

Rispondere alle domande in modo chiaro e preciso.

1. Si dice che Java è un linguaggio sicuro: quali sono le caratteristiche che lo rendono tale, in particolare in relazione alle applicazioni web?
2. Si consideri il seguente pezzo di codice:

```
1 function sum(a,b){  
2   c = a+b;  
3   return c;  
4 }  
5  
6 var c = 'pippo';  
7 sum(4.5, 7.3);  
8 document.writeln(c);
```

Motivare le risposte:

- di quale linguaggio si tratta?
 - Qual è il tipo della variabile c? Rimane lo stesso lungo tutto il codice?
 - Cosa viene stampato nella riga 8?
-
3. Come si passano i parametri di ingresso in un programma CGI?
 4. Cosa si intende per servizio web? In quale/i linguaggio/i si implementa?
 5. A proposito di JSP, cosa si intende per “oggetti impliciti”?
 6. Cosa sono e come si presentano le JSP?
 7. Esistono i tipi di valore in JavaScript? Se sì, quali sono?
 8. Cosa si intende per "session tracking" e quali sono i modi per affrontarlo?
 9. Cosa si intende per "ben formatezza" e "validità"? In che contesto?
 10. Come è fatta una risposta in HTTP?

11. Cos'è la DOM? Se ne descrivano gli aspetti più importanti.
12. Ereditarietà in JavaScript?
13. Ciclo di vita di una servlet.
14. Cos'è e come funziona il session tracking mediante campi nascosti.
15. Che sintassi possono seguire i selettori in CSS?
16. Come funzionano il session storage e il local storage in HTML5?
17. Quali sono i modi per introdurre istruzioni Java in una JSP?

HTTP

6. Come può il server cancellare un cookie?

Se il server vuole che un cookie venga cancellato, ne invia quindi uno con lo stesso nome, host e path di quello da cancellare, ma con expiration date negativa, in modo che quest'ultimo venga prima sovrascritto e immediatamente cancellato.

2. Come funziona il meccanismo di session tracking con autenticazione basato su HTTP?

Il meccanismo di session tracking può basato su HTTP può avere numerose soluzioni.

Ad esempio:

Autenticazione

Cookie

Variabili di sessione

Riscrittura dell'URL

2. Si consideri la seguente linea:

HTTP/1.x 304 Not Modified

Di cosa fa parte e in quale occasione può essere stata generata.

La seguente riga fa parte della status line e indica al client che la pagina visitata non è stata modificata dalla data inserita nella richiesta HTTP del campo If-Modified-Since. Questa risposta si ottiene inviando un messaggio di richiesta GET condizionato.

1. Quali sono i metodi HTTP, a cosa servono e per cosa si differenziano.

In HTTP/1.0:

- GET: richiede un oggetto;
- POST: invia informazione;
- HEAD: come GET, ma restituisce solo intestazione (usato per debugging);

In HTTP/1.1, abbiamo anche:

- PUT: carica un oggetto;
- DELETE: cancella un oggetto.
- altri (TRACE, CONNECT)

3. Il meccanismo dell'autenticazione può essere implementato in due modi diversi: il primo basandosi su delle caratteristiche dell'HTTP, il secondo no. Si spieghino questi due diversi approcci.

Implementata a livello di HTTP, tramite le variabili di sessione, questo è meccanismo di autenticazione che però risulta molto meno sicuro e quindi sconsigliabile. Finché il browser rimane aperto, username e password vengono salvati sul client, in modo che l'utente non debba digitarli tutte le volte che viene inoltrata una richiesta. Il secondo tramite servlet, si può accedere all'informazione della sessione quando si usa un meccanismo di autenticazione via HTTP. Il metodo `getRemoteUser()` fornisce alla servlet l'username dell'utente dopo che questo ha fatto il login fornendogli insieme ad una password. Questo metodo presenta vari vantaggi tra cui: la facilità di implementazione, funziona anche se l'utente accede al sito da macchine diverse e continuerà a funzionare anche se l'utente esce dal sito o dal browser; Presenta però uno svantaggio, ovvero che la richiesta di registrazione si adatta solo a determinate situazioni (transizioni commerciali, informazioni sensibili);

1. Si spieghi come HTTP offre supporto al meccanismo di caching.

Perché un meccanismo di cache possa venir adottato opportunamente, 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. 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.

2. Quali sono le informazioni contenute in un cookie?

Un cookie è un pezzo di testo che viene scambiato tra client e server secondo opportune modalità. Esso contiene alcune informazioni essenziali:

1. host e path dell'applicazione;
2. expiration date;
3. nome e valore del cookie.

1. Come sono fatti i messaggi HTTP?

I messaggi http si dividono in richieste e risposte. La prima riga dei messaggi HTTP nelle richieste è la request line, mentre nelle risposte è status line

- La request line è formata da 3 campi: metodo, url e versione http
- La status line invece da 3: versione http del server, codice risposta e messaggio

Dopo la prima riga si trovano le righe d'intestazione. Una riga vuota. E il contenuto, opzionale, della richiesta/risposta, detto anche body.

1. Si spieghi come l'HTTP offre supporto al meccanismo di autenticazione.

Tramite le variabili di sessione è anche possibile implementare un meccanismo di

autenticazione che però risulta molto meno sicuro e quindi sconsigliabile. finché il browser rimane aperto, username e password vengono salvati sul client, in modo che l'utente non

debba digitarli tutte le volte che viene inoltrata una richiesta.

HTTP prevede il campo Authorization: Basic per accedere a un'area protetta.

L'area protetta risponderà all'user agent con un codice 401 che richiederà l'inserimento di username e password da parte dell'utente.

Una volta effettuata l'autenticazione le credenziali verranno salvate e inserite nelle richieste successive automaticamente.

HTML

7. Quali sono le principali innovazioni introdotte da HTML5?

HTML5 oltre ad introdurre novità importanti, rimuove etichette tese a descrivere la formattazione del documento, ed inserisce delle nuove etichette che, si riferiscono alla struttura del documento, tra cui section, article, footer, progress, nav. Inoltre un documento HTML potrà seguire una tra due possibili sintassi: text/html, "sintassi HTML", oppure application/xhtml+xml o application/xml La "sintassi XML". Un'altra importante caratteristica di HTML5 riguarda la possibilità di memorizzare dati localmente, all'interno del browser, in alternativa ai cookies, sempre come coppia di stringhe nome=valore, precisamente con gli strumenti di session storage e local storage. Entrambi vengono utilizzati con gli oggetti sessionStorage e localStorage, la differenza tra i due è che la session storage perde i dati nel momento in cui viene chiusa la comunicazione con il server, mentre la local storage conserva i propri dati sul browser.

2. Quale è la differenza tra gli attributi NAME e ID in HTML?

In HTML potrebbero esserci più parametri con lo stesso name.

Al contrario il valore di id deve essere univoco: ne consegue che al di fuori delle form, conviene sempre usare id al posto di name, che infatti è deprecato. Come vedremo, in XML esiste un attributo particolare id, ma non name. In HTML5 name si trova solo associato a form.

2. Si spieghi cosa sono e come funzionano il session storage e il local storage in HTML5.

Un'importante caratteristica di HTML5 riguarda la possibilità di memorizzare dati localmente, all'interno del browser, in alternativa ai cookies, sempre come coppia di stringhe nome=valore. Questi dati non vengono inseriti in ogni richiesta HTTP inoltrata al server, ma solo quando risulta necessario farlo.

