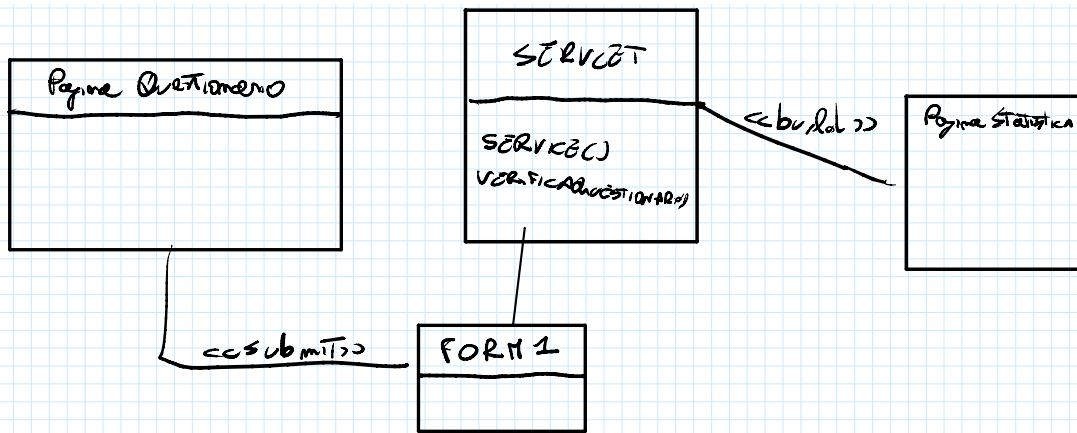
```
Cookie c1 = new Cookie("paperopoli", id);
```

Mi salvo il Cookie e lo passo al client quando il client chiamerà di nuovo la servlet manderà nella url un id univoco che identifica la sessione.

E' possibile farlo anche con form anonime

```
<form hidden=true method Post action="Servelet">.....</form>
```

2. Ricorda l'anonimato
Si dia una site map di un sistema web che propone all'utente un questionario e, una volta riempito, dà le statistiche ottenute fino a quel momento.



3. I campi nascosti: cosa sono, a cosa servono, come funzionano (se ne dettagli bene il funzionamento).

I campi nascosti sono campi che non sono visibili all'utente su una pagina web. Generalmente servono per inviare dati tramite form in modalità anonima o per scambiare dati tra client e server senza che l'utente debba effettuare alcun click. Questa modalità veniva usata principalmente con le pagine dinamiche poiché non c'era modo in precedenza di poter inviare dati al server dal client senza che l'utente non svolgesse manualmente un'operazione. Per avere un campo nascosto banalmente ad esempio una form, grazie ad HTML4 è possibile inserire l'attributo hidden=true ed avremo il campo nascosto.

Vantaggi

- sono supportati da tutti i browser più diffusi;
- non richiedono nessuna particolare funzionalità da parte del server;
- sono anonimi, e non richiedono quindi all'utente di registrarsi.

Svantaggi

- funziona solo quando lo scambio prevede una sequenza di schede da riempire generate dinamicamente;
- fallisce nel caso di errore che provoca la chiusura del browser.

4. Scope delle variabili in JavaScript.

Possiamo definire tipi primitivi di variabili Javascript,
Abbiamo: Undefined, Nullable, Boolean, String, Number

Lo scope delle variabili, o anche ambiente di definizione varia a seconda di dove vengono dichiarate. In javascript è possibile dichiarare variabili globali e locali, le variabili locali sovrascrivono fino al termine della funzione locale la variabile globale (se con lo stesso nome). Le variabili non sono fortemente tipate e vengono dichiarate con il semplice "var". La dichiarazione non è obbligatoria ma consigliabile.

Variabili globali: in un browser Web, vengono tipicamente inizializzate dalle dichiarazioni var eseguite quando il documento viene caricato, e distrutte quando il documento viene eliminato. Tutte le istruzioni di quel documento

possono leggere e scrivere le variabili globali.

Variabili locali: vengono inizializzate all'interno di una funzione (eventualmente anche con una dichiarazione var) e

restano attive fino al termine della funzione.

Gli script possono accedere a metodi e attributi degli oggetti statici: per esempio, tutti gli attributi dell'oggetto Math

agiscono come costanti.

JavaScript è un linguaggio basato sugli oggetti: le caratteristiche di base sia del linguaggio che dell'host sono fornite da oggetti, e un programma JavaScript è un agglomerato di oggetti in comunicazione tra loro. JavaScript è basato sugli eventi.

5. Cosa si intende per servizio web?

Per servizi web intendiamo qualcosa che lavora in ambiente web. Un servizio quindi è una applicazione che permette di far comunicare più applicazioni differenti indipendentemente dal sistema che li fa girare.

I servizi web rappresentano un mezzo standard per interopere tra diverse applicazioni software, eseguiti su piattaforme e/o contesti diversi. Scambio di dati e servizi non solo tra applicazioni diverse, ma addirittura tra imprese diverse.

1. Si dia il codice per una servlet che raccolga in modo anonimo le preferenze rispetto ai tre partiti politici di Paperopoli (PaperoneForPresident, IDirittiDeiCittadini, MarmellataPerTutti), ma che eviti che lo stesso utente esprima più di una preferenza (si ricordi l'anonymità del rilevamento).

```
Import java.*;
Import java.servlet.*;
import javax.servlet.http.*;

Public class Servlet extends HttpServlet {
    Private String array_sessions[];

    Public void init()
    {
        Array_sessions[] = new String ()[1000];
    }

    Public void doGet(HttpServletRequest response, HttpServletResponse request)
    {
        Response.setContentType("txt/html");
        PrintWriter out = Response.getWriter();
        Boolean flag = false;

        String sessionid = request.getPathInfo();
        //Oppure si puo prendere se usiamo i cookie request.getCookie();
        if(sessionid == null) {
            sessionid = generateSessionId();
        }

        For(int i = 0; i < Array_sessions.length(); i++)
        {
            If(array_sessions[i].equals(sessionid))
            {
                Flag = true;
            }
        }
    }
}
```

```
I = array_sessioni.length();  
}  
  
If(flag == false)  
{  
    Array_sessioni[Array_sessioni.length()] = sessionid;  
  
    String scelta_voto = request.getParameterValue("voto");  
    ....  
}  
}  
}
```

A. Si consideri il seguente codice a cui si riferiscono le domande seguenti:

```
GET /www.w3.org/StyleSheets/home.css HTTP/1.1
If-Modified-Since: Tue, 04 Sep 2018 09:40:12 GMT
```

- a) Di che strumento si tratta e quali sono la funzione e le principali caratteristiche di tale strumento.
- b) Si dettagli il funzionamento e le finalità del codice mostrato.

B. Si illustrino le differenze tra:

Il segnato codice è una richiesta GET HTTP scritta in linguaggio HTML. Permette di effettuare una richiesta GET HTTP con versione 1.1 all'indirizzo www... del file home.css

b) If Modified Since c'è una linea d'intestazione che permette al richiedente di verificare se quel determinato file è stato modificato da quella data. In caso肯定 della risposta del server ci sarà anche un body che in questo caso sarà il nuovo file.css altrimenti nulla.

2. Si dettaglino funzionamento e finalità dell'autenticazione basata su HTTP.

HTTP Permette di poter autenticare un utente tramite la sua autenticazione, e' possibile farlo inserendo le credenziali in una finestrella pop-up, altrimenti il server restituisce al client un errore 401 Authorization Required. In tal caso HTTP costruisce la richiesta con un campo nell'intestazione Authorization dove viene rappresentata la chiave di scrittura delle possibili credenziali di accesso, e quindi e' possibile procedere con l'autenticazione sul server.

3. Quali sono le principali innovazioni introdotte da HTML5?

Le principali innovazioni introdotte con HTML5 sono:

Miglioramenti di etichette ed elementi generali;
possibilità di poter salvare dati localmente sempre con coppia nome, valore;
ridurre la necessità di plugins esterni;
Migliorare la gestione degli errori;
Indipendenza dal device;
Ed infine HTML per l'annotazione degli ipertesti che si differenzia dalla costruzione di HTML in sé per se come formattazione del testo.

A. SOAP: a cosa serve e quali sono le sue caratteristiche?

SOAP (Simple Object Access Protocol) permette lo scambio di messaggi. Egli permette di poter inviare e ricevere file XML (o anche diversi) su HTTP. Egli è formato sulla differenza necessaria tra entità e agente, sfrutta le caratteristiche di una WSDL (WebService Definition Language) per poter comunicare con un agent requestor e poter quindi far comunicare due agent (requestor e provider) al fine di poter scambiare messaggi. SOAP principalmente impacchetta e scambia messaggi XML, i quali i formati XML vengono serializzati per poter permettere uno standard di lettura..

1. Cosa è e a cosa serve ECMAScript?
2. Qual è la differenza tra First-party cookies e Third-party cookies?

ECMAScript è un linguaggio che viene utilizzato tramite Javascript per la modifica di DOM. Esso viene utilizzato anche nelle XMLHttpRequest. Questo linguaggio quindi permette la modifica di documenti con dichiarazione W3C.

2. Qual è la differenza tra First-party cookies e Third-party cookies?
3. Cosa si intende per web frameworks, quali sono le sue funzionalità e le principali tipologie.

I cookie first party sono cookie che vengono inviati direttamente dalla pagina la quale l'utente sta visitando, a differenza invece di cookie di terze parti le quali sono siti web che l'utente ha visitato in passato e che vengono inviati da siti presenti e linkabili dal sito che attualmente l'utente sta visitando.

3. Cosa si intende per web frameworks, quali sono le sue funzionalità e le principali tipologie.

Per web frameworks intendiamo frameworks che lavorano in ambiente web. In tal caso i frameworks permettono agli attori di poter sfruttare codice già pronto, specialmente in ambiente server dove la maggior parte delle funzionalità sono ripetitive. Quindi i web frameworks offrono delle funzionalità a chi le implementa senza doverle riscrivere. Non sono librerie ma permettono di offrire uno strumento a supporto dello sviluppo di applicazioni.

I Web Frameworks permettono di implementare correttamente le caratteristiche della OO, inoltre sfruttano il modello MVC basato sull'architettura a tre stati. A differenza della MVC principale i Web Frameworks lavorano sulla cosiddetta web MVC la quale si differenzia dal fatto che tra la Model e la View non ci sia un reale collegamento, essendo che la comunicazione avviene tramite richieste HTTP.

Ultimamente sono stati sviluppati micro stack frameworks, che a differenza dei precedenti si occupano di offrire supporto solo sulle richieste HTTP quindi la parte controller del webMVC.

I web framework nascono dalla consapevolezza che le operazioni svolte da uno sviluppatore con la programmazione lato server sono tipicamente sempre le stesse: connessione alla base di dati, scelta di dettagli quali il content-type della pagina restituita, gestione delle variabili di sessione e altre operazioni simili. Un web framework si fa quindi carico di sistematizzare e di risolvere questi problemi in modo standard, in modo che lo sviluppatore possa concentrarsi esclusivamente sulle caratteristiche significative dell'applicazione che sta sviluppando.

4. Si consideri il seguente pezzo di codice a cui si riferiscono le domande (a) e (b):
- ```
function pippo() {
 var b = new Boolean (false);
 if (b) {
 return 3
 }
 return 0
}
a. Di che strumento si tratta? Si diano le caratteristiche principali di tale strumento.
b. La chiamata pippo () cosa restituisce? Perché?
```

Si tratta di javascript, le caratteristiche principali per designarlo sono: Function alla prima riga, che permette di definire metodi, costruttori e funzioni, la dichiarazione della variabile b con il simbolo "var" che non è fortemente tipata.

B) la chiamata pippo restituisce il valore 0 poiché b viene inizializzato a false ed in tal caso non entra nell'if sottostante e quindi restituisce 0.

2. In una JSP, qual è la differenza tra scriptlet e dichiarazioni?

<% SCR. (T)LET %>

<%! DECLARATION %>

<%= EXPRESSION %>

<%@ DIRECTIVE %>

In una JSP la differenza tra scriptlet e dichiarazioni riguardano:

Una scriptlet viene dichiarata <% scriptlet %> mentre

Dichiarazioni : <%! Declaration %>

La differenza sostanziale e' che una scriptlet esegue codice di scripting, ovvero elementi di scripting per costruire una pagina dinamica come codice css o codice javascript, ma anche produrre commenti. Le dichiarazioni invece permettono di definire variabili e metodi che diventano membri della classe ottenuta dalla traduzione della jsp.

4. Si vuole implementare una pagina web basata su HTML che, senza nessuna azione da parte dell'utente e senza eseguire nulla sul server, al caricamento inserisca in fondo alla pagina una riga con data e ora corrente, il tipo di browser utilizzato e il titolo della pagina:
- quale strumento conviene utilizzare? Quali sono le caratteristiche di questo strumento che lo rendono adatto al task?
  - Si dia il codice necessario.

A) E' possibile utilizzare Javascript, poiche permette di trasformare la DOM e poter quindi modificare un elemento di una pagina nel grafo finale della DOM. Javascript essendo un linguaggio di scripting generalmente utilizzato lato client permette di effettuare modifiche di trasformazione del grafo, quindi lo ritengo opportuno per questa modifica.

Document.appendChild(new Date (sysdatetime));

```
<script type="text/javascript">
function onLoad() {
 var d = new Date();
 var z = d.getHours();
}
var firstVariable = document.createElement("h1");
var secondVariable = document.createTextNode(d);
firstVariable.appendChild(secondVariable);
```

```
document.appendChild(firstVariable);
</script>
```

1 Si consideri un problema in cui l'utente deve riempire un modulo (form). Qual è la differenza tra controllare il contenuto dei campi lato client e lato server? Quali strumenti posso utilizzare nei due casi?

La differenza tra il controllare i dati lato client e lato server è che un linguaggio di scripting che lavora lato client deve verificare il più possibile se i dati inseriti siano conformi a quanto previsto dal server, ovvero se vengono riempiti i moduli form obbligatori, se sono composti in modo corretto (basti pensare ad esempio la mail, dove lato client è possibile verificarne la formattazione di essa tramite una regex); lato server invece quando riceve i campi non verifica se è composta in modo corretto ma verifica se sono presenti i dati nel database (in caso di ricerca) oppure li salva se ha bisogno di salvarli. Questa divisione semplifica di molto il carico di lavoro lato server e permette di avere una comunicazione più veloce tra client e server. Inoltre, i dati vengono elaborati in tempo reale dal browser del client quindi la risposta sarà immediata, ovviamente ciò comporta un carico maggiore sulla macchina client.

Lato Client ad esempio possiamo verificare correttamente tutti i dati con un qualsiasi linguaggio di scripting, come Javascript. Lato Server è possibile farlo in tantissimi modi, possiamo usare pagine dinamiche come le JSP, le Servlet, ma anche un linguaggio di scripting come PHP.

2. Devo implementare un sistema di prenotazioni alberghiere: utilizzerò un web framework o un content management system? Perché?

Per implementare un "sistema" generalmente è consigliabile utilizzare gli strumenti che mette a disposizione un web framework. A differenza di un CMS, il web framework è molto vago, permette di dare libera immaginazione di poter sviluppare una web application come il programmatore desidera. Il CMS invece semplifica di molto lo sviluppo di una web application ma è molto limitato, infatti permette di poter organizzare al meglio e di poter gestire file multimediali, permette di far scambiare messaggi XML e di poter interpretare, permette di avere una **gestione degli accessi** ed un **workflow**. Il tutto però è limitato dal fatto che lo sviluppatore non ha pieno controllo, infatti il tutto è basato su template che lavorano su una base già pronta. Con i webframework invece è possibile inserire tutte le funzionalità che lo sviluppatore desidera, come ad esempio la gestione delle sessioni ... Quindi io credo sia meglio sviluppare su un web framework a meno che non siano funzionalità banali e limitate da poterlo fare con un CMS, in tal caso semplifica di molto lo sviluppo e le tempistiche.

3. Cosa si intende per pattern MVC su web? Che relazione c'è con l'architettura a tre strati?

Per pattern MVC sul web intendiamo una evoluzione WEB del pattern generale MVC. Infatti la principale differenza nel quale è stata fatta questa differenziazione prevede il non collegamento dal MODEL alla VIEW poiché si tratta di una comunicazione indiretta basata su protocollo HTTP. Un pattern MVC quindi è formato da 3 grandi entità:

- 1) Model, che prevede la modellazione dei dati al fine di poterli processare e poterli rendere visibili al cliente, ad esempio effettua il mapping tra il modello dei dati e corrispondente modello oggetti dell'applicazione.
- 2) View, prende i dati per poterli presentare all'utente finale, gestisce la visualizzazione all'utente e quindi organizza il dato finale, ad esempio costruisce la pagina HTML, lavora sui dati XML, JSON...
- 3) Controller, il quale gestisce tutte le operazioni sui dati e si occupa di tutte le operazioni che l'utente non deve vedere, ovvero richieste HTTP, gestione con la base dati al fine di recuperare i dati, gestire la comunicazione con il MODEL dei dati, trasferire dati alla VIEW.

Il pattern MVC è in "relazione" con il modello generale 3 TIER, con Tier che viene inteso un raggruppamento logico di funzionalità. Possiamo intendere il modello 3 Tier in con il suo tier intermedio:

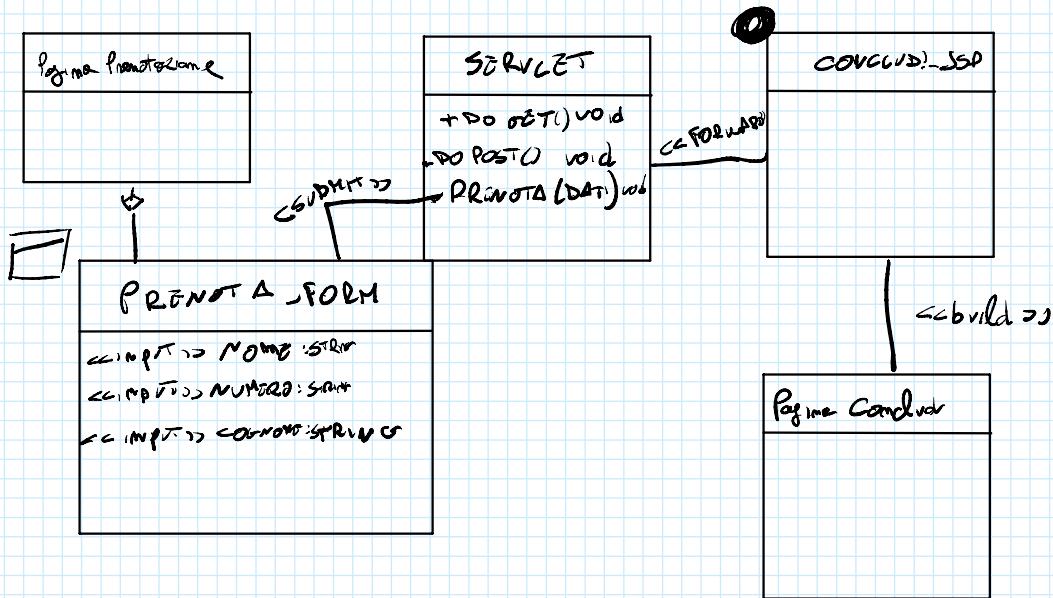
- 1) Business logic: effettua controlli sui dati, verifica che siano affidabili
- 2) Presentation Logic: elabora i dati per presentarli all'utente (HTML, XHTML, XML ecc.)
- 3) Controller logic: elabora le richieste e si procura i dati, da una base dati ecc.

Nella rappresentazione del tier intermedio di una architettura a tre strati possiamo notare come ci sia una sorta di connessione tra entrambe le rappresentazioni.

L'architettura a three tier infine è composta dal Server Web il quale è rappresentato dalle logiche descritte in

precedenza, dall'information source il quale rappresenta (nella maggior parte delle volte una base dati), e da uno user agent che è il cliente finale.

- 4) Si disegni la site map di un semplice sistema di prenotazioni alberghiere, motivando le scelte.



Ho deciso di implementare una servlet che svilupperà una pagina JSP la quale alla fine svilupperà una pagina HTML di conclusione. La pagina Prenotazione quindi prende i dati dall'utente quali NOME, NUMERO, COGNOME, li manda alla servlet, la quale esegue la funzione PRENOTA, elabora i dati, li manda alla pagina JSP la quale svilupperà la pagina HTML finale denominata PaginaConcludi

- 5) Si dia una regola CSS che renda in bold la prima riga di una tabella.