1. Quali sono le differenze tra HTML e XML?

Sia XML che HTML sono linguaggi di markup. Tuttavia, XML non soppianta HTML perché i due linguaggi sono stati progettati con scopi diversi: XML deve descrivere i dati: si concentra dunque su cosa sono i dati; HTML serve a visualizzare i dati: si concentra dunque su come appaiono i Dati

CGI

7. Cosa si intende per CGI e su quale linguaggio di programmazione si fonda?

Un programma CGI può venir scritto in qualsiasi linguaggio di programmazione, tra cui:

- C/C++
- Fortran
- PERL
- TCL
- una qualsiasi shell Unix
- Visual Basic

La scelta dipende da quali linguaggi sono disponibili sul server: non ci sono vincoli imposti dal client. Ovviamente, i programmi scritti in linguaggi compilati, quali ad esempio il C e il Fortran, dovranno venire compilati prima di poter essere eseguiti.

4. Posso accedere alle informazioni contenute nell'intestazione della richiesta HTTP da un programma CGI? Se sì, in che modo?

Si può accedere alle informazioni contenute nell'intestazione della richiesta HTTP da un programma CGI utilizzando opportune variabili d'ambiente, che saranno tante quante le richieste header HTTP. Solitamente le variabili avranno il prefisso HTTP_

4. Come vengono passati i parametri in input a un programma CGI?

possono esserci più metodi, come passare i parametri tramite una variabile d'ambiente oppure tramite lo stdin. Per passare i parametri tramite una variabile d'ambiente si utilizza una richiesta HTTP di tipo GET, in cui viene utilizzata la variabile d'ambiente QUERY_STRING, in cui viene inserito tutto ciò che segue il ? nell'URL, ovviamente con lunghezza limitata.

Per passare i parametri tramite lo stdin si utilizza una richiesta HTTP di tipo POST, in

cui tutto ciò che si trova nel body della richiesta HTTP viene passato al processo CGI sullo stdin. Non viene terminato con un EOF, e quindi occorre conoscerne la lunghezza: a tale scopo viene usata la variabile d'ambiente `CONTENT_LENGTH`.

La risposta HTTP da inviare allo user agent utilizza lo stdout del processo CGI, a cui il server premette la riga di stato, quindi avremo una risposta HTTP di questo tipo:

- status line, ad esempio, `HTTP/1.1 200 OK`
- output del programma (che deve quindi contenere anche il resto dell'intestazione header – HTTP)

5. Come si accede ai parametri di ingresso di una richiesta HTTP da un programma CGI?

6. Si mettano a confronto il CGI e le servlet.

Le servlet rispetto al CGI presentano vantaggi in termini di efficienza, di facilità di sviluppo, di accesso all'informazione di contesto e sicurezza. Tuttavia le servlet per essere implementate hanno bisogno del servlet engine o servlet container. Le CGI invece possono essere implementate senza particolari caratteristiche se non il modo di gestire l'input/output. Mentre nelle servlet le informazioni sono condivise sia all'interno dell'applicazione che col server, nelle CGI l'unico ambito di definizione previsto è quello della richiesta, quindi ad ogni richiesta viene avviato un processo, quindi per condividere informazione tra due richieste diverse, occorre che tale informazione venga salvata in modo persistente. Non dà modo ai programmi CGI di interagire direttamente col server o comunque di usare le funzionalità del server dopo che il programma è entrato in esecuzione. Nelle servlet ad ogni richiesta corrisponde un nuovo thread all'interno del servlet engine: thread diversi possono accedere a variabili comuni, e questo permette di condividere informazioni tra richieste diverse alla stessa servlet. Quello che ancora manca alle servlet è la separazione tra aspetti grafici e aspetti di programmazione.

5. Quali sono i vantaggi e gli svantaggi dell'uso delle servlet rispetto al CGI?

5. Quali sono i vantaggi e gli svantaggi dell'uso delle servlet rispetto al CGI?

JavaScript + Java

4. Quali sono le caratteristiche che rendono Java un linguaggio sicuro?

Java è particolarmente sicuro perché, è fortemente tipato, non sono ammessi costrutti insicuri, come l'accesso ad un vettore senza controllo sugli indici, è assente l'aritmetica dei puntatori, causa di molti problemi di sicurezza, è dotata di una Java

Virtual Machine, che verifica il bytecode, precisamente:

Prima dell'esecuzione

Durante l'esecuzione

Ed inoltre la stessa, è dotata di un Security Manager per l'accesso di risorse di sistema. La JVM utilizza una sandbox in cui i programmi possono essere eseguiti senza rischiare di danneggiare il sistema. Inoltre I programmi Java non possono richiamare funzioni globali, né ottenere l'accesso a risorse del sistema in modo arbitrario.

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

È la possibilità di poter accedere a qualsiasi parte della memoria attraverso i puntatori e operazioni aritmetiche. In Java non è possibile accedere alla memoria in modo non controllato.

5. Qual è il ciclo di vita di una applet?

4. In JavaScript posso assegnare ad un elemento di un vettore la stringa "diciassette" e ad un altro il numero 17? e in Java? In caso di risposta/e affermativa/e si spieghi come si fa.

In javascript è possibile farlo in questo modo

```
var vettore = new Array();  
Vettore[] = 17;  
Vettore[] = "diciassette";
```

In java non è possibile farlo perché sono due elementi di tipo diverso

1. Come funziona l'ereditarietà in JavaScript?

L'ereditarietà in JavaScript è basata sui prototipi. Attributi e metodi vengono ereditati solo dal costruttore, ad ogni costruttore è associata una proprietà prototype usata per implementare l'ereditarietà di struttura. Quando un oggetto viene creato questo ha un riferimento implicito al prototipo del costruttore, quindi ogni proprietà aggiunta al prototipo viene condivisa attraverso l'ereditarietà da tutti gli oggetti che condividono quel prototipo. Un prototipo può avere un riferimento implicito ad un altro prototipo creando in questo modo una catena di prototipi che viene risalita per fare riferimenti alle proprietà. Si sale la catena fin quando non si trova un riferimento a null.

4. Esistono e quali sono i tipi di valore in JavaScript?

Number
String
Boolean
Undefined
Null

1. Si dice che JavaScript è basato sugli oggetti e non orientato agli oggetti: si spieghi quali sono le caratteristiche orientate agli oggetti in JavaScript e soprattutto quali le loro limitazioni.

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. Uno degli svantaggi della programmazione basata agli oggetti è quella di non poter implementare l'incapsulamento, e quindi l'information hiding.

7. Ci sono risultati che posso ottenere con la DOM e JavaScript e non con XSL? E risultati che posso ottenere con la DOM e JavaScript e non con CSS? Si spieghi perché.

SERVLET

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

Possiamo distinguere tre fasi fondamentali nel ciclo di vita di un Servlet: inizializzazione, gestione delle richieste e terminazione. Nello specifico abbiamo tre metodi:

- Init(): Viene caricata la classe servlet e viene istanziata, successivamente viene richiamato il metodo init(). Questo passaggio viene eseguito solo una volta, all'arrivo della prima richiesta.
- Service(): Dopo che una servlet è stata inizializzata correttamente, il container può utilizzarla per gestire le richieste dei client. Le richieste sono rappresentate mediante oggetti di tipo ServletRequest mentre la risposta è "confezionata" dalla servlet chiamando i metodi esposti dall'oggetto di tipo ServletResponse. Entrambi gli oggetti sono passati come argomento al metodo service() dell'interfaccia Servlet. Un container può inviare richieste simultanee attraverso il metodo service. Spetta al programmatore gestire opportunamente l'elaborazione all'interno di tale metodo nel caso in cui arrivino più richieste contemporaneamente.
- Destroy(): la servlet viene rimossa dal container non prima però che quest'ultimo abbia invocato il metodo destroy dell'interfaccia Servlet. Questo metodo assicura che la servlet abbia terminato in maniera corretta la propria elaborazione dopodiché rilascia tutte le risorse utilizzate o create durante il suo ciclo di vita.

6. Dove si scrivono i messaggi di errore in una servlet?

Se c'è un problema durante l'esecuzione della servlet, quindi nel metodo service(), viene lanciata una ServletException o una IOException, che conterrà i messaggi di errore che descrivono il problema.

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

Un servizio Web è un modo in cui i sistemi software comunicano tra loro tramite HTTP ecc...Un servlet è un modo specifico Java di scrivere software che risponde alle richieste HTTP. Un servlet è un gestore di query HTTP. Puoi fare quello che vuoi con le tue domande in arrivo. Un servlet eseguito sulla JVM. Il servizio web non ha bisogno di una JVM.

6. Si mettano a confronto il CGI e le servlet.

Le servlet rispetto al CGI presentano vantaggi in termini di efficienza, di facilità di sviluppo, di accesso all'informazione di contesto e sicurezza. Tuttavia le servlet per essere implementate hanno bisogno del servlet engine o servlet container. Le CGI invece possono essere implementate senza particolari caratteristiche se non il modo di gestire l'input/output. Mentre nelle servlet le informazioni sono condivise sia all'interno dell'applicazione che col server, nelle CGI l'unico ambito di definizione previsto è quello della richiesta, quindi ad ogni richiesta viene avviato un processo, quindi Per condividere informazione tra due richieste diverse, occorre che tale informazione venga salvata in modo persistente. non dà modo ai programmi CGI di interagire direttamente col server o comunque di usare le funzionalità del server dopo che il programma è entrato in esecuzione. Nelle servlet ad ogni richiesta corrisponde un nuovo thread all'interno del servlet engine: thread diversi possono accedere a variabili comuni, e questo permette di condividere informazioni tra richieste diverse alla stessa servlet. Quello che ancora manca alle servlet è la separazione tra aspetti grafici e aspetti di programmazione.

CMS E WEB FRAMEWORK

10. Quali sono le differenze tra un CMS e un framework web?

Mentre il CMS può essere utilizzato da chiunque, poiché non richiede un livello di competenza elevato, i framework sono pensati per i programmatori, e non si possono utilizzare senza avere conoscenza di come si scrive qualcosa da zero.

6. Cosa si intende per Web Framework?

Per Web Frameworks si intende un framework per lo sviluppo di applicazioni web, con lo scopo di alleggerire lo sviluppo di applicazioni web, sia per la programmazione lato server sia per la programmazione lato client. 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.

10. A cosa serve un WebFramework

7. Cosa si intende per Content Management System o CMS?

Un CMS è un sistema per la gestione dei contenuti in ambienti di lavoro collaborativo. Un web CMS è un CMS che funziona su web. La maggior parte dei CMS utilizza una base di dati relazionale per immagazzinare i contenuti. un CMS offre supporto alla creazione, alla modifica e alla pubblicazione dei documenti.

8. Cosa si intende per architettura a tre strati (three tier)?

L'architettura descrive la struttura di un'applicazione, effettuando una separazione tra componenti e descrivendo la loro interazione. Per architettura a tre strati si intende: un raggruppamento formato da tre componenti, un livello top dove viene immagazzinata l'informazione, un livello bottom che rappresenta l'utente finale, ed un tier intermedio che controlla l'interazione tra client e sorgente di informazione, ed implementa:

- 1) Meccanismo di controllo, che elabora la richiesta del client e si procura i dati da restituire.
- 2) Meccanismo di presentazione, che elabora i dati per presentarli all'utente (HTMLXML ecc);
- 3) Regole di dominio applicativo, si assicura che i dati siano affidabili prima di usarli per aggiornare la sorgente di informazione o rispondere all'utente

Le applicazioni Web sono un esempio di architettura a tre strati.

JSP

1. Cosa sono gli oggetti impliciti nelle JSP e quali sono i loro possibili ambiti di definizione (scope)?

Gli oggetti impliciti in JSP del container JSP sono previsti per ogni pagina di oggetti Java, gli sviluppatori possono usarli direttamente, senza una dichiarazione esplicita. Gli oggetti impliciti JSP sono anche noti come variabili predefinite.

I possibili ambiti di definizione possono essere:

- scope di applicazione
 - application: il contenitore all'interno del quale viene eseguita la servlet;
- scope di pagina
 - config
 - exception: solo in una pagina di errore
 - out
 - page
 - pageContext

- response
 - scope di richiesta
- request
 - scope di sessione
- session

5. Cosa si intende per dichiarazioni in JSP?

Per dichiarazioni in JSP si intende tutto ciò che è compreso tra le seguenti tag: `<%! dichiarazione %>`

7. Quali sono le differenze tra scriptlet e dichiarazioni in una JSP?

Uno scriptlet può contenere qualsiasi numero di statement, variabili o dichiarazione di metodi del linguaggio JAVA, o espressioni che sono valide nella pagina del linguaggio scripting. Invece per gli elementi “dichiarazione” si possono dichiarare una o più variabili, o metodi che possono essere usati dopo nel file JSP. Bisogna quindi dichiarare la variabile o il metodo prima di usarla/o nel file JSP.

7. Cosa sono gli oggetti impliciti nelle JSP?

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

Dichiarazioni: `<%! dichiarazione %>`

Scriptlet: `<% scriptlet %>`

Espressioni: `<%= espressione %>`

Session Tracking

6. Come funziona il session tracking tramite campi nascosti?

Il session tracking tramite campi nascosti utilizza delle form con input type di tipo hidden. Grazie a questi campi nascosti, il client e il server riescono a scambiarsi informazioni che identificano la sessione corrente, tuttavia questo meccanismo di session tracking può essere applicato solamente su pagine che richiedono la compilazione di form.

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

Per quanto riguarda il session tracking con variabili di sessione, si utilizzano variabili sul server, in cui l'identificativo di sessione viene scambiato tra client e server tramite due possibili strumenti:

1) Cookies

2) Riscrittura dell'URL

La prima volta che il client fa la chiamata al server viene generato un ID, che è unico e quindi identifica il mittente, e lo salva in un file all'interno del server stesso. A questo punto viene generato un cookie che sarà inviato al client e che conterrà quell'ID identificativo, così che la prossima volta che si farà una chiamata allo stesso server, questo leggerà il contenuto del cookie e saprà a quale client appartiene.

Per quanto riguarda invece il session tracking tramite autenticazione, abbiamo che l'utente effettua l'accesso in un sito, e username e password vengono poi conservate sul browser del client, così che ad ogni richiesta HTTP è possibile identificare l'utente tramite l'username con cui ha effettuato l'accesso. Un'applicazione del session tracking tramite autenticazione, può essere ad esempio, l'uso delle servlet con il metodo `getRemoteUser()`;

3. Cosa sono, a cosa servono e come funzionano le variabili di sessione.

Per quanto riguarda il session tracking con variabili di sessione, si utilizzano variabili sul server, in cui l'identificativo di sessione viene scambiato tra client e server tramite due possibili strumenti:

- 1) Cookies
- 2) Riscrittura dell'URL

La prima volta che il client fa la chiamata al server viene generato un ID, che è unico e quindi identifica il mittente, e lo salva in un file all'interno del server stesso. A questo punto viene generato un cookie che sarà inviato al client e che conterrà quell'ID identificativo, così che la prossima volta che si farà una chiamata allo stesso server, questo leggerà il contenuto del cookie e saprà a quale client appartiene.