TR : first-child {  
font-weight: bold;  
}

TR: first-child {  
Font-weight: bold;  
}

1. Si abbozzi una JSP che prenda come parametro di ingresso il nome dell'utente e scriva sulla pagina un messaggio del tipo "Ciao <nome>: sei il <xxx>-esimo visitatore da <data>" dove <data> è il giorno in cui la JSP è stata creata (il significato di <xxx> è ovvio).

```
<html>
<head><title>JSP</title></head>
<%
Static int visite = 0;
Static Date datetime = sysDate();
If((sysDate().getday() > datetime.getday()) || (sysdate().getmonth() >
datetime.getmonth()) || (sysdate().getyear() > datetime.getyear()))
{
Datetime = sysDate();
Visite = 0;
}
%>
<body>
<form method="get">
<input type="text" name="nome" value="nome">Nome
<input type="submit" value="put">
</form>
```

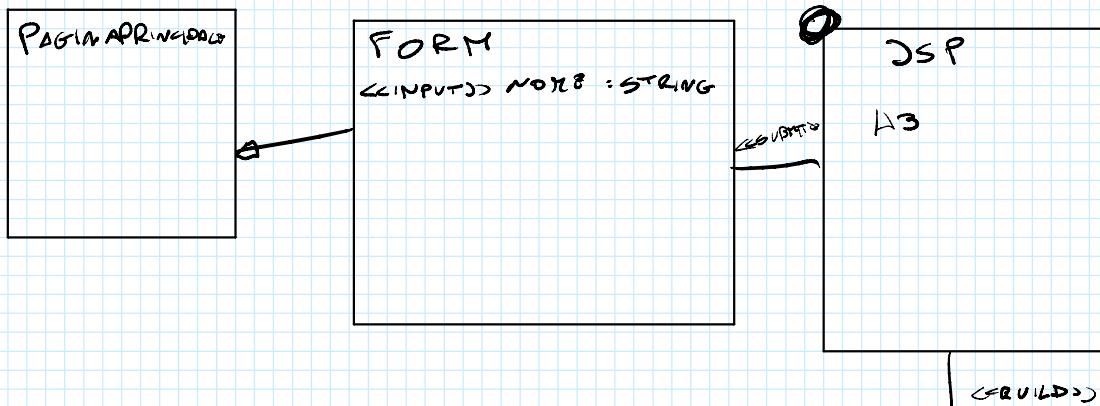
```
<%
visite++;

String nome = request.getParameterValues("nome");
if (nome != null) {
%>
<h3>Ciao <%out.write(nome)%> sei il <%out.write(visite)%> visitatore da <%
out.write(datetime)%></h3>
```

```
<%
}
%>
```

```
</body>
</html>
```

2. Si disegni una site map del sistema che comprende la JSP della domanda precedente.



PARTE PRIMA

4. E' possibile applicare session tracking via variabili di sessione se il client ha disabilitato i cookies?

E' possibile applicare il session tracking via variabili di sessione, poiche le variabili sul server solo l'identificativo di sessione puo essere scambiato tra client e server tramite

- 1) i cookies via DoGet o DoPost
- 2) Riscrittura dell'url tramite DoGet

E' quindi possibile applicare il session tracking sfruttando lato server la funzione DoGet per avere il retrieving dei dati tramite la riscrittura dell'url, con i suoi pregi e difetti.

- Si vuole implementare una pagina web che visualizzi il numero di volte in cui è stata visitata da quello user agent: si proponga una soluzione. Quando viene ripristinato il contatore?

Il contatore che visualizza il numero di volte in cui e' stata visitata la pagina da quello user agent viene calcolato tramite la url che viene presa alla richiesta HTTP che il client fa al server. Quindi, ogni volta che un cliente effettua una richiesta al server viene incrementato il valore salvato sul server che associa nome, valore, quindi viene effettuata un session tracking che permette di associare un numero affianco al nome che in questo caso rappresenta l'id della sessione. Ogni volta che quindi l'utente cambia sessione ad esempio cambiando connessione e computer, il contatore si resetta poiche' la coppia nome, valore non corrisponde a quelle precedenti ed in tal caso il contatore viene ripristinato.

- Qual è la differenza tra struttura logica e struttura fisica di un documento XML?

XML viene rappresentato con elementi e etichette come HTML, essendo un linguaggio di markup anch'esso ogni elemento viene organizzato secondo una struttura logica, in modo gerarchico ovvero a nodi, ogni elemento infatti viene rappresentato in un albero dove ci sono nodi padri (tranne il document) e le foglie che non hanno nodi figli. La struttura logica di XML che viene rappresentata tramite una API denominata DOM, viene tradotta in una corrispondente struttura fisica composta dagli elementi denominati tag.

- Si scriva la DTD per documenti XML in grado di rappresentare una raccolta di ricette con informazioni riguardo il grado di difficoltà della ricetta, il numero di persone per cui si dà la ricetta, la lista degli ingredienti e la procedura da seguire.

#### DTD

```
<!ATTLIST ricetta
gradoDifficoltà CDATA (facile | medio | difficile) "facile"
numeroPersone INT 0
listaIngredienti <!ATTLIST lista CDATA (farina|cioccolata|formaggio) "">
Procedura CDATA ""
>
```

#### XML

```
<ricetta>
 <gradoDifficoltà>difficile</gradoDifficoltà>
 <numeroPersone>10</numeroPersone>
 <listaIngredienti>farina</listaIngredienti>
 <listaIngredienti>formaggio</listaIngredienti>
 <Procedura>Inserire formaggio dopo aggi...</Procedura>
</ricetta>
```

- In che modo AJAX fa uso di XML?

AJAX e' uno strumento web lato client che permette di poter eseguire asincronamente varie operazioni di richiesta dal client al server senza dover ricaricare completamente l'intera pagina del client. Per richiedere dinamicamente cosa il client deve quindi ricaricare AJAX sfrutta XML per lo scambio di informazioni tra Client e Server.

## 5. Cosa sono e come funzionano i namespaces in XML?

I namespaces, detti anche spazio dei nomi in italiano, sono utili per documenti XML che vengono associati alle DTD per poter distinguere elementi con lo stesso nome. Elementi che appartengono ad un determinato padre possono avere lo stesso nome di altri elementi che appartengono ad un altro determinato padre, per rappresentare ciò in una DTD è necessario stabilire i criteri per poter rappresentare tali elementi.

Scopo dei namespaces è qualificare univocamente i nomi degli elementi in modo da evitare ambiguità, anteponendo al nome dell' elemento un prefisso definito precedentemente. Di norma si associa al prefisso un URI web (non sono importanti i dettagli dell' URI in quanto non vengono affatto verificati è semplicemente una stringa). La possibilità di usare questo meccanismo in maniera efficace è garantita dal limitato ambito di scope del prefisso. Infatti il prefisso è valido solo nell' elemento in cui è definito e in tutti i suoi discendenti. Buona norma è dichiarare tutti i namespaces che si vogliono utilizzare nell' elemento radice, in modo che potranno essere richiamati in tutto il documento.

Anche gli attributi di un elemento possono stare in un namespace, con la stessa tecnica del prefisso

Quelli per cui non è specificato nessun prefisso non stanno in nessun namespace ! Anche se l' elemento che li contiene è in un namespace.

```

1. <html>
2. <head>
3. <title>Test</title>
4. </head>
5. <body>
6. <h3>Choose an author:</h3>
7. <form method="get">
8. <input type="checkbox" name="author" value="Tan Ah Teck">Tan
9. <input type="checkbox" name="author" value="Mohd Ali">Ali
10. <input type="checkbox" name="author" value="Kumar">Kumar
11. <input type="submit" value="Query">
12. </form>
13.
14. <%
15. String[] authors = request.getParameterValues("author");
16. if (authors != null) {
17. <%>
18. <h3>You have selected author(s):</h3>
19.
20. <%
21. for (int i = 0; i < authors.length; ++i) {
22. <%> <%= authors[i] %>
23. <%>
24. }
25. <%>
26.
27. <a href=<%= request.getRequestURI() %>>BACK
28. <%
29. }
30. <%>
31. </body>
32. </html>

```

- Di che strumento si tratta? Se ne diano le principali caratteristiche.

In questo codice viene usato linguaggio di markup HTML, lo possiamo notare dai tag di apertura head e body. Alla riga 14 notriamo che viene utilizzato un tag di apertura "<%>" che definisce uno scripting JSP, quindi conosciamo il tipo di pagina, ovvero pagina lato server, inoltre alla riga 15 possiamo notare l'utilizzo di linguaggio Java, lo possiamo notare dalla dichiarazione formalmente tipata di una variabile authors.

## 2. Si dettagli il funzionamento del codice dato sopra.

Il codice in alto, è una pagina lato server che permette di avere 3 input dall'utente in formato checkbox, e che vengono passate le 3 scelte al server. Il server quindi preleva le scelte tramite identificativo della form "author", verifica se non è vuota e modifica dinamicamente la pagina client, aggiungendo una lista ordinata delle scelte che l'utente ha fatto, infine aggiunge un bottone back che riporta alla pagina precedente. La JSP non fa altro che compilare una pagina che prevedere il riepilogo di cosa ha sottomesso l'utente nella form.

## 3. Si costruisca una servlet equivalente.

```

Import java.*;
Import java.servlet.*;
...
Public class Servlet extends HttpServlet {
 Public void init() throws ServletException {
 ...
 }

 Public void doGet(HttpServletRequest request, HttpServletResponse response) throws
 ServletException
 {
 // Set response content type
 response.setContentType("text/html");
 // Actual logic goes here.
 PrintWriter out = response.getWriter();

 String[] authors = request.getParameterValues("author");

 if (authors != null) {
 Out.println("<h3>You have selected author(s):</h3>");
 Out.println("");
 for (int i = 0; i < authors.length; ++i) {
 ...
 }
 Out.println("");
 Out.println("<a href=<%= request.getRequestURI() %>>BACK");
 }
 }
}

```

```
 Out.println("<%= authors[i] %>");
 Out.println("");
 Out.println("<a href=<%= request.getRequestURI() %>">BACK");

}

}

}
```

#### QUESTIONE

4. Si dettagli bene il funzionamento dei cookies. A cosa servono?

Un cookie e' un meccanismo di session tracking. Esso e' composto da una coppia di chiavi nome, valore; oltre all'host e path della url per poter richiedere il session tracking ed infine la expiration date che definisce una data di fine del cookie. Un cookie e' un componente del browser che permette di poter essere inviato nella comunicazione tramite HTTP dal client al server, e permette al server di avere un controllo indiretto sulla sessione del client. Essendo HTTP senza stato, non e' possibile lato server poter salvare la sessione attuale del client, e poter quindi offrire vari vantaggi, come il mantenimento della sessione, precaricare dati interessanti al client. Lo scopo principale del cookie era quella di poter appunto scambiare dati per mantenere le informazioni sulla sessione. All'inizio di una prima comunicazione tra un determinato client e server, il server crea il cookie che poi verra inviato al client, il quale avra cura di salvarlo e poterlo reinviare ad una chiamata successiva del client al server.

Successivamente con l'avvenire delle nuove tecnologie e del Web Analytics, i cookie sono stati reinterpretati per poter non solo recuperare dati necessari al server, ma poter imparare ogni singolo utente e poter quindi recuperare dati per poter proporre siti pubblicitari in base a cosa il client avesse visitato. E' stata fatta quindi una netta distinzione tra cookie di prime party, **first party cookie** e cookie di terze parti **third party cookie**. I cookie di prime parti sono i cookie necessari ed interni al server il quale il client sta richiedendo informazioni, quindi i dati che vengono forniti tramite cookie sono rivolti al server principale. A differenza invece dei cookie di terze parti che sono dati che vengono indirizzati ad altri server i quali provvedono in seguito a produrre pubblicita mirata al client.

#### QUESTIONE

6. Cosa si intende per Web Framework?
- Per web Framework intendiamo un Framework che banalmente si appoggia sul web. Un framework e' uno strumento che mette a disposizione agli sviluppatori vari strumenti per semplificare di molto lo sviluppo di una applicazione. Un web framework quindi permette di facilitare lo sviluppo di web application, agevolando ad esempio lato server la connessione con il database, oppure implementando framework come React che sviluppa lo scambio di messaggi tra server e client.

#### QUESTIONE

7. Cosa si intende per parser a DOM e parser ad eventi?

Un parser e' un traduttore. Generalmente vengono usati parser per documenti di markup come XML, XHTML, HTML ecc. I principali parser sono appunto parser a DOM e parser ad eventi. In XML ad esempio e' possibile poter tradurre tramite il modello logico un file XML per poter avere il suo SCHEMA tramite la DOM, e poter definire quindi un albero corrispondente all'intero documento XML fornito alla DOM. Questo parser ha pregi e difetti, ovvero e' molto lento per quanto riguarda documenti di elevate dimensioni ma e' molto utile quando un documento e' piccolo. A differenza del parser ad eventi invece, rende disponibile all'applicazione il frammento di dati che corrisponde ad un singolo "evento" nel documento di ingresso. Ad esempio possiamo evidenziare le etichette di apertura dei vari elementi, i contenuti testuali, quindi si tratta di una visione frammentaria del documento, e non per la sua interezza.

1. Quali sono le principali caratteristiche di JavaScript?

Javascript e' un linguaggio di scripting e' possibile utilizzarlo sia lato client che lato server. E' un linguaggio interpretato quindi viene compilata ed eseguita ogni singola istruzione in fase di runtime, inoltre e' debolmente tipato quindi e' possibile dichiarare le variabili con un semplice var ed associera automaticamente il tipo a seconda del valore inserito nella variabile.

Ci sono alcuni tipi primitivi come:

Null, Number, String, Boolean, Undefined.

Javascript e' basato sugli oggetti e non orientato agli oggetti, infatti e' possibile dichiarare funzioni con l'acronimo function che possono essere usate come costruttori in fase di instanziazione di un oggetto con l'acronimo new.

Javascript non permette una funzione con n parametri e la sua chiamata con m parametri, con m <> n, inoltre javascript e' basato su eventi.

Essendo un linguaggio di scripting permette di modificare il documento sfruttando la DOM, ed inoltre permette una gestione dell'input inserito dall'utente, nonché verificare correttamente tutti i dati. Permette di modificare la pagina in modo dinamico.

In Javascript l'ereditarietà e' sviluppata sottoforma di prototipi, infatti ogni classe puo avere un prototipo padre la quale eredita tutte le informazioni come variabili e funzioni, a sua volta il padre puo avere un prototipo padre e si forma una catena di prototipi.

2. Si implementino due classi JavaScript, *dipendente* e *meccanico*, in cui *dipendente* ha nome, cognome, matricola, e *meccanico* è una sottoclasse di dipendente e ha anche un macchinario a cui fa riferimento. Entrambe le classi hanno un metodo che stampa tutte le loro caratteristiche.

Si noti che il linguaggio adottato nella formulazione è quello normalmente utilizzato nell'orientamento agli oggetti e si specifichi in quali casi e per quali ragioni può risultare inadeguato nel caso considerato.

```
Function dipendente(nome, cognome, matricola)
{
 This.nome = nome;
 This.cognome = cognome;
 This.matricola = matricola;
}
```

```
Function meccanico(macchinario)
{
 This.macchinario = macchinario;
}
```

```
Meccanico.prototype = new dipendente;
```

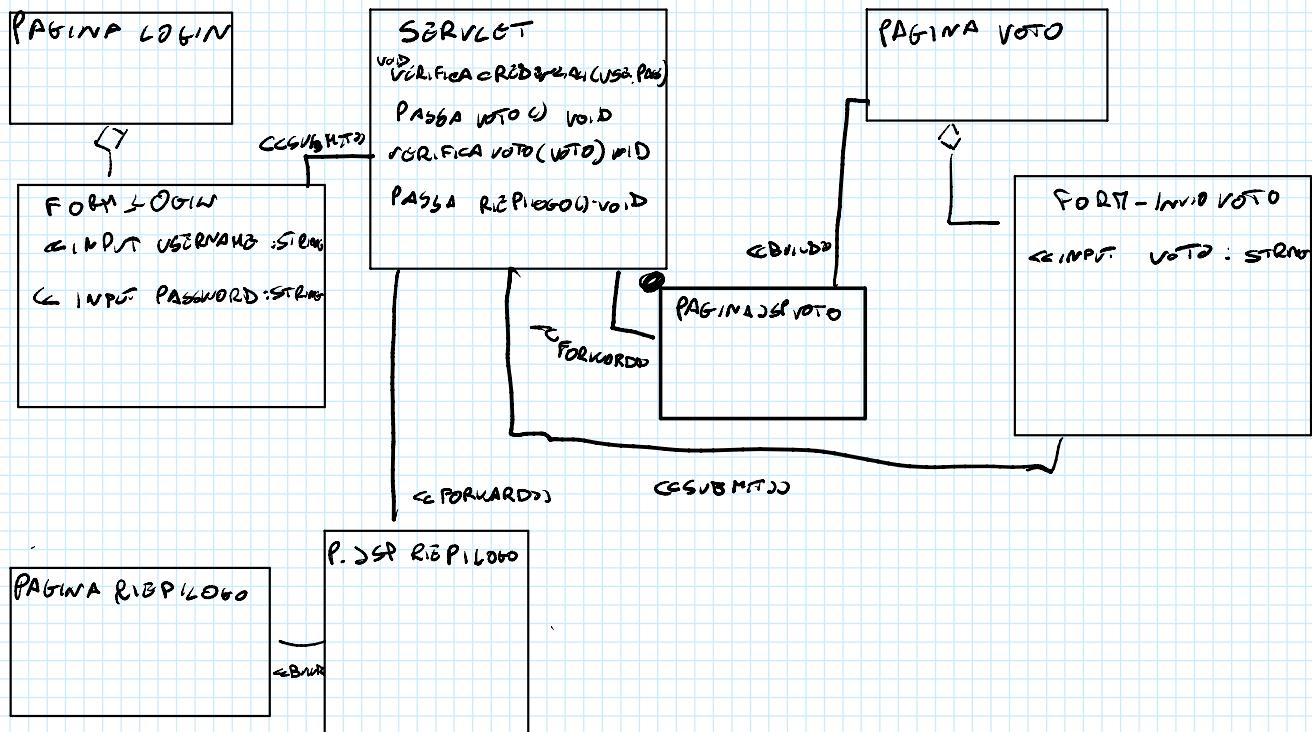
```
Dipendente.prototype.stampa = function () {
 alert(this.nome.toString() + "\n" + this.cognome + "\n" +
 this.matricola);
}
```

```
Meccanico.prototype.stampa = function () {
 Meccanico.prototype.stampa();
 alert(this.macchinario.toString());
}
```

```
Var dip = new dipendente("mario", "rossi", "N86");
Var mec = new meccanico("mario", "rossi", "N86", "...");
```

Essendo Javascript un linguaggio non orientato agli oggetti ma basato sugli oggetti, non prevede l'utilizzo di costruttori esplicativi, pertanto dichiarare classi ed oggetti bisogna utilizzare l'acronimo function che e' ambigua. Essendo che anche la gestione dell'ereditarietà e' gestita mediante prototype e' anch'esso complicato se si vuole definire singolarmente una funzione specifica per ogni classe ereditata. In questi casi ad esempio e' molto più conveniente utilizzare un linguaggio completamente orientato agli oggetti poiché permette di avere una visione migliore dell'insieme.

3. Si disegni la site map per un sistema di votazione on-line. Occorre prevedere almeno una fase di login, di voto e di presentazione dei risultati a fine voto.



4. Un'ente di statistica predisponde un questionario da presentare via web ad un campione piuttosto numeroso per rilevamenti di suo interesse. Il questionario è composto da 500 domande divise in 20 gruppi da 25 domande ciascuna, che ovviamente non possono essere presentate in una sola schermata. Quale tipo di session tracking vi sembra adeguato a questo sistema? Dettagliarne il funzionamento.

Ci sono numerosi sistemi di session tracking. Io penso che il migliore per questo ambito sia sfruttare le variabili di sessione. E' possibile sfruttare i cookie oppure la riscrittura dell'url a seconda di come si vuole implementare lato server. E' possibile quindi salvare ad esempio lo stato delle domande risposte dall'utente, associare ad esempio ad ogni domanda la risposta già presente, in caso l'utente dovesse disconnettersi per mille motivi. La sessione deve quindi essere salvata localmente e' possibile farlo sfruttando quindi i cookie che salvano, effettuando una richiesta periodica al server, i dati dell'utente che sta svolgendo il test. Ovviamente i cookie hanno un numero limitato per contenere informazioni, per aumentarne il raggio e' possibile quindi sfruttare entrambe le tecnologie per poter avere ad esempio le prime 10 domande sui cookie e le restanti su riscrittura dell'url per poter garantire il salvataggio della sessione correttamente.

#### 7. Cosa si intende per SOAP?

Per SOAP Si intende Simple Object Access Protocol e' un protocollo che permette lo scambio di messaggi XML tra un client ed un server basato sul protocollo HTTP. SOAP permette di effettuare una serializzazione di un determinato dato che deve essere spedito al server e di inviarlo, alla ricezione il server potra usare un parser per convertire il file XML e poter quindi ricevere i dati che necessita.

SOAP viene utilizzato principalmente per lo scambio di messaggi tra un agent requester ed agent provider, sfruttando una WSDL Web Service Definition Language che descrive la semantica da utilizzare per lo scambio di messaggi ed avere anche un endpoint dove spedire i messaggi.

Si consideri il seguente pezzo di codice a cui si riferiscono le prime 4 domande:

```
function Person(name, age) {
 this.name = name;
 this.age = age;
}
var you = new Person('You', 24);
```

1. Di quale strumento si tratta? Da cosa si capisce? Cosa fa?

Si tratta di Javascript lo si capisce dalla dichiarazione della funzione (in questo caso costruttore) con l'acronimo "function", dell'assenza dell'assegnazione del tipo ad una variabile, infatti basta dichiararla con var.

Il pezzo di codice dichiara un oggetto di nome you di tipo Person con due attributi:

Nome = You  
Eta = 24

2. Voglio aggiungere a you una proprietà *altezza*: come posso fare?

E' possibile farlo aggiungendo la seguente riga

You.altezza = ?;

3. Voglio aggiungere a tutti gli oggetti creati da *Person* una proprietà *altezza*: come posso fare senza modificare la funzione?

E' possibile farlo aggiungendo la seguente riga:

Person.prototype.altezza = 1.60;

4. Ho un documento XML composto da una lista di elementi di etichetta *film*, ciascuno dei quali ha dei sottoelementi, uno dei quali, di etichetta *regista*, contiene appunto il nome del regista. Voglio trasformarlo in un documento in cui la lista di tutti i registi precede la lista dei film. Posso farlo con CSS? In caso di risposta affermativa si indichi la soluzione. In caso di risposta negativa, si indichino le possibili soluzioni, sviluppandone una.

CSS e' un linguaggio di Presentazione non e' come XSL che permette di poter trasformare l'albero di documento. Quindi con CSS non e' possibile spostare etichette prima o dopo di altre.

5. Si presenti STRUTS.

Cos'è ApacheStruts 2?

è un web framework basato sul cosiddetto modello 2 (basato su pattern MVC per il web). Per la sua modularità e per la sua organizzazione, questo modello è più facile in sviluppo, testing e manutenzione. Apache struts 2 fa uso di JSP ma senza codice Java all'interno di quest'ultimo. La business logic viene implementata da classi Java dette action classes ed inoltre viene fornito un expression language con cui fare accesso agli oggetti del model delle JSP.

6. I campi nascosti: di cosa si tratta e come funzionano.

I campi nascosti sono una possibile soluzione per il session tracking. Infatti vengono utilizzati solamente quando l'utente sfrutta le form e quindi su pagine dinamiche. Vengono implementate alcune form con attributo hidden=true che vengono nascoste alla vista dell'utente, queste form permettono di inviare anonimamente al server le informazioni sulla sessione attuale.

### Vantaggi

- sono supportati da tutti i browser più diffusi;
- non richiedono nessuna particolare funzionalità da parte del server;
- sono anonimi, e non richiedono quindi all'utente di registrarsi.

### Svantaggi

- funziona solo quando lo scambio prevede una sequenza di schede da riempire generate dinamicamente;
- fallisce nel caso di errore che provoca la chiusura del browser.

## 7. Cosa si intende per Java bean.

Per Java Bean intendiamo un supporto utile lato server. Ha avuto il suo successo nella fase di avvento delle JSP le quali permettono di avere una piccola porzione di codice dinamica e la restante che resta statica, poiché il codice Java che si andava a scrivere non era completo e non potevano essere instanziate le classi ecco che nascono le Java Bean. Le Java Bean sono delle classi java che permettono di essere instanziate una sola volta e che mantengono il loro valore fino alla terminazione della servlet. Hanno caratteristiche particolari, ovvero ogni classe Bean deve avere un costruttore vuoto, deve avere metodi per poter richiedere l'istanza di un oggetto e ogni attributo deve essere impostato singolarmente. In questo modo si mantiene la filosofia Java dello sviluppo orientato ad oggetti, e' possibile sfruttare tutte le caratteristiche di ereditarietà di incapsulamento e permette alle JSP di poter organizzare oggetti nel miglior modo possibile.

### Cosa sono le JavaBeans?

Sono delle classi Java pensate per essere facilmente riutilizzate e organizzate per la costruzione delle applicazioni. Queste classi incapsulano più oggetti in un singolo oggetto (il bean), cosicché tali oggetti vengono passati sottoforma di un oggetto. Ogni classe può essere una JavaBean purché soddisfi alcuni requisiti come: la classe deve avere un costruttore senza argomenti, le sue proprietà devono essere accessibili utilizzando metodi get e set e deve essere serializzabile.

1. Si abbozzi una JSP che prenda come parametro di ingresso il nome dell'utente e scriva sulla pagina un messaggio del tipo "Ciao <nome>: sei il <xxx>-esimo visitatore da <data>" dove <data> è il giorno in cui la JSP è stata creata (il significato di <xxx> è ovvio).

```
<%@ page import="java.io.* ,java.util.*" %>
```

```
<html>
```

```
<head>
```

```
<title>Main JSP,</title>
```

```
</head>
```

```
<body>
```

```
<form action = "main.jsp" method = "GET">
 Username: <input type = "text" name = "username">
 <input type = "submit" value = "Submit" />
</form>
```

PAGINA HTML CHIAMA LA JSP

```
<%
```

```
Integer visite = (Integer)application.getAttribute("visit");
```

```
Date data = (Date)application.getAttribute("dat");
```

```
if(visite == null)
```

```
{
```

```
 Visite = new Integer(0);
```

```
}
```

```
If(data == null)
```

```
{
```

```
 Data = Sysdate();
```

```
}
```

```
If(sysdate().getDay() > data.getDay() || sysdate().getMonth() > data.getMonth() || sysdate().getYear() > data.getYear())
```

```
{
```

```
 Visite = 0;
```

```
 Data = sysdate();
```

```
}
```

```
application.setAttribute("visit", visite);
```

```
application.setAttribute("dat", data);
```

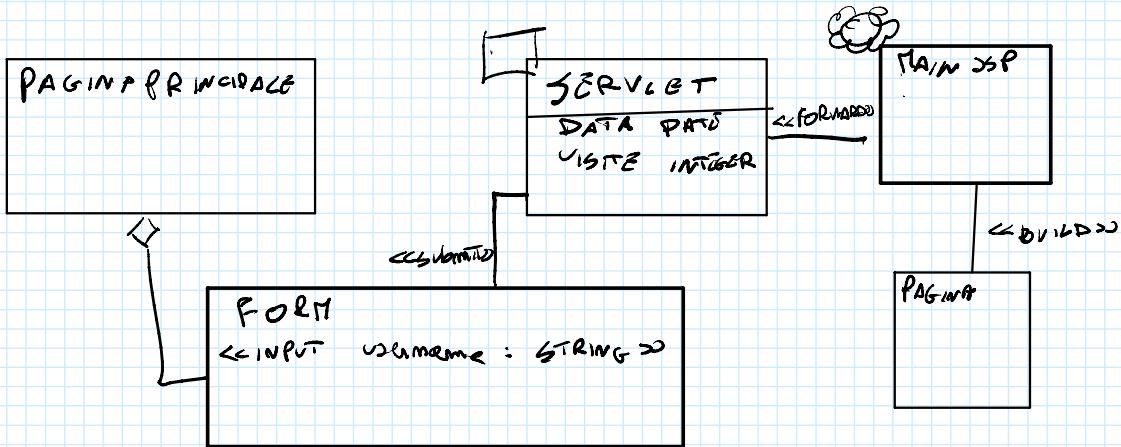
```
%>
```

```
<h2>Ciao <%= request.getParameter("username"); %>: sei il <%= visite %> da <%= data %></h2>
```

</body>

</html>

2. Si disegni una site map del sistema che comprende la JSP della domanda precedente.



### 1. Come funziona il meccanismo dell'autenticazione basato su HTTP?

Il meccanismo di autenticazione basato su HTTP e' un meccanismo di session tracking che sfrutta l'autenticazione manuale dell'utente. Generalmente quando un client HTTP si connette ad un server HTTP egli puo ricevere un messaggio di ritorno nella status line 401 Authorization Required, in quel caso la pagina HTML dovrebbe predisporre un meccanismo (in genere si crea un popup) per poter fare autenticare l'utente ed inviare nella richiesta HTTP le credenziali di accesso tramite il campo authentication.

La servlet che verra eseguita sul server potra quindi recuperare i dati di accesso tramite il metodo getRemoteUser().

#### Vantaggi

- Facile da implementare: il server protegge determinate pagine e può poi identificare ogni client (usando ad esempio il metodo getRemoteUser() per le servlet).
- Il meccanismo delle autenticazioni funziona anche se l'utente accede al sito da macchine diverse.
- Continua a funzionare anche se l'utente esce dal sito o addirittura esce dal browser prima di tornare a concludere la transazione.

#### Svantaggi

- La richiesta di registrazione si adatta solo a determinate situazioni (transizioni commerciali, informazioni sensibili); in tutti gli altri casi occorre un session tracking anonimo.
- (Meno importante) Un utente può, in ogni momento, mantenere attiva una sola sessione dato un sito.

### 3. Si consideri il seguente pezzo di codice:

```
HTTP/1.1 301 Moved Permanently
Date: Sun, 18 Oct 2009 13:19:15 GMT
Server: Apache/2.2.14 (Win32)
Location: http://127.0.0.1:8000/testdir/
Content-Length: 238
Content-Type: text/html; charset=iso-8859-1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
...
</head><body>
</body></html>
```

- Di che cosa si tratta?

Possiamo notare come questa sia una richiesta HTTP, lo possiamo notare dall'intestazione che comprende la versione del protocollo utilizzato, il codice di richiesta, la data di messaggio. In basso possiamo trovare un body della richiesta con la pagina HTML.

- Si commenti la linea 4.

La linea 4 rappresenta la locazione del server, infatti quell'indirizzo rappresenta l'url del server per poter effettuare la richiesta, possiamo notare come la pagina che ha recuperato si trovi sotto una folder root testdir

### 4. In cosa differiscono scriptlet, espressioni e dichiarazioni nelle JSP?

Nelle JSP le scriptlet eseguono codice di scripting, o anche codice che rende la pagina dinamica, quindi potremo scrivere <% scriptlet %> ed all interno dei bracket possiamo far eseguire codice Java.

Le espressioni permettono di dichiarare una espressione java che verra poi convertita in Stringa. Ad esempio possiamo

chiamare una funzione che ci restituisce un oggetto. Si dichiarano: <%= expression %>

Le dichiarazioni invece permettono di dichiarare variabili e/o funzioni in Java. <%! Declaration %>

## 5. Cos'è e a cosa serve la DOM?

La DOM ( Document Object Model ) e' un API (Application Programming Interface) e' ormai uno standard di utilizzo poiche' permette di costruire un albero formato da nodi detto anche a boschetto che definisce un modello del documento che deve essere analizzato. In sostanza la DOM permette di analizzare un documento, di formare una struttura ad albero basata su quel documento. Ha la caratteristica che non e' vincolato al linguaggio e/o alla macchina, quindi e' una API completamente versatile. Il linguaggio che poi lavorera sulla DOM avra la possibilita di accedere alle diverse parti del documento analizzato tramite ID. la DOM genera un albero che e' basato sulla gerarchia di contenimento.

## 6. Perché si dice che JavaScript è object-based?

JavaScript e' un linguaggio di scripting basato su eventi e su oggetti. A differenza di Java che e' un linguaggio completamente orientato agli Oggetti, Javascript e' basato su oggetti poiche' la sua struttura non e' vincolata dall'utilizzo degli oggetti. Basti pensare che e' permesso un utilizzo quasi strutturato, e che per definire classi di oggetti segue l'utilizzo dell'acronimo function che serve a definire una funzione. E' comunque basato sugli oggetti il che permette di instanziare un oggetto con l'acronimo new, ma perde tutte le funzionalita di un linguaggio orientato agli oggetti, come l'ereditarieta', il polimorfismo, l'overriding, l'overloading. Ecc. L'ereditarietà e' stata data una differente versione, ovvero sussegue l'utilizzo dei prototipi che permette di avere una sorta di sottoclassi alle superclassi. Ma e' piu una catena di classi che una vera e propria ereditarietà.

## 1. Quali sono i tipi di dato in JavaScript?

In Javascript esistono tipi di dato primitivi come:

Undefined, che definisce un valore indefinito;

Null valore nullo

Number, valore numerico che puo essere con virgola oppure no

String, stringa di caratteri

Boolean, valore booleano

Poi c'e' il tipo Object che definisce l'oggetto.

## 2. In JavaScript, posso aggiungere un nuovo attributo ad un oggetto già creato? Con quale effetto?

In javascript e' possibile aggiungere un attributo ad un oggetto già creato. Ad esempio se ho la classe Persona che ha come attributi nome e cognome, e' possibile ad esempio preso l'oggetto di nome p di Persona, effettuare la seguente riga

```
P.eta = 10;
```

In questo caso solamente l'oggetto P di Persona avrà il campo eta. Se si vuole aggiungere alla classe Persona l'attributo p bisognerà farlo aggiungendolo tramite prototype.

```
Persona.prototype.eta = 10;
```

In questo modo, qualsiasi oggetto di Persona avrà l'attributo eta.

## 3. Cosa si intende per oggetti impliciti? Si faccia qualche esempio col relativo ambito di definizione.

Per oggetti impliciti intendiamo nelle JSP oggetti che sono stati dichiarati a seconda nel contesto nel linguaggio JAVA.

Ad esempio abbiamo nelle pagine JSP bisogno di poter salvare nel contesto application una variabile counter che salvi il numero di accessi a quella pagina, e' possibile farlo accedendo all'oggetto隐式 Application.setAttribute(...); In questo modo salveremo nel Context della pagina JSP, quindi fino a quando la Servlet che eseguirà la pagina sarà attiva, la JSP potrà accedere ai valori nel Context.

La JSP è un tramite per le Servlet, in questo modo gli oggetti impliciti permettono di avere alcuni oggetti che lavorano direttamente sulle Serlvet e che sono state già definite in fase di instanziazione della JSP.

## 4. Si abbozzi una servlet che prenda in ingresso il nome dell'utente e scriva in uscita il messaggio "Buongiorno <nome>: sei il visitatore numero <n> ", dove n è il numero di visite al sito fino a quel momento.

```
Import Java.*;
Import Java.servlet.*;
```

```
Public static Date data = null;
Public static int visite = null;
```

```
Public class Servlet extends HttpServlet
```

```
{
```

```
 Public void init()
```

```
{
```

```
}
```

```

Public void doGet(HttpServletRequest request, HttpServletResponse response{

 If(data == null)
 {
 Data = sysdate();
 Visite = 0;
 }
 Else If(sysdate().getDay() > data.getDay() || sysdate().getMonth() > data.getMonth() || sysdate().getYear() >
 data.getYear())
 {
 Visite = 0;
 Data = sysdate();
 }

 Response.setContentType("text/html");

 PrintWriter out = response.getWriter();

 String username = Response.getParameter("username");

 Out.println("<h2>Buongiorno :" + username + " Sei il visitatore numero" + visite + "</h2>");

 }

}

```

### 5. Cosa si intende per elementi e per etichette in XML?

XML sta per eXtensible Markup Language: si tratta dunque di un linguaggio di markup come HTML, ma con la caratteristica di essere estendibile, nel senso che le etichette non sono prefissate. Le etichette vengono definite come attributi di un certo oggetto, dove il nome di questo oggetto e' denominato dall'elemento principale.

```

<rubrica>
 <nome>Mario</nome>
 <cognome>Rossi</COGNOME>
</rubrica>

```

In XML le informazioni specifiche di un'applicazione sono contenute all'interno di "tag", marcati da parentesi <>, che descrivono il contenuto di un documento. Ogni tag definisce un tipo di elemento e, delimitando con tag ogni singolo dato, siamo in grado di comprenderne la struttura anche se non conosciamo l'applicazione che l'ha generata.

Ad esempio con la DOM possiamo analizzare che ogni nodo della DOM corrisponde ad un elemento del file.

### 6. Cos'è e a cosa serve WebUML di Conallen? Quali sono le sue parti principali?

Il WebUML di Conallen e' una visione dell'insieme del web che permette di modellare le pagine web. Ogni pagina che sia statica o dinamica viene modellata come un componente, vengono descritte le relazioni che intercorrono tra loro dando origine ad una mappa del sito (site map). Come prima definizione Conallen ha stabilito di differenziare le pagine client e le pagine server. Poiche' c'e' una netta differenza dello sviluppo lato client e sviluppo lato server. Il comportamento di una pagina Web sul server è completamente diverso che sul client: può accedere alle risorse del server (basi di dati, file system, ...) nel primo caso; al browser e alle sue funzionalità nel secondo. Le pagine lato server quindi permettono di costruire (build) le pagine lato client. La differenza sostanziale e' che appunto permette di avere una visione d'insieme corretta dello sviluppo di un sito senza andare ad analizzare le interfacce grafiche e le tecnologie nel dettaglio.

7. Posso usare un CMS per sviluppare un'applicazione JAVA? Che strumenti mi offre?

CMS (Content Management System) permette di fornire agli attori (sviluppatori web) un template per poter sviluppare una web application in modo rapido e veloce. Infatti un CMS permette di velocizzare la creazione di una web application che però presenta alcuni limiti in termini di funzionalità e di sviluppo. I CMS quindi hanno lo scopo di gestire workflows e lo sviluppo di contenuti in ambienti di lavoro collaborativo e vengono quindi progettati per semplificare la pubblicazione di documenti su web. Con un CMS quindi non è possibile sviluppare una applicazione Java poiché il tutto è gestito tramite WEB. Essendo un CMS una RIA (Rich Internet Application) è basata principalmente su scripting lato client, quindi non viene proprio analizzata la questione server. Essendo Java un linguaggio prettamente lato server, non si può fare.

Un CMS fornisce un supporto retrieval dei documenti, permette il controllo degli accessi definendo diverse categorie di utenti con diversi profili e permessi di accesso; definisce un workflow di supporto al lavoro in collaborazione.

1. Come è composta una risposta HTTP?

Una risposta HTTP e' composta da:

Una status line, che comprende l'hostname url, il path,

Un codice di status di risposta

Infine se vengono descritte informazioni sulla sessione permette di riconoscere tramite If-Modified-Since

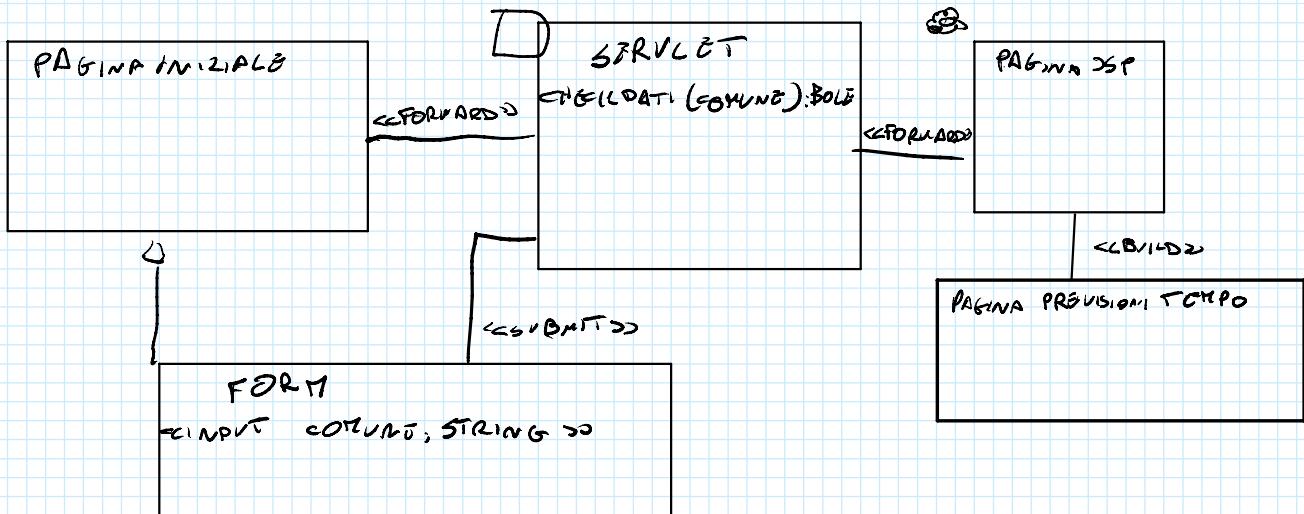
Ed infine c'e' il body che puo essere opzionale.