2. Come funziona il meccanismo di session tracking con autenticazione basato su HTTP?

Il meccanismo di session tracking può basato su HTTP può avere numerose soluzioni. Ad esempio:

Autenticazione
Cookie
Variabili di sessione
Riscrittura dell'URL

3. Per il meccanismo di session tracking mediante campi nascosti occorre supporto dall'HTTP? Si spieghi nel dettaglio come funziona.

Il meccanismo di session tracking tramite campi nascosti, richiede ovviamente un supporto dall'HTTP, per il semplice fatto che nel momento in cui si applica, vi è uno scambio di messaggi tra il client ed il server che permette al server di ottenere un identificativo che identifica la sessione in considerazione.

4. Quali sono le differenze tra session tracking coi campi nascosti e con

le variabili di sessione?

Le differenze fondamentali sono che, mentre il session tracking con campi nascosti necessita di avere una pagina in cui devono essere riempite delle form, il session tracking con variabili di sessione può essere applicato anche su una pagina web statica. Una sessione può scadere, e in aggiunta si può scegliere dove e come memorizzare la sessione: in un processo, in un db, ecc... Un campo nascosto, invece, è un campo quindi i dati sono ottenuti dalla pagina e quindi non scadono.

3. Si spieghi nel dettaglio come funziona il meccanismo di session tracking mediante autenticazione.

Una volta effettuata l'autenticazione il server risponde con una risposta http e il campo Authorization: Basic contenente un codice identificativo che viene conservato dall'user agent e inserito nelle successive richieste.

Il meccanismo di autenticazione può essere implementato in due modi diversi: il primo sulle caratteristiche dell'HTTP, il secondo no. Si spieghino questi due diversi approcci.

Il meccanismo di autenticazione Implementato a livello di HTTP, si appoggia su alcune delle caratteristiche proprie dell'HTTP, ovvero, sui specifici campi dell'intestazione. questo meccanismo di autenticazione che però risulta molto meno sicuro e quindi sconsigliabile. Finché il browser rimane aperto, username e password vengono salvati sul client, in modo che l'utente non debba digitarli tutte le volte che viene inoltrata una richiesta. Il secondo tramite servlet, si può accedere all'informazione della sessione quando si usa un meccanismo di autenticazione via HTTP. Il metodo `getRemoteUser()` fornisce alla servlet l'username dell'utente dopo che questo ha fatto il login fornendogli insieme ad una password. Questo metodo presenta vari vantaggi tra cui: la facilità di implementazione, funziona anche se l'utente accede al sito da macchine diverse e continuerà a funzionare anche se l'utente esce dal sito o dal browser; Presenta però uno svantaggio, ovvero che la richiesta di registrazione si adatta solo a determinate situazioni (transizioni commerciali, informazioni sensibili); in tutti gli altri casi occorre un session tracking anonimo.

CSS

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?

```
Table b { color:green; }
```

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.

```
ul:first-child, ol:first-child{ color:red; }  
ul, ol { color:blue; }
```

3. Si scriva una riga CSS che stampi in grassetto tutti gli elementi di una lista ordinata che è parte di un'altra lista a livello superiore.

```
ul > ul { font-weight:bold; }
```

2. In CSS posso imporre che il primo item di una lista venga scritto in rosso e tutti gli altri in verde? In che modo?

```
ul:first-child { color:red; }  
ul {color: green;}
```

2. In CSS posso imporre che tutti gli item di una lista numerata vengano scritti in rosso e tutti quelli di una lista non numerata in verde? In che modo?

Utilizzando due selettori diversi per le liste numerate OL (order list) per le liste non numerate UL (unorder list)

```
ol{ color:red; }  
ul{ color:green; }
```

3. Cos'è e che sintassi ha un selettore in CSS?

I selettori stabiliscono a quali elementi si applica una regola. Le proprietà (e il loro valore) stabiliscono come l'elemento deve essere rappresentato. Tuttavia, le regole possono anche estendere il contenuto dell'elemento. La sintassi dei selettori _ è molto importante perché determina quali insiemi di selettori possiamo descrivere per associarli nella stessa regola

H1 { color:green; } in questo esempio H1 è il selettore di questa regola.

f) Come si esprimono le regole in CSS?

2. Quali sono le differenze e le similitudini tra CSS e XSL?

DOM

7. Ci sono risultati che posso ottenere con la DOM e JavaScript e non con XSL? E risultati che posso ottenere con la DOM e JavaScript e non con CSS? Si spieghi perché.

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

Per DOM si intende un API che consente al linguaggio di programmazione\scripting di visualizzare il documento in questione con il modello ad oggetti, andando a definire una struttura ad albero in cui sono presenti le varie parti del documento, per poi essere caricate in memoria ed essere manipolate dal linguaggio di programmazione\scripting. Le DOM sono indipendenti dalla piattaforma e dal linguaggio di programmazione, infatti il consorzio W3C ha definito uno standard nelle DOM proprio per consentire un'interoperabilità che prescinde da piattaforma e linguaggio di programmazione.

9. Cosa rappresenta un nodo nella DOM?

Un nodo nella dom rappresenta un elemento del documento. Le DOM presuppongono l'utilizzo di 12 nodi, di cui 7 di questi utilizzati nelle DOM HTML. Tra cui ricordiamo:

Elemento: un nodo che corrisponde all'etichetta associata all'elemento

Attributo: un nodo che corrisponde all'attributo associato all'etichetta

Testo: un nodo che corrisponde ad una porzione di testo presente nel documento

Documento: un nodo che corrisponde al documento stesso

DocumentType: un nodo che corrisponde al DTD

Commenti: un nodo che corrisponde ai commenti HTML

Frammento: un nodo che corrisponde ad elementi esterni alla pagina HTML

7. Che differenza c'è tra parser a DOM e parser ad eventi?

4. Quali sono i tipi di nodo nella DOM?

Risposta sopra.

3. Si schematizzi la DOM per il seguente documento:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<manifest xmlns="http://beatniksoftware.com/tomboy">
```



```

<last-sync-date>2013-06-07T16:32:49.5390520+02:00</last-sync-date>

<last-sync-rev>0</last-sync-rev>

<server-id>185e3b59-f320-433c-b004-702fcf675337</server-id>

<note-revisions>

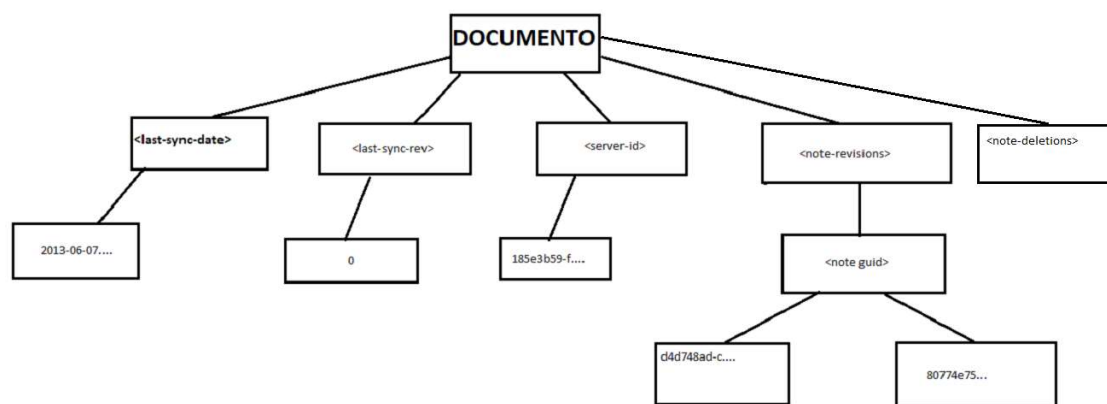
<note guid="d4d748ad-c78a-4bbf-92ad-82b136e873e6" latestrevision="0" />