3. In quali formati una servlet può produrre l'output? In che modo?

Una servlet puo produrre l'output in vari modi. Ad esempio puo scrivere nella response di ritorno al client tramite l'uso di un PrintWriter sulla HttpServletResponse, oppure puo impostare tramite il session tracking valori su cookies. Ovviamente ci sono i classici metodi come inviarlo in modalita Post o get, in get e' possibile riscrivere sulla url di risposta.

I formati rimangono sempre stringhe, infatti anche nelle JSP e' possibile farlo usando l'expression che traduce automaticamente l'output in stringa.

4. Si abbozzi la site map di un'applicazione web in cui l'utente completa una form col nome di un comune italiano e ottiene in risposta le previsioni del tempo. Nel caso in cui il comune inserito non esista, viene riproposta la form iniziale.



5. Qual è la differenza tra bot, spider e crawlers?

Sono tutti elementi che permettono la scoperta di siti web. Un bot infatti permette di scoprire siti web secondo una regola ben precisa, quindi analizza solamente i dati che a lui interessa.

Uno spider invece analizza moltissime pagine web scelte in modo dinamico.

Un crawler permette di indicizzare molte pagine web secondo regole ben precise, e' presente anche un file robot.txt che permette di stabilire i siti a cui deve accedere con determinate regole. Permette di evitare trappole che ingannerebbero un normale spider. Insomma e' uno spider evoluto.

6. Qual è la differenza tra architettura three-tiers e architettura multi-tiers? Nelle applicazioni web, quando si adotta l'una e quando l'altra?

La three tiers architecture e' l'architettura standard di una web application. Infatti veniva presa in considerazione in passato,

adesso ovviamente gli attori sviluppano una loro architettura più complessa per poter analizzare e dividere ulteriormente in micro tier e differenziare maggiormente il carico complessivo.

La three tier architecture è formata quindi da un information source, da un server web e da uno user Agent. Nella architettura multi tiers invece possiamo trovare ad esempio al posto di un information source, molti più server web.

E' possibile adottare la three tiers architecture quando si parla di sistemi legacy e che non diversificano molto dallo sviluppo standard di una web application. A differenza invece di microservizi i quali necessitano di dividere in varie fasi il carico complessivo di lavoro, in questo modo una architettura a microservizi completamente indipendente deve intervenire a molti livelli.

## 7. Come funziona il meccanismo di AJAX?

AJAX Asynchronous JavaScript and XML, permette di poter sviluppare applicazioni web in cui le pagine vengono ricaricate in modo dinamico ed asincrono. Generalmente quando si vuole richiedere un certo valore di dati dal client al server, il client effettua una richiesta al server il quale risponde con i valori di dati tramite XML o altri documenti, la pagina client quindi viene ricaricata affinché possa caricare questi nuovi dati richiesti. Con AJAX invece è permesso poter ricaricare dinamicamente la pagina permettendo di ricaricare singole parti quasi invisibili all'occhio dell'utente. Permette quindi di eseguire aggiornamenti rapidi ed incrementali senza ricaricare l'intera pagina.

AJAX permette di programmare un'interazione asincrona rispetto alle azioni dell'utente, dove una richiesta HTTP viene preparata e inoltrata solo quando risulta necessario per l'applicazione nel suo complesso.

AJAX permette di richiedere al server una richiesta asincrona inviata tramite XML, il server quindi risponderà alla richiesta come una normale operazione, AJAX una volta ricevuta la risposta in formato XML provvederà a scompattarla e ad analizzarla ed adattarla al meglio in modo asincrono alla pagina del client. AJAX sfrutta il meccanismo della XMLHttpRequest per lo scambio asincrono dei dati tra client e server.

### 1. Come funziona il meccanismo di cache e quali supporti deve avere da HTTP?

Il meccanismo di cache deve essere previsto ed attivato nel browser altrimenti non è possibile permettere di raccogliere informazioni riguardo la sessione. La cache viene utilizzata da parte di un browser per permettere di ricaricare facilmente una pagina già visitata, e di avere precaricati in memoria. Perché un meccanismo di cache possa venir adottato proficuamente, occorre che il client sia in grado di controllare se la pagina richiesta dall'utente è stata modificata oppure se quella che è stata salvata in cache è ancora valida senza riscaricare la pagina. Per farlo, HTTP prevede il campo **If-Modified-Since** da inserire nell'intestazione della richiesta e il codice 304 Not Modified da inserire nella status line della risposta. Potremo quindi riaccedere al sito chiedendo di scaricare la pagina solo se è stata modificata. In questo modo il client non deve scaricare ogni volta la pagina dal server.

### 3. Cosa si intende per JavaBeans e come si integrano con le JSP? anche l'altra su JavaBeans precedente

Per JavaBeans si intendono classi Java con determinate caratteristiche. Sono classi che possono essere richiamate nelle JSP, essendo che il codice JSP è limitato e non sfrutterebbe la sua potenzialità del linguaggio Java ovvero Orientato agli oggetti, si è pensato di sfruttare alcune classi Java con il nome di Beans. Le classi devono avere un costruttore vuoto e devono avere un metodo che restituisca l'istanza, che la classe sia quindi serializable e che ogni attributo abbia i getter e setter. In questo modo anche le pagine JSP possono dichiarare classi e instanziare oggetti, semplicemente chiamando il nome della classe.getInstance.

### 4. Cosa si intende per aritmetica dei puntatori e perché è importante rispetto a Java?

In Java non è possibile utilizzare un puntatore poiché va contro le sue regole di sicurezza. Purtroppo ci sono attori che riescono a generare virus tramite l'aritmetica dei puntatori, accedendo a zone della memoria che non potrebbero accedere e quindi corrompendo il sistema o peggio prelevando dati che non sono autorizzati. Java quindi non prevede questa possibilità limitando il più possibile l'aritmetica dei puntatori ed assegnando uno spazio di memoria riservato tramite l'utilizzo delle JVM che permettono di eseguire programmi in una sandbox che è come se fosse una macchina a parte. In questo modo ogni singolo processo di Java è racchiuso nel suo spazio dedicato.

### 5. Con CSS posso scrivere in italico tutti e soli i paragrafi che seguono immediatamente un titolo di tipo <H1> o <H2>? Se la risposta è positiva, si abbozzi la soluzione, se è negativa, si indichi quali strumenti andranno utilizzati.

E' possibile usare CSS per scrivere in italico tutti e soli i paragrafi che seguono immediatamente un titolo.

Ad esempio possiamo definire

```
H1 + P
{
 font-style:italic;
}
```

```
H2 + P
{
 font-style:italic;
}
```

### 7. Si scriva una DTD che considera un elenco di camere d'albergo: ciascuna di esse sarà caratterizzata dal numero di occupanti, il prezzo e l'eventuale presenza di uno o più balconi che possono affacciare sulla piscina o sulla spiaggia.

```
<!ATTLIST camere
 NumeroOccupanti CDATA #REQUIRED
 PREZZO CDATA #REQUIRED
 Balcone balc #REQUIRED
 >

<!ATTLIST BALC
 BALCONE CDATA #REQUIRED
 POSTO CDATA #REQUIRED
 >
```

## 6. Come funziona l'ereditarietà in JavaScript?

Javascript e' un linguaggio di scripting basato sugli oggetti e su eventi. Ciò comporta che essendo che non è un linguaggio orientato agli oggetti sviluppa l'ereditarietà in modo diverso. Javascript sfrutta la prototipazione per definire una catena di ereditarietà. Infatti una classe che vuole estendere una classe padre lo deve aggiungere come prototipo, in questo modo la classe figlio eredita tutte le funzioni e gli attributi della classe padre. La classe padre a sua volta può estendere tramite prototipo un'ulteriore classe formando così una catena di prototipi.

- Si abbozzi una JSP che stampi il numero di richieste ricevute dal server per quella specifica JSP.

```
<%@ page import="java.io.* ,java.util.*" %>
<html>

<head>
 <title> MAIN JSP </title>

</head>

<body>

<%
 Integer visite = (Integer)application.getAttribute("visit");
 If(visite == null)
 {
 Visite = new Integer(0);
 }

 Visite++;
 Application.setAttribute("visit", visite);

 Out.println("<h2>Numero Visite: " + visite + "</h2>");
%>

</body>

</html>
```

- Come funziona e a cosa serve l'autenticazione basata su HTTP?

L'autenticazione basata su HTTP e' prevista dal browser nei casi in cui si prova ad accedere tramite una richiesta HTTP ad un server che richiede l'autenticazione, il client quindi ricevera nella response nella status line un codice 401 Authorization Required che significa che il server si aspetta una richiesta di autenticazione. A quel punto il browser generalmente puo aprire un popup per richiedere all'utente di inserire le credenziali da inviare al server. Il client quindi invierà nella request nell'intestazione un campo authentication con le credenziali inserite dal client. A quel punto il server verifichera se le credenziali sono idonee all'accesso.

- Di cosa fa parte, come funziona e a cosa serve il campo If-Modified-Since?

Il campo IF-Modified-Since e' un campo di intestazione del protocollo HTTP, permette di avere un sistema di caching basato sul browser che permette di capire se la pagina e' stata modificata da una certa data specifica. Quindi nel caso in cui ad esempio l'utente provi a ricaricare una pagina, il client invia al server una richiesta http con il campo if-modified-since, e nel campo dato verrà aggiunta la data di ultimo caricamento della pagina, a quel punto il server riceve la richiesta, nel caso in cui la pagina non sia stata modificata restituisce un semplice status 200, nell'eventualità il client dovesse ricaricare la pagina il server invierà nel body anche il documento della pagina da ricaricare. E' stato utilizzato questo metodo per evitare che il client dovesse ricaricare ogni volta la pagina quando eseguiva una richiesta periodica al server.

- Nelle variabili di sessione è necessario capire se lo user agent ha abilitato i cookies: come si fa?

Nelle variabili di sessione e' possibile effettuare lo scambio tra client e server attraverso due metodi principali: Cookie e riscrittura della url. Ovviamente il primo caso e' quello che ha preso piede molto facilmente e che si reputa il migliore

per molti aspetti. C'e' un ma ovvero il browser puo scegliere di non abilitare i cookie in tal caso si sfrutta il secondo metodo. Da lato server e' molto piu complicato capire se il browser ha abilitato i cookie, ad esempio e' possibile settare i cookie tramite la funzione setcookie() e riconoscere ad una prossima chiamata se il cookie viene passato, oppure e' possibile farlo tramite javascript per riconoscere se un browser ha i cookie abilitati e nel caso non fosse abilitato si imposta dell'intestazione un cookie fittizio che fa capire al server che il browser non ha i cookie abilitati.

### 5. Si confrontino gli attributi *name* e *id*? Di che strumento stiamo parlando?

Gli attributi name ed id sono due attributi del linguaggio HTML, generalmente vengono utilizzati per definire valori ad un elemento. L'attributo name e' stato deprecato in molte occasioni poiche' in precedenza veniva utilizzato per referenziare un elemento, da HTML4 e' stato sostituito con l'attributo ID il quale puo essere univoco e non ci possono essere duplicati per avere un'albero del documento della DOM il piu pulito possibile.

### 6. Si confrontino web services e servlet.

I web services e le servlet sono cose completamente diverse. Le servlet sono applicazioni scritte in Java per la modellazione e la creazione di pagine web lato Server, oltre che alla gestione del back-end e della comunicazione con il database nonche alla gestione dei dati che il server invia al client. I web services invece sono applicazioni che girano nel mondo web e permettono di far comunicare un provider ed un requester per mettere in comunicazioni client e server. Egli sfrutta i linguaggi WSDL per il pattern di comunicazione ed invia tramite SOAP messaggi XML serializzati tra provider e requester.

### 7. Cos'e e a cosa serve un CMS?

Un CMS e' un Content Management System, permette di sviluppare una web application sfruttando template e sviluppando maggiormente il codice lato client. Infatti un CMS e' una RIA ovvero una Rich Internet Application poiche' si basa maggiormente sullo sviluppo lato client. Un CMS permette di poter essere eseguito su browser e di poter essere sviluppato interamente da browser, facilita di molto la creazione di una web application ma e' molto limitata con le sue funzionalita. Le sue caratteristiche principali sono: Gestione degli accessi ai controlli e il workflow. I CMS vengono utilizzati maggiormente per modellazione di documenti e di presentazione come PDF, file multimediali o altro.

### 8. Come si definisce e come funziona un namespace in XML?

Un namespace in XML denominato anche spazio dei nomi permette di identificare maggiormente gli elementi delle varie etichette. In XML purtroppo gli elementi di un oggetto possono avere lo stesso nome rispetto ad altri elementi di altri oggetti, a quel punto quando il parser XML che provvedera ad effettuare il parser DOM, l'albero finale non potra essere fatto poiche' generera errore. A quel punto il namespace entra in campo per identificare questi elementi e aggiornare il loro nome aggiungendo la url completa del path, permettendo quindi di essere riconosciuto univocamente e di poter generare in fase di parsing un albero corretto. E' possibile fare il tutto aggiungendo ad esempio il path URN per identificare univocamente un oggetto.

Si consideri il seguente documento a cui si riferiscono le domande dalla 1 alla 5:

```
HEAD /www.unina.it HTTP/1.1
Host:
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/68.0.3440.106 Safari/537.36
Accept: /*
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,it;q=0.8
Cookie: __utmz=64676715.1536063129.1.1.utmcsr=(direct)
```

1. Di che strumento si tratta e quali sono le caratteristiche generali di questo strumento?

Si tratta di una chiamata da parte di un client al server. In questo momento il server ha ricevuto una richiesta da parte di un client. E' possibile notare nell'intestazione la request line composta dalla head che rappresenta il path, la versione e il protocollo usato, il tipo di host, e possiamo notare che ha un cookie impostato quindi vuol dire che ha già una sessione in questa richiesta.

4. Ci può essere un'ulteriore parte di questo tipo di documento? Con quali caratteristiche e quali funzionalità?

Dubito ci sia bisogno di un ulteriore parte di questo documento ma ciò non toglie che è possibile inviarlo al server. In questo caso forse si intende la parte body che non è obbligatoria.

5. La linea 8 supporta un particolare meccanismo: a cosa serve tale meccanismo e come funziona?

I cookie permettono di avere un session tracking da parte del server. In pratica questo meccanismo permette di salvare una sessione oppure dati dell'utente per poter ridurre i tempi di ricarica e di gestione tra client e server. Infatti il protocollo HTTP che viene usato da questa richiesta è senza stato e quindi un server non può salvare una sessione di un determinato client, questo ovviamente porta a dei vantaggi ma anche ad alcuni svantaggi come ad esempio in fase di crash o di problemi del client, il client deve ricaricare la pagina e se la pagina fosse stata dinamica in quel caso l'utente avrebbe perso tutti i suoi dati. Il session tracking quindi permette di salvare i dati della sessione, i cookie sono un meccanismo per effettuare ciò. In un cookie vengono salvati dati come pathname e url; nome e valore e tempo di scadenza del cookie. In questo modo il server può decifrare il cookie e prelevare le informazioni necessarie a restituire una pagina al client completa.

7. Quali sono le caratteristiche di Java che lo rendono adatto alle applicazioni web?

Java è un linguaggio sicuro, ovvero esegue ogni sua esecuzione in una sandbox che permette di allocare spazio riservato di memoria, oltre che a veicolare il tutto c'è la sua JVM. La maggior parte delle applicazioni scritte in linguaggio C sono vulnerabili agli occhi degli hacker poiché riescono ad accedere a spazio di memoria non consentito e quindi riescono a prelevare dati o a corrompere sistemi, Java invece non permette questo e quindi l'aritmetica dei puntatori non è prevista. Inoltre Java ha un garbage collector che elimina spazio di memoria allocato dinamicamente nello HEAP quando non è più indicizzato. Java poi è un linguaggio fortemente tipato quindi è molto fiscale sul controllo dei tipi e cerca di prevedere ogni errore in fase di compilazione. Java infine è un linguaggio orientato agli oggetti.

8. Cosa si intende per architettura orientata ai servizi in ambito web? Su cosa è basata?

L'architettura a three tier ad esempio è una architettura orientata ai servizi in ambito web e permette di descrivere la struttura di una applicazione, la decomposizione in componenti, le loro interfacce e interazioni. Per tier intendiamo un raggruppamento logico di funzionalità. Le applicazioni web più semplici sono a three tier ovvero: Information Source, Server Web, User Agent. Molte volte però le architetture possono avere multi tier e quindi ad esempio avere più server web al posto di un information source. In quel caso vengono usate maggiormente dai microservizi. L'architettura a three tier quindi focalizziamoci sul server Web che è composto da tre unità logiche ovvero il controller logic che rappresenta la costruzione del dato e si procura i dati da presentare al client, la presentation logic dove il dato viene modellato e reso disponibile al client ad esempio costruendo una pagina html, ed infine la business logic che assicura che i dati siano affidabili prima di usarli.

E 14/06/2018

domenica 19 giugno 2022 16:29

```
<%@page import="DBManager.DBManager"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
<head>
<%!
 /* Connessione al database */
 private DBManager dbm;
 private ResultSet rs;
 public void init(){
 /* Instanzio la connessione al database */
 this.dbm = new DBManager((ServletContext) getServletContext());
 }
 /**
 * Funzione che aggiunge un sensore al database
 */
 private void aggiungiSensore(String operazione, HttpServletRequest req)
 throws Exception{
 ...
}
 /**
 * Funzione che cancella un sensore dal database
 */
 private void cancellaSensore(String operazione, HttpServletRequest req)
 throws Exception{
 ...
}
 /**
 * Funzione per la modifica di un inquinante
 */
 private void modificaInquinante(String operazione, HttpServletRequest req)
 throws Exception{
 ...
}
</head>
<body>
```

1. Di che strumento si tratta e quali sono le caratteristiche generali di questo strumento?

Lo strumento utilizzato e' JSP lato server, infatti lo possiamo notare dall'uso di direttive all'inizio, nella prima riga infatti viene importata una pagina manager per il database, nella seconda viene importato il content type di una pagina html. Poi nello head possiamo notare come ci sia una declaration che descrive la funzionalita di connessione al DB e vengono dichiarate alcune funzioni. Ovviamente il linguaggio nelle JSP e' Java.

4. La parte di codice compresa tra la linea 5 e la linea 34 cosa rappresenta? Si commenti, tenendo conto che l'indicazione ... indica una parte di codice non riportato, ma che non modifica la struttura generale del codice. In particolare, quando vengono messe in esecuzione le istruzioni riportate?

La parte di codice rappresentano delle funzioni che rispettivamente aggiungono, cancellano e modificano dati dal database. In questo caso viene inizializzata la connessione al database viene dichiarato un database e vengono effettuate operazioni riguardo aggiunta e cancellazione di un sensore alla volta; e della modifica di un inquinante. Possiamo notare come le funzioni hanno tutti una stringa operazione che penso sia stata presa in precedenza dall'utente. Le funzioni verranno eseguite nel body del documento poiche in questo momento la JSP le sta solamente dichiarando e quindi non possono essere richiamate nell'head.

5. Cosa si intende per HTTP e che funzione ha?

HTTP HyperText Transfer Protocol e' un protocollo basato su TCP/IP che permette lo scambio di comunicazioni tra un client ed un server. Egli permette di scambiare messaggi tra due endpoint. Possiamo quindi definire due principali agent, abbiamo il provider ed il requestor.