<note guid="60774e75-165d-4e2e-bd17-cf577a5bcac4" latestrevision="0" />

</note-revisions>

<note-deletions/>

```



XML + XSL

8. Cos'è e per cosa viene usato l'oggetto XMLHttpRequest?

6. Come si dichiara un nuovo attributo in DTD?

Un attributo si dichiara in questo modo:

```

<!ATTLIST nomeElemento
    nomeAttributo tipo default1>

```

Dove, nomeElemento rappresenta l'elemento in questione, nomeAttributo è l'attributo che stiamo definendo, tipo è il tipo dell'attributo, e default1 è il valore che assume di default se non è assegnato nessun valore.

6. Cosa sono e come funzionano ID e IDREF?

Gli attributi ID definiscono in modo univoco un elemento, non possono avere valori ripetuti all'interno del documento. Ci si riferisce ad essi mediante l'attributo IDREF, il cui valore deve essere il valore di un attributo ID all'interno del documento. Si usa IDREFS se deve contenere uno o più IDREF. Inoltre, per il valore di un attributo ID, visto che deve essere sempre diverso, non può essere specificato un valore di default.

8. A cosa serve e come è fatto XSL?

5. Cosa è una DTD e come si associa ad un documento?

Per DTD, Document Type Definition, si intende un documento che descrive gli elementi utilizzabili in un documento XML e rappresenta una definizione di grammatiche per documenti XML. Per associare un DTD ad un documento XML è necessario dichiarare una DOCTYPE nel file XML in questo modo:

Per una dichiarazione interna abbiamo:

```
<!DOCTYPE nomeDocumento SYSTEM "documento.dtd">
```

Per una dichiarazione esterna del DTD abbiamo:

```
<!DOCTYPE nomeDocumento SYSTEM [  
  <!ELEMENT ... > ]
```

Per una dichiarazione mista abbiamo:

```
<!DOCTYPE nomeDocumento SYSTEM "documento.dtd" [  
  <!ELEMENT ...> ]>
```

5. Cos'è e a cosa serve XSL?

2. Si dica di che strumento si tratta e cosa significa la seguente riga:

```
<!ATTLIST camicia  
  taglia (S|M|L) #REQUIRED>
```

Si sta definendo un elemento in una DTD in cui si obbliga a selezionare una taglia che abbia un attributo S,M, oppure L

6. Cosa sono, a cosa servono e come funzionano i name spaces?

1. Quali sono le differenze tra HTML e XML?

Sia XML che HTML sono linguaggi di markup. Tuttavia, XML non soppianta HTML perché i due linguaggi sono stati progettati con scopi diversi: XML deve descrivere i dati: si concentra dunque su cosa sono i dati; HTML serve a visualizzare i dati: si concentra dunque su come appaiono i Dati.

4. Dato un documento XML che rappresenta un libro con titolo e una serie di capitoli, XSL permette di costruirne l'indice, CSS no. Si spieghi come si fa a costruire l'indice in XSL e perché non lo si può fare in CSS.

5. Dato un documento XML che rappresenta un libro con titolo e una serie di capitoli, XSL permette di costruirne l'indice, CSS no. Non volendo usare nessuno di questi due strumenti, come si può fare?

4. Cosa si intende per struttura fisica in XML?

Per struttura fisica, si intende il punto di vista di un unico file XML, che però può essere spezzettato in più file distinti, detti entità, che verranno in seguito ricostruiti da un parserXML

8. Quali sono i diversi tipi di parser XML?

7. Come funzionano in namespaces in XML?

ALTRO

10. Quali sono le differenze tra bot, spider e crawler?

9. Cosa è SOAP?

5. L'istruzione

`header ("Location: checkcookies.php?set=yes");`

coinvolge un header: di quale header si tratta? Qual è lo scopo di questa riga?

4. Si commenti la seguente istruzione:

`setcookie ('test', 'test', time() + 60);`

8. Cosa si intende per spider?

2. AJAX si fonda su un certo numero di strumenti, alcuni obbligatori, altri tipicamente utilizzati in questo framework: si spieghi quali sono e che funzione hanno in AJAX.

7. Qual è la differenza tra scoperta manuale e autonoma di un servizio web?

2. Si consideri il seguente frammento di codice:

`var status = "ON";`

```
function cambiaStato() {
var status ="OFF";

document.write("Nella funzione: " + status);
}

document.write("Fuori della funzione: " + status);
```

Di che linguaggio si tratta? Qual è il risultato della sua esecuzione?

Si consideri il seguente codice a cui fanno riferimento le domande da 1 a 6:

```
1. <%@ page import="java.util.*" %>
2. <HTML>
3. <BODY>
4. <%!
5. Date data = new Date();
6. Date getDate()
7. {
8. System.out.println( "All'interno del metodo getDate()" );
9. return theDate;
10. }
11. %>
12. Salve! In questo momento sono le <%= getDate() %>
13. </BODY>
14. </HTML>
```

1. Di che strumento si tratta? Da cosa si capisce? Si tratta di codice lato server o lato client?

2. Cosa produce questo codice e in che modo lo realizza?

3. Di che tipo è il blocco alla riga 1 ?

4. Di che tipo è il blocco alle righe 4-11 ?

5. E il blocco <%= getDate() %> alla riga 11?

6. Si commenti la seguente istruzione:

```
System.out.println( "All'interno del metodo getDate()" );
```

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

provvedere?

7. Cosa si intende per “Cloud computing”?

Si consideri il seguente codice a cui fanno riferimento le domande da (a) a (d):

```
15. function Employee() {
```

```
16. this.name = "";
```

```
17. this.dept = "general";
```

```
18. }
```

a) Di che linguaggio si tratta? Da cosa si capisce? Si tratta di codice lato server o lato client?

b) Usando il medesimo linguaggio, si implementi una classe Seller che estenda Employee,

ma abbia come ulteriore proprietà una funzione di stampa delle proprietà dell'oggetto.

c) Si diano le istruzioni che assegnano alla variabile `rossi` un oggetto Employee con name

`'Paolo Rossi'`.

d) E` possibile assegnare a tutti gli oggetti della classe Employee una proprietà specializzazione? Come si fa?