HTTP permette di definire una intestazione del messaggio da inviare che e' composta in questo modo

La richiesta deve essere formata in : Request line: PATH, URL, PROTOCOL, VERSION PROTOCOL  
MORE INFORMATION,  
CODE REQUEST

La status e' uguale solamente che cambia la Status Line, Codice di risposta  
body

8. Cosa si intende per RDF e per cosa viene impiegato?

RDF detto anche Resource Description Framework, permette di avere una analisi ontologica dei dati sfruttando l'XML Come sintassi di interscambio. Egli permette di descrivere le risorse per rappresentare i dati per il web semantico. Lo scopo di RDF e' quello di rappresentare l'informazione in modo che sia facile da leggere ed elaborare via calcolatore, non per rappresentarla all'utente umano. In questo modo indicizzando ulteriormente e' possibile riconoscere le pagine web in fase di ricerca, come sta già effettuando Google da un bel po di tempo. RDF e' a stretto contatto con le basi di dati infatti tramite XML permette di

poter salvare direttamente dati nella base dati, tramite anche gli statement, i predicati e gli oggetti permette di definire un suo namespace di XML per poter identificare le risorse e poterle immagazzinare.

```

public class HelloWorld extends HttpServlet {
 private String message;
 public void init() throws ServletException {
 message = "Hello World";
 }
 public void doGet(HttpServletRequest request, HttpServletResponse response)
 throws ServletException, IOException {
 response.setContentType("text/html");
 PrintWriter out = response.getWriter();
 out.println("<h1>" + message + "</h1>");
 }
}

```

1. Di che strumento si tratta e quali sono le caratteristiche generali di questo strumento?

Si tratta di una Servlet, infatti lo possiamo notare dal linguaggio utilizzato che è java, possibile notarlo dall'incapsulamento delle public, della definizione delle variabili fortemente tipate. Le caratteristiche generali è che permette la comunicazione tra client e server, in questo momento la servlet sta recependo le informazioni del client e le sta elaborando generando nella funzione doget una pagina html con un messaggio in un tag h1.

## 2) Cos'è HTTPServlet?

HTTPServlet specifica il tipo di servlet su quale protocollo viene utilizzato, in questo caso HTTP. Infatti la classe servlet estende la classe astratta HTTPServlet che permette di ereditare tutte le funzioni obbligatorie per permettere lo sviluppo di una servlet Java.

3. Si descriva nel dettaglio l'esecuzione del codice, includendo in quale situazione viene messo in esecuzione.

Il codice prende una stringa di nome messaggio che viene inizializzata ogni volta che la servlet viene inizializzata a Hello World, poi la medesima stringa messaggio viene aggiunta ad un tag h1 generando quindi una pagina html con un solo tag h1 con il messaggio all'interno. La servlet viene messa in esecuzione quando viene richiamata da una form dalla pagina chiamante e nella form ci sia la chiamata a questa servlet, in tal caso la servlet viene eseguita e mostra il messaggio in una nuova pagina html buildata.

4. Quali altri metodi avrebbero potuto venir implementati e con quali finalità?

Si sarebbe potuto implementare un metodo doPost nell'eventualità la chiamata alla servlet fosse stata di tipo Post e sarebbe stato anche meglio essendo che comunque non vengono passati valori durante la chiamata e quindi tra client e server non ci sarebbero stati parametri inviati.

5. Si scriva una JSP del tutto equivalente al codice dato.

```
<%@ page import="java.io.* ,java.util.*" %>
<html>
```

```

<head>
 <title> MAIN JSP </title>
 <%
 String message = "Hello World";
 >
</head>
<body>
 <%

```

```
 Out.println("<h1>" + message + "</h1>");
%>
```

```
</body>
```

```
</html>
```

```

<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
 $cookie_name = "user";
 $cookie_value = "John Doe";
 setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
} else {
 echo "Cookie " . $cookie_name . " is set!
";
 echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

<?php
setcookie("user", "", time() - 3600);
?>

</body>
</html>

```

1. Di che strumento si tratta e a cosa serve?

Si tratta di PHP lo possiamo notare dalla apertura di un tag di script. PHP e' un linguaggio di scripting lato server, permette di eseguire codice lato server per la modifica di una pagina server, e' molto utile per effettuare collegamenti con il DB e altro, la sua sintassi inoltre facilita di molto lo sviluppo. E' un linguaggio orientato ad oggetti, non effettua ereditarietà e encapsulamento se non in PHP5.

In questo caso il codice permette di controllare se il cookie inviato dal client non e' impostato, verificando nella variabile globale \$\_COOKIE, in caso non fosse impostato, lo inizializza associando al nome un "user" e al valore del cookie un "John Doe", infine utilizza la funzione setcookie che lo impone nella risposta al client e impone la sua durata ad 1 giorno.

In caso in cui il cookie fosse già settato quando il client lo invia, allora semplicemente lo mostra nella pagina HTML visualizzando il nome ed il valore.

Successivamente poi impone un ulteriore cookie con nome = user e valore nullo impostando un timer negativo.

3. Si discutano finalità e funzionamento dei cookies.

I cookies nascono per avere informazioni sulla sessione di un utente. Il suo scopo principale era quello di salvare con una coppia nome valore le informazioni riguardo la sessione di un utente che permettesse di ridurre le tempistiche di caricamento da parte del server e di fornire in tempo 0 le risorse al client. Successivamente i cookies sono stati trattati a finalità di marketing e di indicizzazione, inoltre vengono usati anche per prelevare dati. Esistono quindi due tipologie principali di cookie: Cookie di prime parti che equivalgono al server che sta eseguendo la pagina che il client sta visitando e cookie di terze parti per varie finalità di sponsor o altro.

5. Analogie e differenze tra XML e HTML.

XML e HTML solo entrambi linguaggi di markup. HTML e' un linguaggio per la visualizzazione di dati, non permette di modificare gli elementi e di costruire oggetti, a differenza di XML che e' un linguaggio per modellare i dati e quindi permette di modificare le informazioni seguendo uno schema che definisce i dati. In tal caso quindi hanno due scopi diversi, HTML viene usato maggiormente per la visualizzazione in un browser avendo tag predefiniti e quindi rendendolo del tutto interpretabile, a differenza invece di XML che viene usato per lo scambio di messaggi poiché essendo del tutto dinamico permette di essere modellato a seconda della tipologia di messaggio da mandare.

HTML per l'annotazione degli ipertesti; va notato che mentre all'inizio lo scopo dell'HTML era quello di dare una definizione abbastanza precisa della formattazione del testo, nel corso degli anni cresce sempre di più la necessità di separare le indicazioni di formattazione, da quelle più legate ai contenuti del testo. I vantaggi che si ottengono con questa separazione sono sia di portabilità che di manutenibilità.

6. Ciclo di vita di una servlet.

Una servlet viene definita e chiamata da una richiesta nel protocollo che venga utilizzato, parte in generale con la sua creazione con il metodo INIT che definisce l'inizializzazione della servlet e quindi anche la sua costruzione, dopodiché ci sono metodi come doGet, doPost, Service che permettono di instanziare un thread dedicato ad ogni chiamata ed essere quindi disponibile alla risposta. Quando terminano solo e tutte le richieste di quella determinata servlet vengono distrutte con il metodo destroy().

7. Voglio costruire un foglio di stile che mi renda in verde tutti gli elementi di una lista non ordinata, e lasci nel colore di default tutto il resto del testo.

E' possibile farlo con CSS. CSS e' un linguaggio di presentazione che permette di poter modificare gli attributi di ogni elemento di un documento che sia XML o HTML. Non permette di trasformare il documento in se e quindi di modificare la struttura.

Con CSS quindi e' possibile eseguire la seguente riga:

```
UL : {
 color: green;
}
```

8. Cosa si intende per WSDL? In che contesto?

Per WSDL intendiamo un WebServiceDefinitionLanguage, ovvero un linguaggio che definisce un rapporto tra un agent requestor e un agent provider. Nel contesto di scoperta di un servizio web e' possibile sfruttare varie caratteristiche che permettono di far scoprire ad un determinato requestor un determinato provider. Qui entrano in gioco differenti elementi come:

SOAP (Simple Object Access Protocol) permette lo scambio di messaggi in formato XML serializzato.

WSDL, definisce il linguaggio per poter mettere in comunicazione i due agent e quindi definisce le regole (rules) per comunicare UDDI Universal Description, Discovery and Integrationm, permette di registrare la descrizione di un servizio attraverso registri.

Il WSDL definisce quindi un linguaggio che verra poi scambiato tramite il protocollo SOAP in XML e che permetterà la scoperta da parte dei due agent.

Si consideri il seguente documento a cui si riferiscono le domande 1, 2 e 3:

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:si="https://www.w3schools.com/rdf/">
```

```
<rdf:Description rdf:about="https://www.w3schools.com">
 <si:title>W3Schools</si:title>
 <si:author>Jan Egil Refsnes</si:author>
</rdf:Description>

</rdf:RDF>
```

- Di che strumento si tratta e a cosa serve?

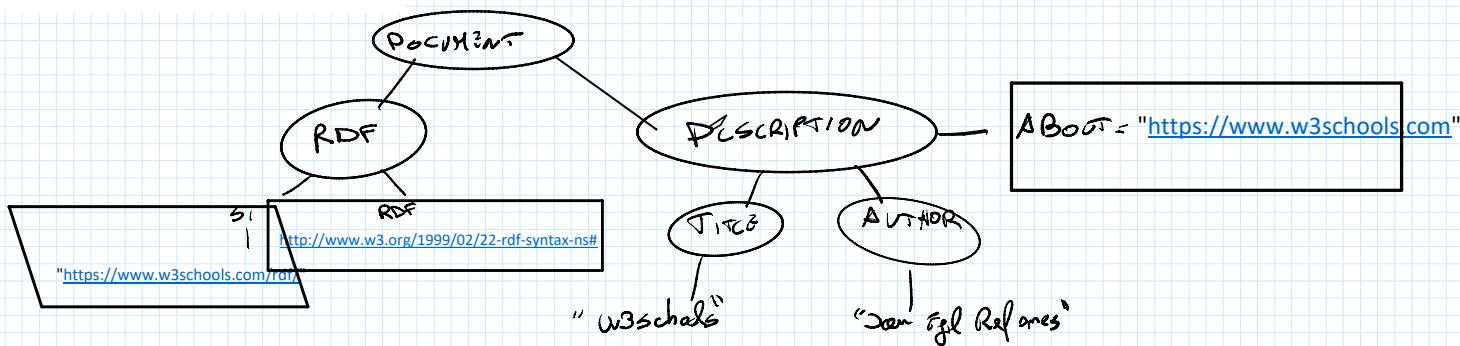
Si tratta di RDF ovvero Resource Description Framework. Serve a definire le risorse che verranno poi salvate generalmente in un database relazionale. RDF permette di definire gli statement, le risorse, per raccogliere tutti i dati dal web ed organizzarli in un modo che sia di facile comprensione per la macchina. Infatti RDF si appoggia generalmente su ontologie e dati e permette una indicizzazione ed una lettura migliore per le macchine. RDF fa uso di XML per il salvataggio di queste risorse ed infatti possiamo notare la sintassi.

- Si individuino soggetti, predicati ed oggetti.

<rdf:Description e' uno statement che si riferisce al soggetto <https://www.w3schools.com>

I due predicati sono <si:title> e <si:author> e con i loro oggetti rispettivamente W3Schools e Jan Egil Refsnes

- Si schematizzi la DOM del documento.



- Si confrontino i metodi di session tracking basati su autenticazione e su variabili di sessione.

I metodi di session tracking permettono al browser di poter raccogliere dati da poter inviare al server, al fine di permettere un funzionamento immediato e ridurre le tempistiche di caricamento di una pagina web. Il session tracking basato su autenticazione impone al client di far apparire un popup o compilare in qualche modo una form per permettere all'utente di inserire le credenziali di autenticazione, in questo modo il server salva le informazioni del client e puo far ottenere facilmente anche quando l'utente dovesse cambiare browser o pc la pagina dedicata. Essendo HTTP un protocollo senza stato risulta difficile purtroppo poter riconoscere un client. Il session tracking basato su autenticazione presenta dei pregi e difetti. Come vantaggi abbiamo l'accesso indipendente dal browser, quindi e' possibile accedere anche da una zona completamente diversa, gli svantaggi sono che i browser alcune volte non implementano l'autenticazione e che l'utente deve inserire ogni volta le credenziali per poter accedere alla pagina. Le variabili di sessione invece permette di costruire una variabile con l'identificativo della sessione e poter essere scambiato tramite due possibili strade: cookies e/o riscrittura della url. E' molto efficace poiche il browser molte volte puo disattivare i cookies e quindi non permettere il session tracking verso il server, in tal modo con le variabili di sessione lavorerebbe tramite GET sulla url.

- Si descriva il funzionamento di una JSP.

Una JSP e' una pagina server che permette di eseguire codice Java in contemporanea al codice del documento usato, ad esempio HTML. Una JSP quindi permette di essere molto semplificata poiche' quando viene eseguita viene tradotta in una Servlet. Infatti la JSP e' una pagina che viene interpretata e tradotta, nell'eventualita ci fossero errori durante la traduzione verranno segnalati in fase di compilazione della pagina. E' molto piu leggibile il codice del documento rispetto ad una servlet, oltretutto una JSP possiede anche oggetti impliciti per definire il contesto principale di una servlet. Presenta alcuni svantaggi, ad esempio quando la pagina e' del tutto dinamica le JSP presentano uno svantaggio a differenza delle Servlet, ma sono molto comode per definire una piccola

*soggetto*

*Predicato*

*Oggetto*

porzione di codice di una pagina dinamica. Il ciclo di vita di una JSP e' il medesimo di quello di una Servlet, quando la JSP viene chiamata, viene chiamato il metodo INIT della servlet, quando poi entra nel vivo ovvero nello scriptlet viene chiamata una DoGet o DoPost o service, infine quando termina l'esecuzione viene chiamata la destroy.

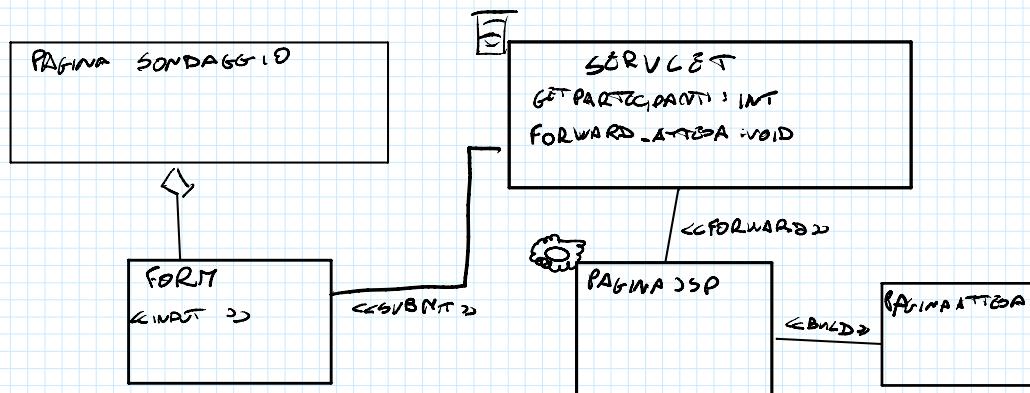
Una pagina JSP ha in più alcuni elementi:

- direttive: <%@ directive %>
- dichiarazioni: <%! declaration %>
- scriptlet: <% scriptlet %>
- espressioni: <%= expression %>
- librerie di etichette

7. Quali sono le caratteristiche di JAVA che lo rendono un linguaggio sicuro e quindi adatto alle applicazioni web?

Java e' un linguaggio sicuro poiche' a differenza di altri linguaggi non ammette l'aritmetica dei puntatori, o meglio la elimina quanto più possibile. Inoltre Java ad ogni esecuzione esegue la parte di codice in una sandbox che si tratta di una macchina virtuale che esegue il programma su una zona riservata di memoria. Questo e' stato fatto per evitare intrusioni nella memoria non consentita che e' la causa delle fallo di sicurezza della maggior parte delle applicazioni. Java inoltre e' un linguaggio che e' fortemente tipato cio permette di riconoscere in fase di compilazione eventuali errori. Java inoltre compila il codice in bytecode e lo esegue in una JVM. Il codice compilato il bytecode permette di essere compilato ed eseguito su varie piattaforme.

8. Si consideri un sistema web per un sondaggio: mi viene presentata una form da compilare e una volta che ho compilato e sottomesso mi viene presentata una pagina col numero di partecipanti al sondaggio fino a quel momento. Si schematizzi la site map del sistema tramite l'estensione UML di Conallen.



1.

```

1. <html>
2. <body>
3. <%!
4. int cube(int n){
5. return n*n*n;
6. }
7. <%= "Cube of 3 is:"+cube(3) %>
8. </body>
9. </html>
```

Di che strumento si tratta? Da cosa si capisce? Se ne descriva nel dettaglio il funzionamento.

Si tratta di una pagina JSP, lo possiamo notare dalla definizione delle expression e delle declaration in formati che solo la JSP puo usare. Inoltre il linguaggio all'interno e' java lo possiamo notare dalla dichiarazione della funzione cube che restituisce un intero. Il resto della pagina che e' statico e' scritto in HTML. La pagina permette di scrivere il valore "Cube of 3 is: 27", infatti a riga 2 viene dichiarata una funzione cube che prende un intero come parametro in ingresso, esegue la moltiplicazione di questo intero con se stesso per 3 volte restituendo quindi alla funzione chiamante il risultato del valore al cubo. A riga 7 viene effettuata una espressione Java che restituisce una stringa che in questo caso viene aggiunta all'html, viene chiamata la funzione cube e viene passato il valore 3, la funzione restituisce 27 e il risultato finale risulta "Cube of 3 is: 27".

2. Scope delle variabili in Php.

Le variabili in PHP possono essere di diversi tipi, questi tipi sono: Integer, Double, String, Boolean, Array, Object e NULL. Il loro scope può essere superglobale (sono visibili ovunque), globale (visibili nello script stesso) ed infine locali (visibili solo all'interno delle funzione in cui sono definite).

4. In che senso AJAX è “asincrono”?

AJAX acronimo di Asynchronous JavaScript and XML permette di poter caricare una pagina in modo asincrono. La difficoltà del browser e' quella che ad ogni richiesta che il client svolge al server, il server risponde rispedendo l'intera pagina al client il quale deve rielaborarla e riaggiornare l'intera pagina. Con AJAX invece e' stato eliminato questo metodo ed infatti grazie anche ad XML permette tramite XMLHttpRequest di poter ricaricare la pagina solo in determinati punti permettendo quindi di non ricaricare l'intera pagina ma di ricaricare una piccola parte. Questa soluzione permette non solo una reattività migliore del browser ma all'occhio dell'utente sembra quasi invisibile. Il tutto e' possibile grazie ad un lavoro svolto da microsoft con gli ActiveX di cui AJAX ne ha ereditato.

In generale Ajax permette di ottimizzare lo scambio di dati tra client e server andando a richiedere solo i dati necessari a costruire la parte della pagina che va modificata anziché ricaricare ogni volta tutta la pagina. Come accennato sopra, il modello Ajax si basa sugli strumenti seguenti:

JavaScript per permettere la comunicazione col browser e per rispondere ai diversi eventi;

DOM per poter accedere e modificare la struttura della pagina HTML o XHTML;

XML per rappresentare i dati che vengono scambiati tra server e client;

L'oggetto XMLHttpRequest per permettere uno scambio asincrono dei dati tra client e server.

5. Voglio implementare uno spider: quali sono i criteri che uso per scegliere il linguaggio più adatto?

Il problema della analisi di un sito web e' che essendo scritto in HTML ne risulta complicato effettuare il parsing. La sintassi non precisa non permette quindi allo spider o al bot di estrarre i dati correttamente, cadendo molte volte in trappole o in loop.

Ovviamente uno spider e' un bot che esegue moltissime operazioni di retrieving sui browser e ne analizza moltissimi, quindi sicuramente uno spider necessita di essere molto veloce, avere poche istruzioni su cui lavorare. Ad esempio io so che adesso i migliori tracker e indicizzatori sono scritti in Python. Anche PHP viene usato per recuperare informazioni. PERL.

- Si convertano le seguenti classi Java nell'equivalente in JavaScript:

```

public Class Employee {
 public String name;
 public String dept;
 public Employee () {
 his.name = "";
 this.dept = "general";
 }
}

public class SalesPerson extends WorkerBee {
 public double quota;
 public SalesPerson() {
 this.dept = "sales";
 this.quota = 100.0;
 }
}

```

Function Employee ()

```
{
 This.name = "";
 This.dept = "general";
}
```

Function SalesPerson ()

```
{
 This.quota = 100.0;
 This.dept = "sales";
}
```

```
SalesPerson.prototype = new Employee;
```

### 2. Come funziona il metodo di session tracking via cookies?

Il meccanismo di session tracking via cookies permette al server di riconoscere il client quando riceve una richiesta. Essendo HTTP un protocollo senza stato presenta alcuni vantaggi e svantaggi, come svantaggio c'e' quello di non poter identificare un client, ma purtroppo puo capitare di dover ripristinare la sessione o dover salvare i dati di uno specifico client. I cookie quindi entrano in gioco come una possibile soluzione, ad un primo accesso il client quando effettua la sua prima chiamata ad un server, il server verifica se il browser del client supporta ed ha abilitato i cookie ed una volta verificato se e' presente già un cookie preleva i dati dal cookie e ripristina la sessione tramite identificativo, altrimenti genera un nuovo cookie dove viene impostato con la coppia nome-valore ed infine imposta un tempo di scadenza del cookie. Il client quindi riceve il cookie che potra scegliere se salvarlo o meno. A questo punto il cookie permette di essere mandato tramite intestazione di una richiesta HTTP e quindi permette di ridurre i tempi di caricamento di una pagina e soprattutto di login o altro.

### 3. Cosa si intende per CMS e quali sono le sue caratteristiche principali?

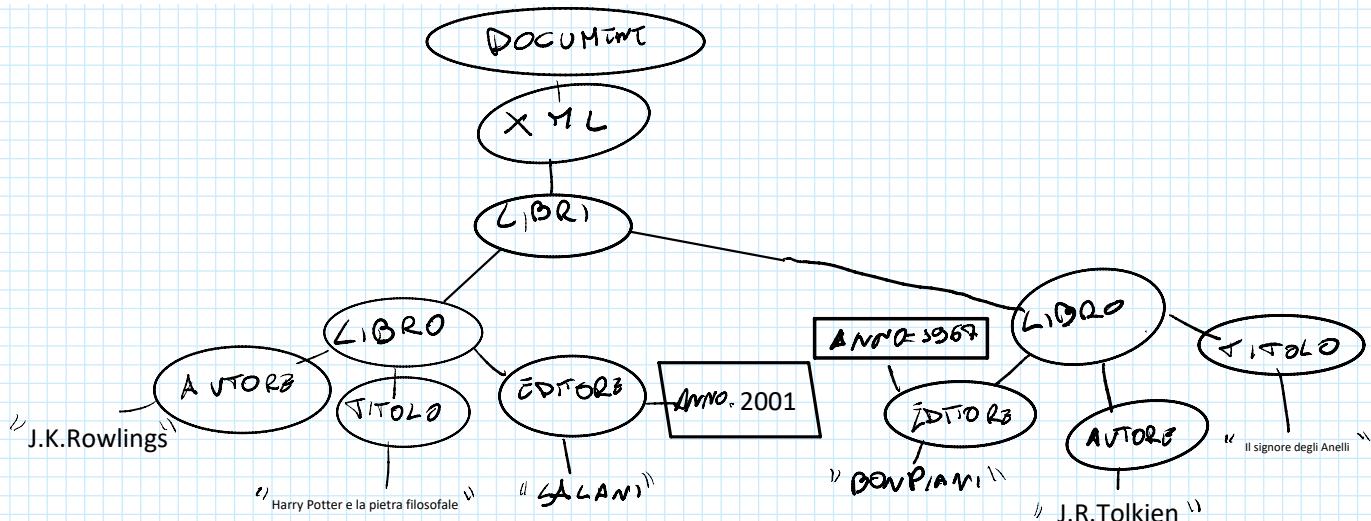
Per CMS intendiamo Content Management System, egli e' un fornitore di servizi che a differenza dei web services e' molto limitato. Infatti e' molto semplice andare a costruire una web application tramite CMS perche' e' stato concepito come una RIA (RICH INTERNET APPLICATION) il che permette di eseguire la maggior parte delle funzionalita lato client. Lo sviluppo avviene su una pagina web ed e' basato su template, le sue caratteristiche principali e' che puo modificare un documento, inserendo anche file multimediali, PDF e OCR. Le sue caratteristiche inoltre sono riguardo al workflow che consente di avere una gestione del lavoro corretta ed infine la gestione degli accessi gestiti tramite utenti con privilegi diversi.

### 4. Cosa si intende per *session storage* in HTML5?

HTML5 ha implementato oltre a varie novita, come tag aggiuntivi, rimozione del attributo name; migliorare la gestione degli errori, rimozioni dei plugin esterni, indipendenza dal device. Per session storage quindi si intende la possibilita da parte del browser di memorizzare dati localmente in alternativa ai cookies, con sempre una coppia di stringhe nome, valore, questi dati vengono inviati al server solo quando vengono espressamente richiesti, a differenza dei cookie ad esempio.

5. Si schematizzi la DOM per il seguente documento XML (non occorre considerare i caratteri di formattazione):

```
<?xml version="1.0"?>
<libri>
 <libro>
 <autore>J.K.Rowlings</autore>
 <titolo>Harry Potter e la pietra filosofale</titolo>
 <editore anno="2001">Salani</editore>
 </libro>
 <libro>
 <autore>J.R.Tolkien</autore>
 <titolo>Il signore degli Anelli</titolo>
 <editore anno="1967">Bonpiani</editore>
 </libro>
</libri>
```



1. Si convertano le seguenti classi Java nell'equivalente in JavaScript:

```
public Class Employee {
 public String name;
 public String dept;
 public Employee () {
 his.name = "";
 this.dept = "general";
 }
}

public class Manager extends Employee {
 public Employee[] reports;
 public Manager() {
 this.reports = new Employee[0];
 }
}
```

```
Function Employee ()
{
 This.name = "";
 This.dept = "general";
}

Function Manager ()
{
 This.reports = new Employee[0];
}
Manager.prototype = new Employee;
```

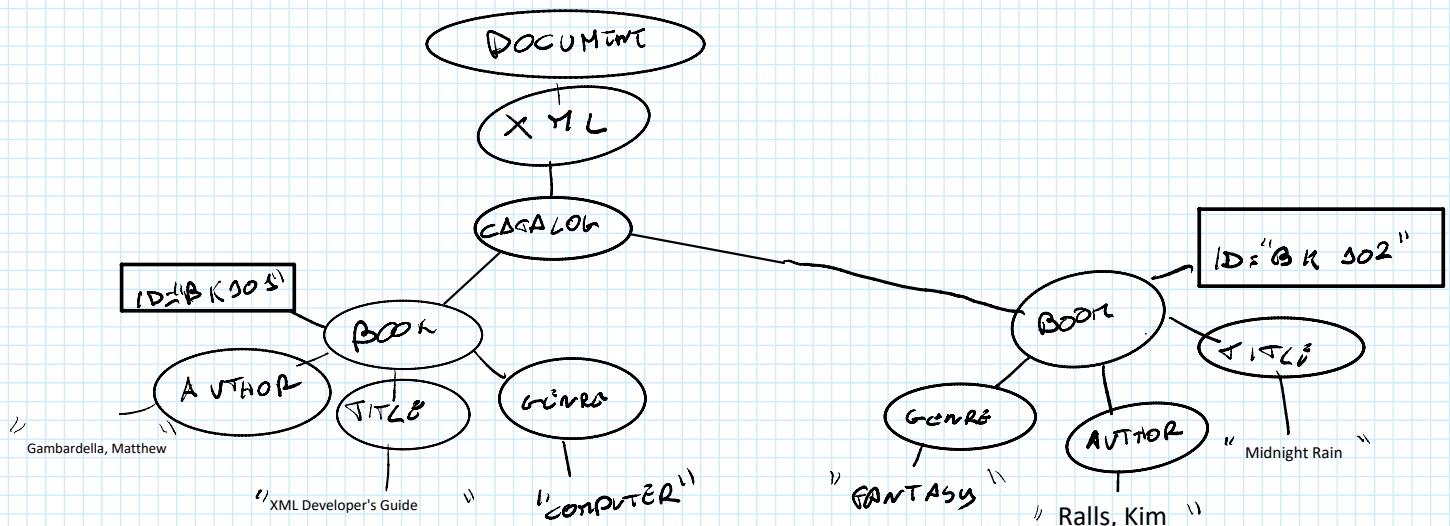
3. Cosa si intende per web framework e quali sono le sue caratteristiche principali?



Per web framework si intendono una serie di strumenti che vengono messi a disposizione per gli sviluppatori affinché possano ridurre i tempi di sviluppo di una web application. Basti pensare ad esempio lato server quante volte bisogna riscrivere il codice per la connessione al db oppure ad esempio metodi per la gestione lato client delle sessioni. I web framework quindi permettono di strutturare una architettura standard di sviluppo di una web application, quella classica e' la three tier architecture con tier che significa insieme logico di funzionalità. La three tier architecture ad esempio e' composta dall'information source, server web e user agent, i quali poi permettono di avere una visione reale dell'insieme. Inoltre poi il server web viene suddiviso in tre logiche distinte ovvero il presentation logic, la business login e il controller logic.

5. Si schematizzi la DOM per il seguente documento XML (non occorre considerare i caratteri di formattazione):

```
<?xml version="1.0"?>
<catalog>
 <book id="bk101">
 <author>Gambardella, Matthew</author>
 <title>XML Developer's Guide</title>
 <genre>Computer</genre>
 </book>
 <book id="bk102">
 <author>Ralls, Kim</author>
 <title>Midnight Rain</title>
 <genre>Fantasy</genre>
 </book>
</catalog>
```



1. Cosa si intende per AJAX e su quali strumenti si basa? Distinguere tra gli strumenti indispensabili e quelli solo utili.

Per AJAX intendiamo Asynchronous Javascript and XML. Si basa sullo sviluppo asincrono che permette di poter ricaricare una pagina dal server non interamente. Quindi in modo del tutto asincrono si basa su strumenti necessari come XML per lo scambio di messaggi e di interpretazione in seguito del codice da aggiornare e quindi su XMLHttpRequest, infine abbiamo lo strumento di Javascript che permette di far ricaricare dinamicamente solo una piccola parte della pagina in modo del tutto invisibile all'occhio dell'utente.

2. Qual è la differenza tra dichiarazioni ed espressioni nelle JSP?

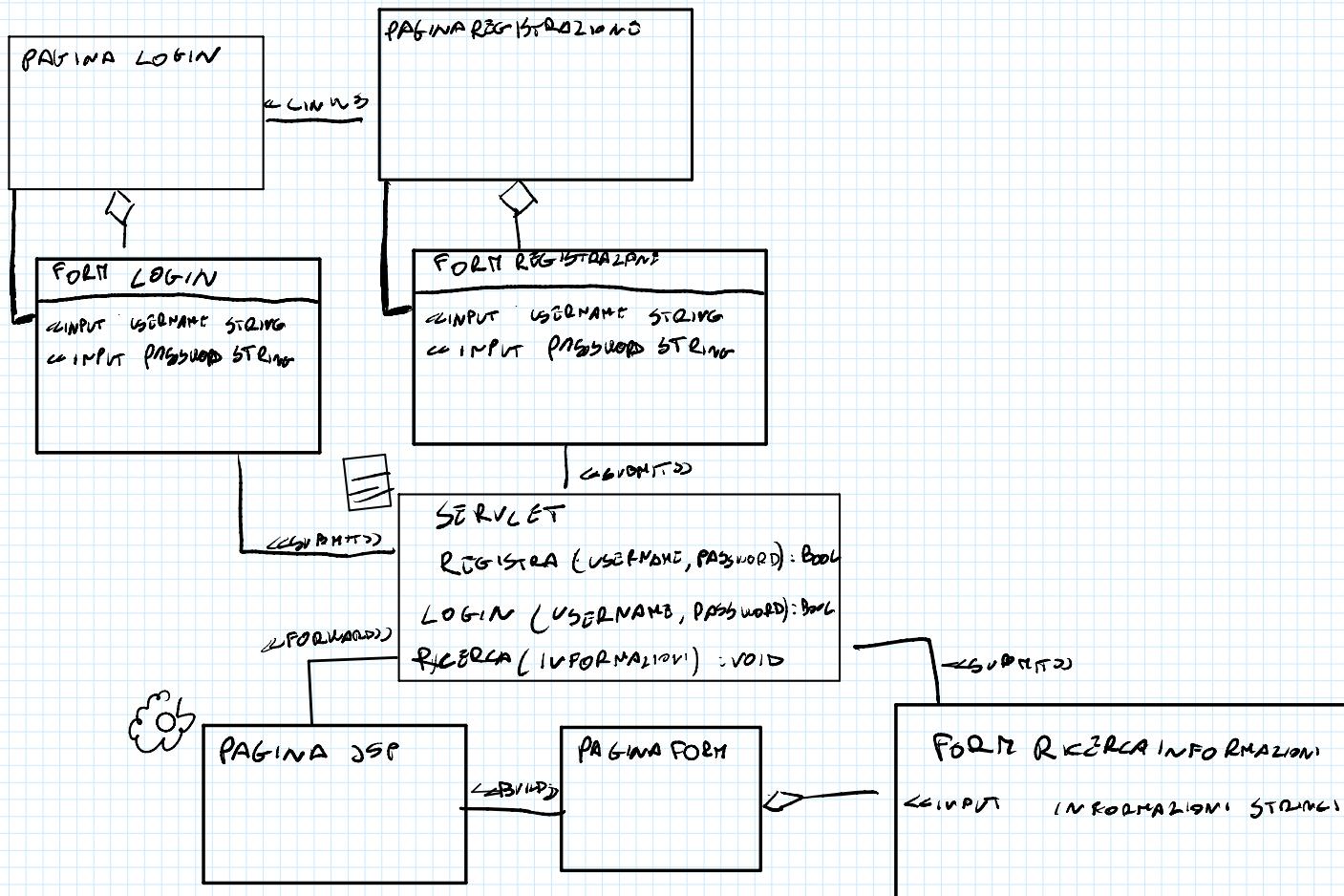
Le dichiarazioni o declaration permettono di definire e dichiarare una variabile e/o funzione nella JSP, a differenza invece delle espressioni o expression che permettono di esprimere una espressione Java che verrà poi convertita in Stringa.

```
<%= Expression %>
<%! Declaration %>
```

3. Quali sono le differenze e le analogie tra XML e HTML?

Entrambi sono linguaggi di markup sono quindi costruiti da etichette (tag) ed elementi ovvero oggetti dei tag. XML serve a definire i dati, quindi permette di definirli, modellarli e costruirli. Permette di avere tag personalizzati con elementi personalizzati a seconda dei dati utilizzati, necessita di uno schema per definirne la struttura. HTML invece visualizza i dati, ha una serie di tag predefinite che permettono di visualizzare i dati. Negli anni XML viene utilizzato maggiormente per lo scambio di messaggi mentre HTML per la visualizzazione di pagine web. C'è anche un'estensione di HTML denominata XHTML che permette di avere entrambi i linguaggi sulla stessa pagina.

5. Si consideri un sistema che proponga all'utente di collegarsi tramite le sue credenziali oppure di registrarsi, gestendo le diverse condizioni di errore. Una volta avvenuto il collegamento, il sistema prevede una form di ricerca di informazioni. Se ne disegni la site map secondo l'estensione di UML di Conallen.



7. Quali funzionalità devono fornire le diverse parti del sistema per supportare un sistema di session tracking tramite campi nascosti?

Per session tracking tramite campi nascosti viene utilizzato solamente quando la pagina prevede form, altrimenti se la pagina e' statica non vi e' la possibilità di poter inviare dati al server. In sostanza nel browser web vengono inserite alcune "form" con attributo hidden=true invisibili nel browser, e permette di inviare dati sulla sessione corrente al server quando l'utente sottomette una form. In tal caso però il server non deve svolgere alcuna particolare funzionalità, deve essere a conoscenza però del fatto che il client vuole adoperare a questo tipo di session tracking. Inoltre sono anonimi e non richiedono all'utente di registrarsi.

Svantaggi

- funziona solo quando lo scambio prevede una sequenza di schede da riempire generate dinamicamente;
- fallisce nel caso di errore che provoca la chiusura del browser.

1. Come funziona l'ereditarietà in Javascript?

Javascript basa la sua ereditarietà su prototipi. Javascript è un linguaggio basato sugli oggetti e su eventi ciò significa che le classi vengono dichiarate tramite l'utilizzo delle function. Per definire una sottoclasse di una classe è possibile farlo dichiarando la prototipazione, infatti ad esempio se abbiamo una classe A padre ed una classe figlia B possiamo dire che B eredita A tramite una catena di prototipi, a sua volta anche A può ereditare da un'altra classe generando quindi una catena.

2. Scope (ambito di definizione) delle variabili in PHP.

Lo scope delle variabili in PHP. PHP prevede le variabili SUPERGLOBALI che vengono posizionate all'inizio e valgono per tutti i documenti in genere sono variabili quasi implicite che contengono informazioni utili per una corretta esecuzione server. Ci sono le variabili GLOBALI che sono condivise con tutto il documento ed infine una funzione può avere anche una variabile LOCALE.

3. Si consideri il seguente documento:

```

1. <html>
2. <body>
3. <p>
4. <script type="text/javascript">
5. $pippo = "pippo";
6. function maiuscolo(){
7. var $pippo = "PIPPO";
8. }
9. maiuscolo();
10. document.write($pippo);
11. </script>
12. </p>
13. </body>
14. </html>
```

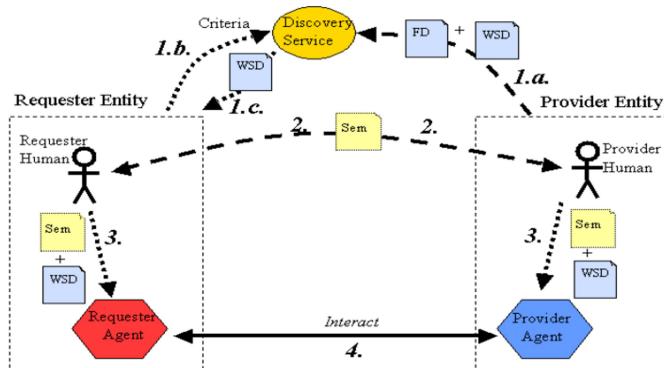
Che linguaggi sono coinvolti (si specificino le righe) e cosa si ottiene mettendolo in esecuzione?

I linguaggi coinvolti sono PHP e Javascript. Javascript lo si può notare dalla riga 4 in poi dove viene aperto uno script type di tipo javascript e le funzioni vengono dichiarate con function inoltre le variabili con var che definisce un linguaggio debolmente tipato. Mettendo in esecuzione il codice si creerà una pagina html all'interno di un paragrafo verrà scritto "pippo" purtroppo la chiamata alla funzione maiuscolo non produce alcun risultato poiché non viene restituita la variabile di ritorno e \$pippo viene sovrascritta come variabile locale nella funzione.

4. Cosa si intende per "oggetti impliciti"? Quale strumento stiamo considerando?

Per oggetti impliciti intendiamo oggetti che sono nel contesto di appartenenza. In questo caso stiamo parlando di JSP, esistono oggetti impliciti che vengono collegati al contesto ed alle applicazioni della Servlet. In questo modo la JSP riesce a sfruttare tutte le funzionalità di una servlet vera e propria, essendo che poi viene tradotta in essa.

5. Si commenti la seguente figura:



La figura rappresenta un meccanismo di scoperta. Per scoperta intendiamo quando un Agent Requester (generalmente) richiede una informazioni il quale trova un Agent Provider che permette di offrire tale informazione. Si parla quindi di servizi web i quali si scoprono a vicenda, la scoperta quindi definisce la

localizzazione di un servizio web. Per servizio di scoperta invece intendiamo un servizio che aiuta nel processo di scoperta. Viene intesa la scoperta quando un servizio web viene trovato e permette di mettere in comunicazioni due Agent. In generale quindi parliamo di un linguaggio WSDL che permette di avere un linguaggio di comunicazione denominato appunto Web Service Definition Language, abbiamo una UDDI che permette il registro di informazioni ed infine tramite la SOAP (Simple Object Access Protocol) possiamo scambiare i dati tra un agent provider ed un agent requester sfruttando la serializzazione di XML.

7. Come si presenta una richiesta HTTP?

Una richiesta HTTP viene presentata ad un server nel seguente modo:

Abbiamo una intestazione formata da una request line, composta da:  
Protocollo utilizzato, hostname, path, versione del protocollo

Nelle righe di intestazione abbiamo Host, connection, user agent

Poi vengono aggiunte informazioni come la IF-MODIFIED-SINCE che permette di capire se la pagina e' stata aggiornata lato server.

Inoltre vengono aggiunte informazioni sulla autenticazione se presente oppure dei cookie.

Infine abbiamo il body che puo essere facoltativo.

2. Posso utilizzare il CSS per raggruppare tutti gli elementi con una data etichetta all'inizio di un documento HTML? Quali altri strumenti posso utilizzare?

Non e' possibile utilizzare CSS per trasformare la DOM di un documento. CSS e' un linguaggio di presentazione che non permette di trasformare il documento, quindi permette solo di effettuare modifiche a livello di presentazione. E' possibile utilizzare ad esempio javascript per la modifica della DOM.

4. Che tipo di commenti posso inserire in una JSP? Quali sono le differenze?

In una JSP e' possibile inserire tramite varie forme i commenti con differenti scopi. Ad esempio e' possibile inserire un commento JSP nel seguente modo: <%- ... ->, mentre i commenti XHTML <!-- ... -->, infine i commenti java. Le differenze sono che con commenti Java o commenti JSP non vengono visualizzati all'utente finale ovvero il client, che quindi non visualizza questi commenti ma rimangono solamente nella jsp, invece i commenti XHTML permangono anche in seguito e quindi anche quando viene inviata al client.

5. Cos'è e a cosa serve il file robots.txt?

Il file robots.txt viene utilizzato dai crawler per permettere una indicizzazione prioritaria su determinati siti. I crawler sono dei bot spider che pero mirano a specifici siti, permette di indicizzare e salvare tutte le informazioni dei siti che visita e ne visita molti. E' piu intelligente di uno spider perche' evita spider trap e i loop.

6)

```
<html>
 <body>
 <h3>Choose an author:</h3>
 <form method="get">
 <input type="checkbox" name="author" value="Tan Ah Teck">Tan
 <input type="checkbox" name="author" value="Mohd Ali">Ali
 <input type="checkbox" name="author" value="Kumar">Kumar
 <input type="submit" value="Query">
 </form>

 <%
 String[] authors = request.getParameterValues("author");
 if (authors != null) {
 %>
 <h3>You have selected author(s):</h3>

 <%
 for (int i = 0; i < authors.length; ++i) {
 %>
 <%= authors[i] %>
 <%
 }
 %>

 <%
 }
 %>
 </body>
</html>
```

1. Di cosa si tratta? Da cosa lo vedo? Cosa succede quando viene messa in esecuzione?

Si tratta di un file JSP, lo possiamo notare dallo scriptlet e dall'uso del linguaggio java. Quando viene messo in esecuzione genera una pagina html che in questo caso da la possibilita di scegliere un autore, scegliendo tramite checkbox uno o piu di essi, sottomette alla jsp la form se il client conferma, in tal caso la JSP prende i valori tramite l'oggetto implicito request che e' della servlet Java, prende il valore degli autori grazie alla funzione getParameterValues con name "author", in tal caso crea quindi un array di stringhe di autori verifica se non e' vuoto e poi genera una pagina html con un tag h3 che riporta al riepilogo della scelta effettuata dall'utente. In tal caso esegue un ciclo for per poter ciclare un codice html che rappresenta una lista non ordinata.

1. Dove è consigliabile mettere la connessione con la base di dati nell'implementazione di una servlet?

Il ciclo di vita di una servlet comprende varie funzioni di configurazione: init (dove generalmente vengono implementate tutte le dichiarazioni ed è consigliabile mettere la connessione con la base di dati poiché viene eseguito una sola volta per tutti i thread di esecuzione della servlet), doGet/doPost/service che eseguono il codice vero e proprio ed infine la destroy che viene chiamata solo quando ogni thread della servlet termina.

2. Si consideri il seguente codice:

```
1. function Mammifero(nome, specie){
2. this.nome = nome;
3. this.specie = specie
4. }

5. var kaki = new Mammifero("Kaki", "zebra");
6. kaki.eta = 7;
```

- a) Di che linguaggio si tratta? Da cosa lo vedo?
- b) Che tipo di funzione è *Mammifero*? Da cosa lo vedo?
- c) L'istruzione alla linea 6 è legale o produce un errore? Perché?

Si tratta di Javascript, lo notiamo dalla dichiarazione della funzione con function a riga 1 ed anche nella dichiarazione di una variabile con var il che fa significare che è un linguaggio debolmente tipato.

Mammifero è una classe, lo si può vedere dalla dichiarazione del suo costruttore.

L'istruzione alla linea 6 è legale perché aggiungiamo all'oggetto kaki della classe Mammifero un attributo che non è presente, in tal caso solamente l'oggetto di nome kaki avrà l'attributo eta a differenza di un normalissimo oggetto Mammifero.

3. Quali sono le differenze e le analogie tra bot, spiders and crawlers?

Tra bot spiders e crawlers sono tutti bot che permettono la ricerca avanzata di dati. I bot verificano solamente alcuni dati sotto specifiche locazioni. Uno spider è un bot ottimizzato che esegue tantissime verifiche e visita un numero molto grande di pagine web, scelte in modo dinamico e/o casuale. Un crawler invece rappresenta l'evoluzione finale, è più intelligente di uno spider ovvero evita tutte le spider traps e loop che un normale spider cadrebbe, indicizza tutte le pagine che visita e segue uno schema ben preciso. È possibile assegnare al crawler un file robots.txt che esegue con particolari funzionalità una ricerca particolare nelle pagine web che lo possiedono.

6. Cosa si intende per JavaBean?

Per JavaBean intendiamo una classe Java con determinate caratteristiche. Nelle JSP poiché si è notata la difficoltà di creare classi ed oggetti in un codice non definito interamente in linguaggio Java, si è deciso quindi di sfruttare i Bean che sono classi Java che hanno il costruttore vuoto, attributi con getter e setter e permette quindi di essere serializzato ovvero deve essere presente una funzione che restituisce l'istanza di quell'oggetto. I Java Bean quindi permettono alle pagine JSP di accedere direttamente a queste classi e di poter usufruire come oggetti impliciti.

1. In che senso diciamo che Java è un linguaggio sicuro?

Java e' un linguaggio sicuro perché evita il più possibile l'aritmetica dei puntatori, evita come altri linguaggi ai programmi di poter accedere alla memoria riservata e non accessibile, dedica uno spazio di memoria ad ogni singolo programma eseguito tramite la sandbox che quindi costruisce una macchina virtuale apposita tramite anche la JVM che alloca uno spazio di memoria apposito. Inoltre e' un linguaggio fortemente tipato il che lo consente ad effettuare tutti i controlli in fase di compilazione, per questo viene usato molto facilmente lato server.

2. Si consideri il seguente codice:

```

1. <?php
2. $age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

3. foreach($age as $x => $x_value) {
4. echo "Key=" . $x . ", Value=" . $x_value;
5. echo "
";
6. }
7. ?>
```

- a) Di che linguaggio si tratta? Da cosa lo vedo?
- b) Che tipo di costrutto è `age`? Da cosa lo vedo?
- c) Qual è l'effetto di questo script?

Si tratta di PHP lo possiamo notare da riga 1 dove c'e' il costrutto iniziale per php, anche dal ciclo for in riga 3 e dalla printout verso la pagina (echo) a riga 4 e 5.

Age e' un array di oggetto scritto in PHP, lo possiamo notare poiche' ci sono piu valori per ogni singolo oggetto infatti ogni oggetto e' composto dal nome e la sua eta, molto probabilmente si tratta di un oggetto Persona. Poiche' PHP e' un linguaggio orientato agli oggetti e' plausibile.

Lo script crea un array di oggetti Persone con attributo nome ed eta, a quel punto esegue un ciclo enumerativo che parte da 0 ed arriva fino al termine dell'array age, per mostrare sulla pagina html dei valori con scritto ad esempio Key=Peter, value = 35 e via di seguito, infine mette un br per tornare a capo.

3. Perché si dice che JavaScript è basato sugli oggetti e non orientato agli oggetti?

Javascript non e' un linguaggio orientato ad oggetti poiche' non vi e' possibile sfruttarne tutte le caratteristiche come encapsulamento, overriding, ereditarietà (come si deve). E' un linguaggio basato su oggetti, permette di costruire prototipi e di sfruttare la prototipazione come ereditarietà, di instanziare oggetti ma non permette di avere tutte le caratteristiche associate ad esso. Poiche' la stessa definizione di classe non e' concepita in javascript.

4. Cos'è e a cosa serve SOAP.

Simple Object Access Protocol e' un protocollo che esegue lo scambio di messaggi XML serializzati. Viene utilizzato per effettuare lo scambio di messaggi tra un agent provider ed un agent requester con un linguaggio predefinito del servizio web, WSDL. SOAP permette di serializzare un file xml e di effettuare lo scambio, e' basato su HTTP quindi su protocollo TCP/IP.

6. Cosa sono e come funzionano le etichette personalizzate in JSP?

In JSP e' possibile definire alcuni tag personalizzati che permettono tramite lo scriptlet definito `<%....%>`.

Le etichette personalizzate sono elementi di linguaggio JSP definiti dall'utente. Quando la pagina JSP contiene un tag personalizzato verrà convertito in servlet, la conversione etichetta per l'oggetto si chiama operazioni di un tag handler che, una volta eseguito chiama un contenitore Web servlet che esegue tali operazioni. In molti casi, le etichette personalizzate rappresentano un'alternativa all'uso dei JavaBeans, ma questi ultimi non possono manipolare il contenuto della pagina, e comunque il loro uso richiede una qualche conoscenza di Java.

tag personalizzati in JavaServer Pages (JSP) sono tag XML creati per eseguire azioni personalizzate in una pagina JSP. tag personalizzati separano il design e il layout di una pagina JSP dal codice Java che guida l'azione. Utilizzando tag personalizzati rende le pagine JSP più leggibile, consente di riutilizzare il codice, e rende l'applicazione più gestibile. Web designer possono utilizzare il codice senza sapere nulla di Java semplicemente utilizzando i tag personalizzati progettati

nelle loro pagine JSP.

### 1. Come sono fatti i messaggi HTTP?

I messaggi HTTP sono divisi in messaggi di request e response. Infatti quando un client effettua una request ad un server e' composto dal:

Request line: Versione HTTP, url, metodo (GET, PUT, DELETE, POST, HEAD)

Intestazione

Riga bianca,

Body opzionale

Nella response dal server al client, invece:

Status Line: Codice risposta, descrizione codice risposta, versione protocollo,

Intestazione

Riga bianca

Body opzionale

Quando ad esempio viene richiesta una pagina con una GET si presentano i dati nella url che viene inviata al server, al contrario quando viene usata ad esempio una POST viene caricato il tutto nel body.

### 2. Con CSS, è possibile fare in modo che i caratteri in bold dentro una tabella siano resi in verde? Se sì, in che modo?

CSS e' un linguaggio di presentazione e permette quindi di poter modificare la formattazione nel rendering di contenuti di un documento, che sia HTML o anche XML. E' possibile quindi far svolgere a CSS la modifica dei caratteri in bold in questo modo:

```
Table b : {
 Color-font = "green"
}
```

### 3. JavaScript ha tipi di dato? Se sì, quali?

Javascript ha vari tipi di dato. Ci sono i tipi di dato primitivi che sono:

Null, Number, Object, String, Undefined, Boolean

Poi ci sono gli oggetti built-in e gli oggetti che vengono costruiti dinamicamente dal programmatore. Gli oggetti built-in sono come le classi Wrapper in java, vengono inclusi vari oggetti con piu funzionalita.

### 5. Cosa si intende per DOM? In che linguaggio è implementato?

Per DOM intendiamo Document Object Model e' una API, ovvero una Application Programming Interface che permette di definire la struttura logica di un documento. Essendo una interfaccia la DOM permette di avere diverse implementazioni a seconda del linguaggio e della piattaforma utilizzata, per questo la DOM viene resa indipendente dal linguaggio e dalla piattaforma. Egli permette di costruire una struttura logica ad albero di un documento e di poter andare ad indicizzare il tutto per poter accedere agli elementi del documento in modo diretto. E' una struttura che non e' ad ereditarieta ma e' di contenimento, ovvero ogni elemento e' figlio di un elemento padre che pero non eredita nulla. Ogni elemento e' figlio di un altro elemento tranne la radice che e' il documento. La DOM permette di lavorare con SCRIPT che ne facilita l'uso, come accesso e aggiornamento dinamico di contenuti, strutture e stile.

La DOM viene implementata in linguaggi di scripting come Javascript, ma non e' limitata al singolo linguaggio.

### 6. Cosa si intende per aritmetica dei puntatori? Java la ammette?

L'aritmetica dei puntatori e' una tecnologia che viene utilizzata particolarmente dai linguaggi C e C++, poiche' permette al programmatore di poter sviluppare applicazioni che permettono di accedere alla memoria in modo dinamico senza limiti e confini. Infatti in questi linguaggi e' possibile tramite una applicazione accedere a memoria che non e' di proprietà della applicazione, e quindi ad andare a corrompere dati che non gli appartengono. Per questo Java non ammette l'aritmetica dei puntatori, poiche' Java e' stato progettato come un linguaggio per essere sicuro, questa tecnologia non permette di avere la stessa sicurezza poiche' la maggior parte delle applicazioni di hacking permettono di accedere a memoria che non gli appartiene. In Java vengono fatti molti controlli sulla sicurezza, vengono effettuati controlli in fase di compilazione sulle variabili e sul tipo di variabili, permette di eseguire applicazioni su memoria riservata in una sandbox veicolata dalla JVM, sfrutta il garbage collection per eliminare variabili di memoria non indicizzate.

## 7. Come funziona il session tracking via campi nascosti?

Il session tracking via campi nascosti e' una delle prime tecniche di session tracking. Infatti il session tracking via campi nascosti permette a chi sviluppa la pagina lato client di imporre alcune form con attributo hidden = true, permettendo quindi di fare inviare al server in modo totalmente anonimo i dati sulla sessione. E' un ottimo meccanismo di session tracking se pero la pagina che si dovrà usare sia sviluppata in modo dinamico e quindi abbia una serie di sequenze di schede da riempire generate dinamicamente, altrimenti risulta impossibile andare ad inviare dati al server, inoltre il server non deve modificare nulla dal proprio lato e può quindi rispondere alla richiesta in modo normale, però si possono creare errori lato client non permettendo di risolvere e costringendo il client a chiudere il browser.

### 1. Cosa si intende per "oggetti impliciti"?

Per oggetti impliciti intendiamo oggetti che vengono caricati in fase di inizializzazione di una pagina JSP. In una pagina JSP viene caricato il ServletContainer per poter inizializzare correttamente la pagina JSP con la Servlet, infatti nella fase di init della Servlet vengono caricati alcune variabili definite come oggetti impliciti che permettono alla pagina JSP di poter eseguire anche la parte di codice relativa alle response, alle request, al context application, definiamo quindi:

Application,  
Page,  
Request,  
Session

Questi sono gli scope degli oggetti impliciti che permettono di poter eseguire alcune funzionalità che altrimenti risulterebbero impossibili da eseguire, lato server.

### 2. Si commenti la seguente istruzione:

```
<!ELEMENT paragrafo (#PCDATA | bold)+>
di che strumento si tratta? Cosa si prefigge?
```

In questa istruzione stiamo definendo una DTD ovvero una Document Type Definition e viene definito un elemento in questo documento di nome paragrafo, possiamo notare che è un tipo "Parsed Character Data", che indica del testo generico. Ed infine sappiamo che deve essere bold. Viene definito un elemento a contenuto misto poiché questo singolo elemento è composto da un altro elemento denominato bold.

### 3. Cosa si intende per RDF e a cosa serve?

Per RDF intendiamo una Resource Definition Framework, si tratta di un Framework per definire le risorse del web semantico. Infatti RDF permette di indicizzare i dati come se fossero su una grandissima base di dati relazione, è quasi possibile infatti fare un paragone tra di essi, RDF quindi permette di associare ad ogni elemento una forma ontologica che però è di facile interpretazione per il computer e non per una persona, la quale permette in un secondo momento di poterla ricercare in modo ontologico, permettendo una lettura molto veloce ed anche un'ottima indicizzazione. RDF quindi permette di lavorare tramite XML e di indicizzare le risorse tramite alcuni meccanismi, vengono definiti quindi gli statement (o anche soggetti) poi i sottoelementi vengono definiti predici e infine il contenuto dei predici viene definito come oggetto.

### 4. Qual è la differenza tra un CMS e un Web framework?

Definiamo cosa è un CMS. Un CMS detto Content Management System permette di definire una web application partendo dalla definizione di template. Infatti permette agli sviluppatori di poter costruire una web application in poco tempo del tutto autonoma e con alcune limitazioni. Un CMS viene categorizzata come RIA ovvero una Rich Internet Application che si basa principalmente sullo sviluppo lato client, infatti un CMS non permette lo sviluppo lato server ma solamente quello lato client. Lo sviluppo è possibile tramite un browser che mette a disposizione una interfaccia per poterla sviluppare. Le funzionalità sono limitate ma permette ad esempio di controllare documenti, di effettuare OCR di PDF, di gestire file multimediali. Le due importanti funzionalità sono: gestione degli accessi al contenuto, infatti permette di avere una gestione degli utenti e di poter avere una gestione degli accessi alle varie viste; ed infine il workflow, infatti permette di avere una gestione ottimale per poter permettere lo sviluppo coordinato per costruire la web application. In poche parole un CMS permette di offrire un sistema che sviluppa automaticamente la parte di software lato server. A differenza invece di un Web Framework, che altro non è che un framework basato su applicazioni web. Un framework rappresenta una serie di strumenti che vengono messi a disposizione di uno sviluppatore per permettere di alleggerire il carico sullo sviluppo sia server che client, come ad esempio la gestione della connessione con la base di dati, oppure lo scambio di messaggi HTTP.

1. Quali sono le differenze tra session tracking tramite variabili di sessione e session tracking tramite autenticazione? Spiegare come funzionano entrambi i sistemi.

Il session tracking tramite variabili di sessione permette di far scambiare l'identificativo della sessione tra client e server attraverso due più noti metodi di session tracking, ovvero riscrittura dell'url e cookies. Infatti tramite questi due meccanismi è possibile far scambiare queste variabili di sessione al fine di poter far ottenere al server tutti i dati richiesti. Il session tracking tramite autenticazione invece veniva utilizzato principalmente con le richieste HTTP, infatti quando il client invia una richiesta al server ed il server risponde, può rispondere con il codice 401 Authorization Required, in questo modo il browser web deve richiedere tramite popup o tramite form di far inserire all'utente le credenziali. L'utente inserisce le credenziali il client invia le credenziali al server tramite il campo nell'intestazione della richiesta HTTP Authentication, ed il server salva i dati della sessione. È molto comodo perché permette al server di identificare un client senza grosse spese, il client inoltre anche se cambia browser o macchina può sempre effettuare l'accesso e quindi ripristinare la sessione che poi viene salvata in cache sul browser. La richiesta però si adatta solo a determinate situazioni come transizioni commerciali ed informazioni sensibili in tutti gli altri casi occorre un session tracking anonimo.

3. Quali sono le applicazioni che di solito fungono da client web e che funzionalità devono provvedere?

Le applicazioni che fungono da client web sono i browser web che permettono all'utente di potersi interfacciare ad una pagina web lato client. I browser web devono poter inviare richieste tramite i vari protocolli ad esempio HTTP ad un endpoint server, nell'eventualità la pagina fosse dinamica, deve poter ricaricare la pagina. Inoltre deve mantenere l'anonimato non può salvare i dati della macchina utente come dati personali o accedere a file non definiti dall'utente, accedere a risorse non dichiarate dall'utente.

Quindi, tra le funzionalità essenziali di un browser sta la visualizzazione di oggetti web, che possono essere documenti HTML, ma non solo.

4. Qual è il ciclo di vita di una servlet?

Una Servlet viene definita come applicazione Java lato server che permette di comunicare con un client e di rispondere a determinate richieste che un server normalmente non potrebbe rispondere. Permette quindi di effettuare da tramite tra le richieste dirette al server e le risposte che il server deve inoltrare al client. Il ciclo di vita di una servlet quindi è formato da:

Init, ovvero la servlet viene inizializzata, qui vengono definite tutte le variabili che permettono ad esempio la connessione con il Database, le variabili che non cambiano valore, ecc.

Service che viene eseguita quando arriva una richiesta da un client e quindi associa un thread ad un client per l'esecuzione, si noti che il metodo service viene eseguito molte volte quante sono le richieste client.

Se ad esempio viene usato il protocollo HTTP il metodo service viene eseguito dalla classe Servlet padre ovvero HttpServlet, che veicola poi le richieste a delle funzioni di nome doXXX le quali poi con XXX rappresenta il tipo di richiesta effettuato, come ad esempio doGet, doPost, ecc.

Destroy viene chiamato alla fine quando ogni thread della servlet viene chiuso.

5. Quali sono gli ambiti di definizione (scope) delle variabili in PHP?

In PHP vengono definiti gli scope delle variabili come SUPERGLOBALE si trattano di variabili definite tra tutti i documenti PHP e che hanno funzioni importanti al fine dello sviluppo lato server, troviamo ad esempio la variabile \$\_COOKIE, oppure \$\_GET, ecc. Poi ci sono le variabili GLOBALI che hanno modo di essere accessibili a qualsiasi funzione all'interno del documento e non, e poi ci sono le variabili LOCALI che permettono alle funzioni locali di potergli accedere.

6. Cosa si intende per "parser a eventi"? In che ambito?

In XML è possibile effettuare un parsing di un documento XML in una struttura logica che permette di poter accedere tramite indici agli elementi di un documento. È possibile quindi svolgerlo in due modi, parser a DOM oppure parser ad eventi. Il parser ad eventi permette di poter trasformare il documento XML in una struttura logica a singoli eventi ovvero, ogni funzionalità presente nel documento XML viene tracciata e viene effettuata la trasformazione solamente di una singola funzionalità per volta, senza evitare di trasformare l'intero documento, a differenza invece del parser a DOM che effettua la trasformazione logica di tutto il documento, egli rappresenta un vantaggio in termini di prestazioni e di scalabilità. Anche se il parser a DOM è consigliato con documenti di piccole dimensioni, al contrario con documenti di grandi dimensioni il parser ad

eventi risulta più efficiente.

## 7. Cosa si intende per "Cloud computing"?

Il cloud computing comprende un insieme di tecnologie informatiche che permettono di fornire come servizi disponibili su richiesta, quindi per cloud computing si intende l'elaborazione come un servizio.

## 8. Cosa si intende e da cosa è supportata la scoperta (discovery) nei servizi web?

La scoperta di un servizio web rappresenta appunto la necessità di un requestor di poter scoprire un servizio web provider. In generale viene effettuata la ricerca di un possibile servizio web disponibile che viene indicizzata in un registro UDDI (che può essere in forma di registro o di indice) UDDI = Universal Description, Discovery and Integration; il quale poi viene visualizzato dal requestor che sceglie o manualmente quindi l'entity ovvero la persona umana sceglie manualmente il servizio oppure in modo automatico quindi tramite servizio di scoperta la macchina sceglie il servizio web che più rispecchia le richieste del requestor. In questo modo quindi tramite la scoperta di un servizio web è possibile far mettere in comunicazione i due agent, requester e provider che comunicano tramite un linguaggio WSDL che viene definito in fase iniziale e che prevede le comunicazioni di scambi di messaggi XML serializzati tramite protocollo SOAP (Simple Object Access Protocol).

2. Nella realizzazione di un documento HTML vogliamo che il primo elemento di ogni lista, sia essa ordinata o meno, sia rappresentato in rosso, e tutti gli altri in blu. Si proponga una soluzione.

E' possibile farlo tramite l'uso di CSS che e' un linguaggio di presentazione. Essendo che in questa richiesta non bisogna necessariamente trasformare il documento, e' possibile farlo tramite CSS, altrimenti si sarebbe richiesto l'uso di Javascript che e' un linguaggio di scripting che permette la trasformazione e modellazione della DOM del documento. In HTML e' possibile inserire un tag style e dichiarare se internamente o esternamente lo stile del CSS tramite "text/css".

Con CSS quindi:

```
Li {
 Font-color = "blu"
}
Li:first-child {
 Font-color = "red"
}
```

Sono quindi previste due richieste style. Siccome CSS esegue le sue istruzioni a livelli, e' previsto quindi una forma di priorità da parte di questi livelli, infatti se nella prima richiesta abbiamo chiesto di rendere ogni elemento di una lista di colore blu, nella seconda abbiamo specificato che ogni primo elemento di una lista deve essere di colore rosso, quindi la specifica e' più importante della richiesta generale e quindi precede in termini di importanza.

3. Quali sono le differenze tra l'attributo NAME e l'attributo ID in HTML?

Le differenze sostanziali sono che fino ad HTML4 l'attributo name veniva usato per identificare un elemento in HTML, invece e' una cosa sbagliata poiche con name e' possibile definire più elementi con lo stesso name, per identificare quindi un singolo elemento in HTML e' possibile farlo con l'attributo ID. In HTML4 e HTML5 l'attributo name e' stato eliminato completamente ed e' rimasto solamente nelle form.

*VEDERE HTML5 Implementazione*

4. Si consideri la seguente classe JAVA:

```
public class SalesPerson extends WorkerBee {
 public double quota;
 public String dept = "sales";
 quota = 100.0;
}
```

come posso implementare un equivalente in JavaScript?

E' possibile farlo tramite l'uso di prototipi. Essendo Javascript un linguaggio basato sugli oggetti, il concetto di classe in javascript non esiste e quindi per poter effettuare una definizione di una superclasse o di un oggetto che estende un altro oggetto e' possibile farlo tramite l'uso di prototipi.

```
Function SalesPerson {
 This.quota = 100;
 This.dept = "sales";
}
SalesPerson.prototype = new WorkerBee;
```

In questo modo ho definito un prototipo SalesPerson che estende un prototipo WorkerBee, a sua volta WorkerBee potrebbe estendere altri prototipi generando una catena di prototipi.

5. Quali sono i principali tipi di nodo previsti dalla DOM del W3C?

Nella DOM del W3C e' possibile identificare i tipi di nodi:

Elemento, corrisponde all'etichetta associata all'elemento;

Attributo, corrisponde a nome e valore associati ai vari elementi HTML;

Testo, si indica con la stringa text, ed indica il contenuto del testo;

Commento, indica i commenti HTML ed ha come valore il testo del commento;

Document, indica la radice del documento;

TipoDocumento, si associa a specifiche DTD;

Frammento, indica uno o più nodi fuori dal documento.

7. Cosa si intende per *crawler* e quali sono le sue principali caratteristiche?

Per crawler intendiamo uno spider evoluto, che permette di associare ad ogni visita un indice di visita. Egli permette di visitare un grande numero di pagine web permettendo di indicizzarle per poterle recuperare in una ontologia in un web semantico. E' possibile quindi indicizzare tutti i link di una pagina. Il crawler deve evitare le spider trap che capiterebbe uno spider, deve gestire eventuali loop e deve essere intelligente al fine di prelevare dati ed indicizzarli. Deve essere rispettoso delle politiche imposte dagli amministratori dei siti.

Il file robots.txt è un file ideato per i crawler, esso descrive i limiti che i crawler devono avere durante la loro operazione di una singola pagina web, infatti il file è posto nella head della pagina visitata.

8. Quali sono i tre livelli dei servizi del cloud computing?

**Quali sono i diversi livelli dei servizi forniti dal cloud?**

-**Infrastructure as a Service (IaaS)** sono i servizi che gestiscono le risorse fisiche e le corrispondenti macchine virtuali, e comprendono i server di elaborazione, la memorizzazione dei dati, i firewall e il gestore dei carichi computazionali;

-**Platform as a Service (PaaS)** rappresentano l'ambiente di sviluppo che permette di implementare le applicazioni e includono linguaggi di programmazione, librerie, frameworks, editor di mashup, dati strutturati;

-**Software as a Service (SaaS)** comprendono le applicazioni vere e proprie; sono gestiti dal browser web, comprendono i social networks, le applicazioni di utilità, quali office, gli strumenti di elaborazione video, etc.; l'utente accede ai servizi di questo livello attraverso un portale web e non deve più preoccuparsi della manutenzione e dell'aggiornamento del software.

Si consideri la seguente pagina web:

```

1. <%@ page import="java.util.*" %>
2. <html>
3. <head>
4. <script type="text/javascript">
5. <!--
6. function alertUser() {
7. alert("Se hai compilato tutti i campi clicca submit");
8. }
9. // -->
10. </script>
11. </head>
12. <body>
13. <H1>DisplayFormData</H1><H3>
14. <%>
15. Enumeration inputParameters = request.getParameterNames();
16. if(!inputParameters.hasMoreElements()) { %>
17. <form method="POST" action="DisplayFormData.jsp">
18. <% for(int i = 0; i < 10; i++) { %>
19. Field<%=i%>: <input type="text" size="20"
20. name="Field<%=i%>"
21. value="Value<%=i%>">
22. <% } %>
23. <INPUT TYPE="button" name="bottone" value="Cliccami!"
24. onClick="alertUser()">
25. <INPUT TYPE=submit name=submit value="Submit"></form>
26. <%} else {
27. while(inputParameters.hasMoreElements()) {
28. String field = (String)inputParameters.nextElement();
29. String value = request.getParameter(field);
30. }
31. <%>
32. <%= field %> = <%= value %>
33. <% }
34. } %>
35. </body>
36. </html>
```

- Quali linguaggi sono coinvolti? Si indichino i numeri delle righe per ognuno di essi.

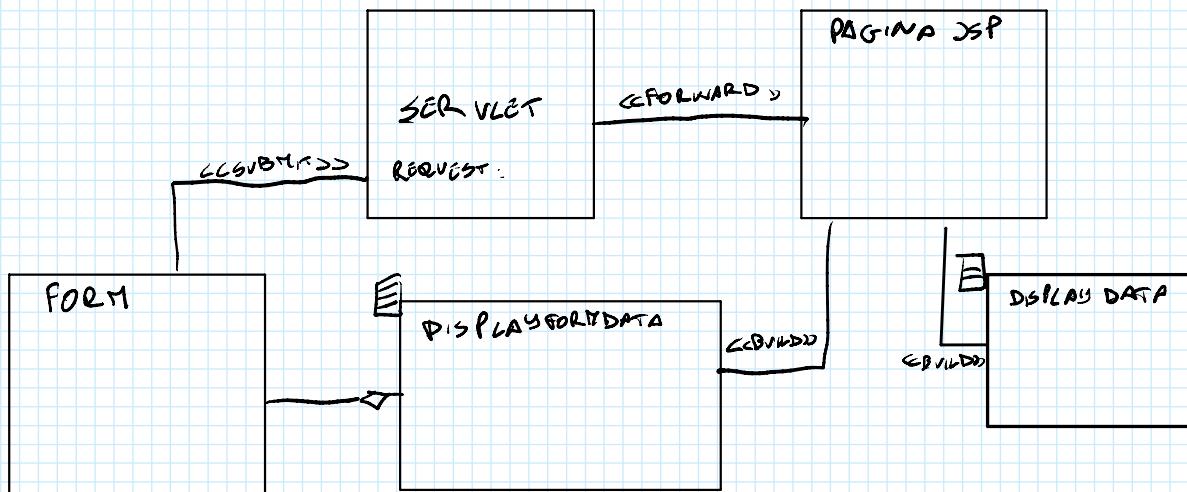
In questo documento sono inclusi i seguenti linguaggi:

Alla riga 1 abbiamo una declaration di una JSP, quindi possiamo subito intendere che questa e' una pagina JSP con codice sia HTML che Java.

Alla riga 4 pero abbiamo uno script lato client che viene scritto in Javascript. Di seguito poi abbiamo tutto linguaggio Java nel tag JSP scriptlet, infine abbiamo una expression a riga 32 che converte i valori presenti nelle variabili java in stringhe.

- Si rappresenti la pagina con l'estensione di UML di Conallen.

Nell'estensione UML di Conallen dobbiamo rappresentare la seguente come 2 pagine, una pagina lato client ed una pagina lato server che rappresenta la JSP. Inoltre poiche la JSP lavora su una servlet in modo automatico dobbiamo rappresentare anche la servlet. La pagina lato client chiama la JSP che restituisce una nuova pagina HTML lato server.



3. Si spieghi il funzionamento della pagina. (vale come due risposte)

La pagina lato client effettua una richiesta al server con una form di tipo post alla pagina DisplayFormData.jsp la quale questa pagina lato server verifica se NON ci sono dati nei parametri passati in input allora mostra di nuovo la medesima pagina ma con le form dei dati, facendo un ciclo for da 10 posizioni che inserisce 10 input type con field iesimo e value iesimo. Altrimenti se riceve la risposta con i dati compilati allora mostra una lista di questi dati con campo field e value.

4. Quali sono le differenze tra session tracking coi campi nascosti e con le variabili di sessione?

Il session tracking con i campi nascosti e' un metodo di session tracking anonimo che permette di ottenere i dati della sessione del client senza che il client stesso lo sappia. E' usato con pagine dinamiche con tante form di invio al server, infatti con pagine statiche non potrebbe funzionare. Permette al server di non modificare la sua struttura e di ricevere le informazioni in modo anonimo. Puo pero provocare crash inattesi del browser.

Le variabili di sessione invece permettono di sfruttare due metodi già ampiamente diffusi come i cookies e la riscrittura dell'url. Infatti il client permette di inviare al server una grossa mole di dati e di permettere di scambiare solamente l'identificativo della sessione. Lo svantaggio è il carico computazionale per il server, soprattutto se deve memorizzare una gran quantità di variabili di sessione.

5. Cosa si intende per SOAP e a cosa serve?

Per SOAP acronimo di Simple Object Access Protocol intendiamo un protocollo che permette lo scambio di messaggi XML tra due agenti. Infatti il protocollo SOAP viene usato per lo scambio di messaggi serializzati in formato XML per la comunicazione tra due agenti, provider e requester. In questo modo dopo aver definito un linguaggio per il servizio web WSDL e dopo aver quindi fatto scegliere il servizio web al requester, il provider può inviare messaggi sfruttando la WSDL e serializzando i file XML tramite SOAP ed inviarli.

6. Quali sono le differenze tra un servizio web e una servlet?

Un servizio web è completamente differente da una servlet. Un servizio web permette di mettere in contatto applicazioni diverse con contesti e linguaggi completamente diversi. Infatti un servizio web è un servizio che deve essere implementato da un agente ovvero un programma che permette di spedire e ricevere messaggi mediante SOAP basato sul protocollo che si implementa, mediante una definizione di un linguaggio WSDL (Web Service Definition Language) che definisce le regole da usare per la comunicazione, ed infine UDDI un registro per permettere il salvataggio di dati dei servizi web.

Una servlet invece rappresenta una applicazione basata sul server, scritta in linguaggio Java e permette di implementare tutte le funzionalità di un vero e proprio server, rappresenta quindi l'interfaccia lato server, e permette di accedere al database di effettuare operazioni, di inviare la risposta al client che richiede, di accettare richieste da client e distribuirle in thread.

```
GET /libraries/expose/interface/css/joomla.css HTTP/1.1
Host: cs-informatica.dieti.unina.it
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://cs-informatica.dieti.unina.it/
Cookie: __utma=64676715.1050198353.1462264360.1469710999.1469795430.22;
__utmz=64676715.1469795430.22.7.utmcsr=google|utmccn=(organic)|utmcmd=organic|
utmctr=(not%20provided); cookieaccept=yes;
3fdcba0619f3002a1a78a7906ab41a31=8e2f3ki9p9o7k0qat9idqpkfl5
Connection: keep-alive
If-Modified-Since: Mon, 28 Apr 2014 10:55:44 GMT
If-None-Match: "5e24f7-1bc4-4f8182834f335"
Cache-Control: max-age=0
```

```
HTTP/1.1 304 Not Modified
Date: Wed, 07 Sep 2016 09:14:39 GMT
Server: Apache/2.2.22 (Ubuntu)
Connection: Keep-Alive
Keep-Alive: timeout=5, max=99
Etag: "5e24f7-1bc4-4f8182834f335"
Vary: Accept-Encoding
```

1. Di cosa si tratta? Da chi viene prodotto, chi ne usufruisce e con quale finalità? Manca qualcosa?

La prima parte si tratta di una richiesta da un client ad un server, lo possiamo notare vedendo la prima riga di request line composta da URL, Protocollo utilizzato e metodo; la riga di intestazione con i vari attributi come anche il cookie che definisce la sessione utilizzata, il campo if-modified-since. Non c'e' un body.

La seconda parte rappresenta risposta da parte del server al client. Possiamo notare come nella status line troviamo il protocollo con la versione utilizzata, il codice di risposta e la descrizione del codice di risposta. Infatti nel codice di risposta troviamo 304 che si riferisce alla richiesta del client di comprendere se il file joomla.css sia stato modificato, infatti nella risposta non c'e' un body perche essendo che il file non e' stato modificato il server evita di reinviarlo al client poiche lo ha gia. Poi c'e' l'intestazione e basta.

Non manca niente perche se il client non avesse chiesto se il file avesse avuto modifiche allora il server lo avrebbe inviato al client mediante il body del messaggio, ma invece siccome il server dice al client che il file joomla.css non e' stato modificato allora non lo reinvia.

2. Il codice comprende dei meccanismi di session tracking? Se sì quali e come funzionano?

Il codice comprende meccanismi di session tracking mediante cookies. I meccanismi di session tracking mediante cookies vengono composti da path, valore e nome e permettono al server di salvare le informazioni relative alla sessione del client. Poiche HTTP e' un protocollo senza stato, non permette al server di poter salvare le informazioni di ogni richiesta da parte di un client, in questo modo tramite i meccanismi di session tracking il server riesce a salvare le informazioni a riguardo. I cookies sono un ottimo metodo di session tracking infatti permettono di essere salvati sul browser web (se vengono abilitati dal browser) e quindi poter riaccedere alla sessione anche in un momento successivo. I cookies hanno informazioni come la data di utilita, ovvero fino a quando possono essere attivi, vengono identificati con nome valore e il path.

3. Nel codice viene menzionata una risorsa joomla.css: di cosa si tratta e a cosa serve?  
Si tratta della risorsa che il client chiede al server di valutare, infatti il client ha gia questa risorsa in formato css scaricata dal server, vuole sapere pero se e' stata modificata dall'ultima volta che l'ha presa ovvero il 2014. Il server risponde al client che non e' stata modificata e quindi la risorsa del client e' aggiornata all'ultima versione.

4. Cosa si intende per "capture phase" e "bubbling phase"? In che contesto?

La fase di capture phase e bubbling phase avviene nel contesto di una DOM quando viene attivato un evento UI o evento logico. In tal caso viene visionato completamente la DOM per permettere di catturare l'evento e quindi avviene una prima fase di cattura per analizzare ogni singolo elemento del documento, una volta analizzati e dopo aver raggiunto la fase di tag phase, ovvero dopo aver acquisito l'elemento che ha generato l'evento avviene la fase di rilascio ovvero la bubbling phase che permette di rilasciare tutte le risorse e riprendere l'esecuzione normale.

5. Si parli di RDF.

RDF acronimo di Resource Definition Framework, permette di definire le risorse per avere un dizionario di definizioni ontologiche al fine di permettere al computer di poter risalire a tali risorse in modo veloce ed intuitivo. Infatti stiamo parlando di web semantico il quale e' composto da ontologie ma non sono rivolte all'uomo bensì al computer, infatti il tutto viene visto come delle risorse che vengono indicizzate mediante predicati, oggetti e soggetti e tramite l'uso del framework RDF possiamo utilizzare XML per la rappresentazione di queste risorse, nonché lo scambio e l'accesso in modo totalmente diretto.

#### 6. Cosa si intende per Web Framework?

Per Web Application Framework o anche solo Web Framework intendiamo un framework basato su ambito Web. Infatti un framework permette di offrire agli sviluppatori una serie di strumenti utili allo sviluppo delle web application che siano sia lato server che lato client. Un web framework quindi permette di poter avere strumenti lato server come ad esempio lo sviluppo della connessione con la base di dati, oppure la gestione delle richieste HTTP ecc ecc. Quindi sono strumenti utili allo sviluppo sia lato client che lato server delle web application.

#### 7. WSDL: di cosa si tratta e in che contesto?

Per WSDL intendiamo il Web Service Definition Language, viene inteso come il linguaggio che permette di definire i pattern di comunicazione tra un agent provider ed un agent requester. Stiamo parlando quindi di servizi web, i quali sono software che permettono lo scambio di informazioni tra due entità, gli agent si scoprono a vicenda e definiscono il linguaggio di comunicazione tramite la WSDL. Una volta definita la WSDL parte lo scambio di messaggi XML tramite SOAP. Ogni servizio web che viene scoperto viene salvato inoltre in uno UDDI che può essere un registro oppure un indice.

#### 8. Qual è la differenza tra dichiarazioni, scriptlet ed espressioni nelle JSP?

Nelle JSP è possibile definire degli scope per poter dichiarare codice dinamico Java e non. Le dichiarazioni infatti permettono di definire le variabili e metodi Java che diventano membri della classe ottenuta dalla traduzione della JSP e che sono resi globali a tutta la pagina JSP. <%! Declaration %>

Gli scriptlet invece eseguono codice Java detto di scripting e viene eseguito in <% scripting %>

Le espressioni invece permettono di definire codice Java che viene poi convertito in output in Stringa quindi ad esempio il valore di una variabile oppure la chiamata ad un metodo che restituisce una stringa, ecc ecc. <%= expression %>