```
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. }
10. // -->
11. </script>
12. </head>
13. <body>
14. <H1>DisplayFormData</H1><H3>
15. <%
16. Enumeration inputParameters = request.getParameterNames();
17. if(!inputParameters.hasMoreElements()) { %>
18. <form method="POST" action="DisplayFormData.jsp">
19. <% for(int i = 0; i < 10; i++) { %>
20. Field<%=i%>: <input type="text" size="20"
21. name="Field<%=i%>"
22. value="Value<%=i%>">
23. <% } %>
24. <INPUT TYPE="button" name="bottone" value="Cliccami!"
25. onClick="alertUser()">
26. <INPUT TYPE=submit name=submit value="Submit"></form>
27. <%} else {
28. while(inputParameters.hasMoreElements()) {
29. String field = (String)inputParameters.nextElement();
30. String value = request.getParameter(field);
31. %>
32. <li><%= field %> = <%= value %></li>
33. <% }
34 } %>
35. </body>

```

36. </html>

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

6. Quali sono i tre diversi livelli nel cloud computing?

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

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

3. Come si indica una form in una site map nel formalismo di Conallen?

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

```
if ($_GET['set' != 'yes']) {  
    setcookie ('test', 'test', time() + 60);  
    header ("Location: checkcookies.php?set=yes");  
} else {  
    if (!empty(['test'])) {  
        echo "Cookies are enabled on your browser";  
    } else {  
        echo "Cookies are NOT enabled on your browser";  
    }  
}
```

1. Di che linguaggio si tratta? Da cosa si capisce? Si tratta di codice lato server o lato client?

2. Cosa si prefigge questo codice e in che modo lo realizza?

3. Che tipo di variabili sono \$_GET[] e \$_COOKIE[]? Qual è il loro ambito di definizione?

7. Cosa è un crawler e come funziona?

g) Cosa si intende per servizio web ed in quale linguaggio è implementato?

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?

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

3. Nel codice viene menzionata una risorsa joomla.css: di cosa si tratta e a cosa serve?

5. Cos'è e a cosa serve WSDL?

1. Si consideri il seguente frammento di codice:

```
<?
```

```
$assigned = $instance;
```

```
$reference =& $instance;
```

```
$instance->var = '$assigned will have this value';
```

```
$instance = null; // $instance and $reference become null
```

```
?>
```

Di quale linguaggio si tratta? Quali valori avranno alla fine le variabili

\$assigned, \$reference e \$instance?

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

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

5. Si parli di RDF.

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

(*) domande che non sono uscite nei compiti precedenti, ma con possibile uscita nei prossimi.

[HTTP] Cap.2

cos'è HTTP?

Il protocollo HTTP (HyperText Transfer Protocol) si può definire come il cuore del protocollo internet in quanto permette la comunicazione tra il client e server. Principalmente l'HTTP richiede solo due tipi di messaggio che sono: la richiesta del client verso il server e la risposta del server verso il client. Ogni messaggio è costituito da una prima riga detta request line (che ha metodi GET che richiede un oggetto, POST invia informazione, HEAD come il GET ma restituisce solo l'intestazione, PUT carica un oggetto, DELETE cancella un oggetto, TRACE e CONNECT), una seconda riga detta status line (200 OK, 301, 400 bad request, 404 not found, ecc.), l'intestazione (Connection: close) ed il body che può essere anche vuoto.

messaggi HTTP quali sono le informazioni contenute in un cookie? e cos'è un cookie?

Un cookie è un messaggio contenente l'host e path di un'applicazione, expiration date, nome e valore del cookie. Il cookie fa parte del meccanismo di session tracking ed esso serve per evitare che l'utente faccia delle operazioni ripetitive come quello di mettere la password ed username.

Come può il server cancellare un cookie?

Il server invia al client un cookie con i valori del nome, host e path del cookie precedente ma con un'expiration date negativo, in modo da sovrascrivere il cookie precedente ed infine cancellarlo.

si spieghi come HTTP offre un meccanismo di caching

HTTP offre un meccanismo di caching tramite il campo 304 Not Modified. Ogni volta che il client fa richiesta HTTP al server, esso risponde al client a seconda se sono avvenute modifiche dalla data riportata dal client nella sua ultima richiesta. Se non è avvenuta nessuna modifica allora verrà inviato il campo lo status line 304 dal server al client.

si spieghi come HTTP offre un meccanismo di autenticazione

Il meccanismo di autenticazione avviene tramite una richiesta da parte del server: il server invia una risposta con status line 401 Unauthorized Request, ove chiede che il client inserisca username e password. Una volta che il client invia i dati, lo user agent salva i dati ed ogni qualvolta invia una richiesta HTTP, l'inserisce insieme alla richiesta.

Come si presenta una richiesta HTTP?

GET /books/search.aspx?query=23k9j HTTP/1.1

Accept: image/gif, image/xbitmap, image/jpeg, image/pjpeg,

Accept-Language: en-gb,en-us;q=0.5

Referer: http://miaapplicazione.miosito.com/books/default.aspx

User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)

Host: miosito.com

Cookie: lang=it; XSESSIONID=sklk87979HHIhhkhJKNAH8768983

La prima riga indica il metodo GET assieme all'URL della risorsa richiesta ed il protocollo di http utilizzato. Le linee seguenti indicano le tipologie accettate e le lingue accettate. La terza riga indica l'URL da dove parte la richiesta mentre la quarta riga indica il browser utilizzato.

[CGI] Cap.3

cos'è? cosa fa?

Le CGI (Common Gateway Interface) è un tipo di meccanismo lato server per l'avvio di programmi.

Come si passano i parametri di ingresso e di uscita in un programma CGI?

I parametri d'ingresso vengono passati tramite due metodi e sono lo "stdin" (richiesta di tipo POST) e tramite variabili d'ambiente (richiesta di tipo GET). La prima richiesta viene soddisfatta conoscendo la lunghezza del body tramite una variabile d'ambiente CONTENT_LENGTH in quanto non usa EOF, mentre nella seconda richiesta i campi vengono riempiti ove c'è "?".

La gestione degli output avviene tramite stdout, il server Web prepara la risposta HTTP includendo lo status line HTTP/1.1 200 OK e l'output del programma che contiene l'intestazione

Posso accedere alle informazioni contenute nell'intestazione della richiesta HTTP da un

programma CGI? Se sì, in che modo?

Sì, utilizzando la comunicazione HTTP: si utilizza il metodo HEAD durante la comunicazione tra client-server per accedere alle informazioni della richiesta HTTP.

come si fa ad impostare il content-type della risposta con CGI?

cgi e cookie, cosa si può dire (*)

Devo implementare un sistema basato su CGI che prende in ingresso la matricola di uno studente e ne restituisce nome e cognome:

a. In che linguaggio andrà implementato lo script?

b. All'interno dello script, come ottengo il numero di matricola e come restituisco nome e

cognome dello studente?

a. Lo script può essere implementato in un qualsiasi dei seguenti linguaggi: C, C++, Fortran, Perl, TCL, ecc.

b. Per ottenere il numero di matricola si utilizzano o variabili d'ambiente o stdin. Tipo GET: come la QUERY_STRING (i parametri da inserire sono seguiti da un punto interrogativo "?"). Tipo POST: tutto ciò

che si trova nel body della richiesta HTTP viene passato al processo CGI sul stdin. La risposta HTTP da inoltrare allo user agent viene preparata utilizzando lo stdout del processo

CGI, a cui il server permette esclusivamente la riga di stato. Ricapitolando:

- lo user agent invia una richiesta HTTP al server;
- il server web la riconosce come programma Common Gateway Interface, e quindi
- il server web manda in esecuzione il programma (passando i parametri come spiegato sopra);
- se l'esecuzione del programma produce dei risultati (ma il caso contrario è solo teorico!),

questi vanno organizzati sullo stdout;

- il server Web prepara la risposta HTTP da inviare al client includendo:

– status line, ad esempio, HTTP/1.1 200 OK

– output del programma (che deve quindi contenere anche il resto dell'intestazione –

header – HTTP)

[HTML] Cap.4

cos' è?

L'HTML (HyperText Markup Language) è un linguaggio di markup nato per la rappresentazione di dati. Segue dei nodi, detti “elementi”, ed è costituito da una struttura ad albero.

quali sono le principali innovazioni introdotte da HTML5?

HTML5 introduce:

- definisce un linguaggio HTML standard che utilizza sintassi di HTML ed XML;
- definisce dettagliati modelli di processo per favorire la creazione di applicazioni interoperabili;
- migliora il mark-up per i documenti;
- introduce nuove etichette e API per varianti dell'HTML che stanno emergendo, tra cui le applicazioni web
- Storage: similmente ai cookie memorizza le stringhe come nomi

differenza tra gli attributi name e id in HTML, com'è formato?

Gli attributi sono caratterizzati da un "name" ed un "id", il name può essere utilizzato da uno o più attributi mentre l'id è univoco.

cross-scripting, clickjacking e same-origin policy (sicurezza nel web) (*)

Per evitare che gli user-agent avviino contenuti malevoli, si fa utilizzo di questo meccanismo di sicurezza, ovvero i contenuti avviati dallo user agent sono contenuti che derivano da una stessa origine.

Con il termine "cross-site scripting attack" si vuole intendere un pezzo di codice malevolo, inserito all'interno di un sito tramite l'utilizzo di form. Questi frammenti di codice possono alterare i dati presenti sul server con lo scopo di sottrarre dati utili.

Il "clickjacking" è un metodo utilizzato da hacker per sottrarre il click di un utente per fini loschi. L'inganno più utilizzato è quello di produrre un frame piccolo sotto il mouse dell'utente in maniera improvvisa, per ovviare il problema viene usata una pagina critica che chiede all'utente se vuole confermare la procedura.

differenze tra HTML e CSS (*)

[CSS] Cap.5

cos'è?

Il CSS (Cascading StyleSheets) è un linguaggio dichiarativo che serve alla formazione di diversi fogli di stile, l'organizzazione avviene a livelli ed ogni livello ha diverse definizioni della pagina in maniera

dettagliata. Principalmente il CSS si occupa quindi dello stile della pagina.

che sintassi possono seguire i selettori in CSS?

I selettori possono seguire due sintassi e sono l'HTML o l'XML.

regole CSS in CSS posso imporre che tutti gli item di una lista numerata vengano scritti in rosso e tutti quelli di una lista non numerata in verde? in che modo?

Le regole permettono di associare delle caratteristiche ad un documento in ingresso, caratteristiche come dimensione, colore, ecc.). Una regola è costituita da una sopraregola in quanto le regole formano una struttura ad albero. Inoltre una regola è formata da due parti e sono il "selettore" ed una "dichiarazione", la dichiarazione a sua volta è suddivisa in "proprietà"(color) e "valore"(blue). Modificando opportunamente i selettori si può arrivare a colorare una lista o tutta di rosso o tutta di verde, numerata o non.

in CSS posso imporre che il primo elemento di una lista venga scritto in rosso e tutti gli altri in verde? in che modo?

???

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.

???

Si scriva una riga CSS che stampi in grassetto tutti gli elementi di una lista ordinata che è parte di un'altra lista a livello superiore.

???

differenze tra XSL e CSS (CSS e XSL)

Principali differenze tra XSL e CSS:

1. XSL segue la sintassi di XML, CSS usa una sintassi a parte;
2. CSS non può modificare la struttura dell'albero in modo sostanziale, XSL sí questo in particolare implica che CSS sia adatto ad essere usato, oltre che su documenti HTML, anche su documenti XML purché abbiano una struttura immediatamente convertibile nella presentazione;
3. CSS è ad un grado di maturità maggiore.

In realtà alcuni degli oggetti di formattazione e molte delle proprietà usate in XSL derivano

direttamente dalle specifiche CSS2

[JavaScript] Cap.6

cos'è? differenze con Java esistono i tipi di valore in JavaScript? se si, quali sono?

JavaScript venne ideato con l'intento di manipolare oggetti computazionali all'interno di un ambiente ospite. È stato creato per essere un linguaggio per il web, un linguaggio di scripting per client ed server, esso ha le caratteristiche principali di un linguaggio di scripting per ambienti ospite diversi.

Java si differenzia da JavaScript per moltissime cose: è un linguaggio di programmazione, è stato ideato per diversi scopi, è compilato controlla gli accessi a più livelli, verifica gli array, ha una gerarchia delle classi.

Java	JavaScript
Usato per scopi diversi	Solo per programmazione Web
Compilato	Non compilato
Codice compilato in applets separati dal sorgente (ad es. HTML)	Codice direttamente incluso nel sorgente (ad es. HTML)
Linguaggio di programmazione	Linguaggio di scripting
Controllo dei tipi delle variabili forte	Controllo dei tipi delle variabili debole
Controllo di accesso a più livelli (private, protected, public)	Controllo di accesso semplice
Rigida verifica degli array	Nessuna verifica degli array
Gerarchia delle classi	Gerarchia delle istanze

Tabella 12.1: Principali differenze tra Java e JavaScript

scope delle variabili in JS (*)

scope di applicazione

- application: il contenitore all'interno del quale viene eseguita la servlet;

scope di pagina

- config
- exception: solo in una pagina di errore
- out
- page
- pageContext
- response

scope di richiesta

- request

scope di sessione

- session

si consideri il seguente pezzo di codice:

```
1 function sum(a,b) {  
2     c= a+b;  
3 return c;  
4}  
5
```

```
6 var c= 'pippo';
```

```
7 sum(4.5, 7.3);
```

```
8 document.writeln(c);
```

motivare le risposte:

di che linguaggio si tratta?

Si tratta di Javascript.

qual'è il tipo della variabile c? Rimane lo stesso lungo tutto il codice?

Le variabili c sono due e sono due variabili locali, quindi le variabili non cambiano di tipo durante il codice.

cosa viene stampato nella riga 8?

Verrà stampato "pippo" perché alla funzione sum() non viene assegnata una variabile dov esalvare il ritorno

ereditarietà in JavaScript incapsulamento (*)

L'ereditarietà è basata sui prototipi, e non sulle classi come in linguaggi orientati agli oggetti come Java o C++. In JavaScript, infatti, attributi e metodi (quindi le proprietà) vengono ereditati solo dal costruttore. Ad ogni costruttore è associata una proprietà prototype usata per implementare l'ereditarietà di struttura, stato e comportamento. Quando viene creato un oggetto, ad esso viene associato un riferimento implicito al prototipo del suo costruttore.

Per incapsulamento si intende un meccanismo in cui i dati parte di un oggetto dovrebbero rimanere privati ed essere manipolati solo attraverso dei metodi disponibili pubblicamente. In altre parole, i programmi non dovrebbero avere la possibilità di manipolare direttamente gli attributi di un oggetto. Le versioni più recenti di JavaScript permettono la costruzione di funzioni annidate, usando le quali è possibile nascondere le informazioni all'interno degli oggetti, per farlo occorre omettere la parola chiave this nelle informazioni che devono rimanere private.

Si dice che javascript è basato sugli oggetti e non orientato ad oggetti: si spieghi quali sono le caratteristiche orientate ad gli oggetti in JavaScript e soprattutto le loro limitazioni

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. Uno degli svantaggi della programmazione basata agli oggetti è quella di non poter implementare l'incapsulamento, e quindi l'information hiding. (giusto?)

in JavaScript posso assegnare ad un elemento di un vettore la stringa "diciassette" e ad un altro il numero 17? e in java? in caso di risposta/e affermativa/e, si spieghi come si fa

Sì, in Javascript è possibile inserire in un array sia caratteri che numeri in quanto Javascript li riconosce come primitive. Questo tipo di array sono detti array associativi e seppur fattibile in Javascript utilizzarli, non sono consigliati perché potrebbero generare casi di errore. In Java non vengono usati.

Si consideri il seguente frammento di codice:

```
var status= "ON";

function cambiaStato(){

    var status= "OFF";

    document.write("Nella funzione: " + status);

}

document.write("Fuori dalla funzione: " + status);
```

di che linguaggio si tratta? qual'è l'output della sua esecuzione?

Il linguaggio in questione è Javascript. Nella funzione interna verrà stampato OFF mentre nella funzione esterna verrà stampato ON

local storage, session storage, delete web storage (web storage)

I metodi di caching utilizzati in JS sono divisi in due parti:

local storage: salva grosse quantità di dati, che vengono salvati nel browser e rimangono salvate finché non se ne fa richiesta di rimuoverli.

Session storage: dati che vengono salvati dal browser ma che vengono eliminati nel momento in cui si chiude la sessione. Sono più adatti per piccole quantità di dati.

Delete web storage: (?)

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

```
public Class Employee {

    public String name;
```

```

        public String dept;

        public Employee () {

            this.name = "";

            this.dept = "general";

        }

    }

    public class SalesPerson extends WorkerBee {

        public double quota;

        public SalesPerson() {

            this.dept = "sales";

            this.quota = 100.0;

        }

    }

```

Basta eliminare i "this" (???)

Cosa si intende per dichiarazioni in JSP?

In JSP le dichiarazioni si utilizzano i simboli "<%! ...%>" sono usati per la dichiarazioni di variabili e metodi, che diventano membri della classe ottenuta dalla traduzione della JSP.

[DOM] Cap.7

Cos'è la DOM? se ne descrivano gli aspetti più importanti. In che linguaggio è implementato?

La DOM (Document Object Model) è un interfaccia indipendente dal browser e dai linguaggi di programmazione che serve per la funzione di alcune applicazioni. La DOM permette ai programmatori della gestione dinamica dei documenti, rielaborando gli input e modificando i documenti in tempo reale.

E' implementato in linguaggio HTML o XML e la sua struttura è detta a "boschetto" in quanto è costituita da diversi alberi. Gli alberi contengono elementi che, se non sono la radice, ad ogni elemento ne

corrisponde un altro al di sopra di esso.

Ogni nodo è di una tipologia diversa dall'altra, i tipi sono: elemento, attributo, testo, commento, document, tipo di documento, frammento. Ogni nodo ha un indice che lo caratterizza.

DOM del W3C (*)

Il DOM W3C prevede 12 tipi di nodo di cui 7 si applicano ai documenti HTML.

Gestione degli eventi (*)

Per evento s'intende una reazione programmata che avviene in maniera asincrona in corrispondenza ad un'azione. Gli eventi nel DOM sono: eventi UI (sono eventi causati dai dispositivi degli utenti, tipo mouse, ecc.), eventi logici UI (non dipendono dal dispositivo) ed eventi di mutazione che sono causati da azioni che modificano la struttura del documento. Ogni evento è associato ad un oggetto che è il suo event target.

Cosa rappresenta un nodo nella DOM?

Ogni nodo della DOM è un oggetto con moduli e proprietà.

si schematizzi la DOM per il seguente documento XML

```
<?xml version="1.0">

<catalog>

  <book id="bk161">

    <author>Gambardella, Matthew</author>

    <title>XML developer's Guide</title>

    <genre>Computer</genre>

  </book>

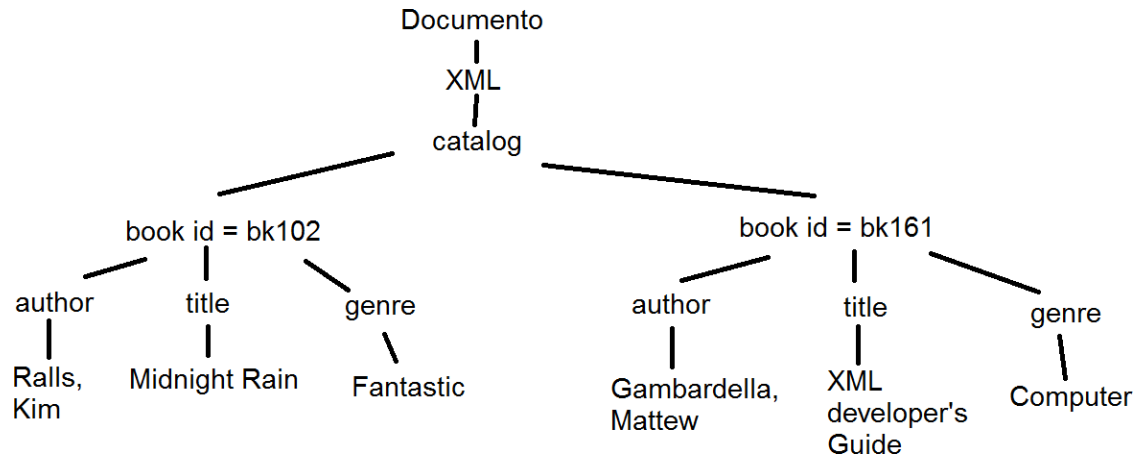
  <book id="bk102">

    <author>Ralls, Kim</author>

    <title>Midnight Rain</title>

    <genre>Fantastic</genre>
```

`</book>`
`</catalog>`



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">
```

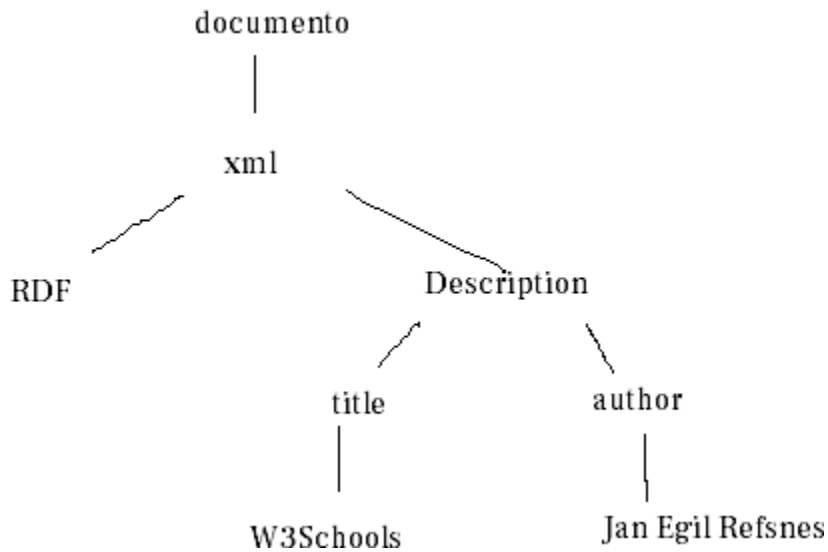
```
    <si:title>W3Schools</si:title>
```

```
    <si:author>Jan Egil Refsnes</si:author>
```

```
  </rdf:Description>
```

```
</rdf:RDF>
```

Si schematizzi la DOM del documento.



[CMS e WEB FRAMEWORK] Cap.8

cos'è un CMS?

CMS (Content Management Systems) come suggerisce il nome stesso serve per la gestione di contenuti, come film, audio, immagini. Il CMS venne ideato con l'intento che l'utilizzo fosse dedicato a più persone e che ogni persona non avesse competenze di tipo informatiche.

cos'è un Web Framework?

I web framework non sono altro che framework utilizzati per il web, dai programmatori durante lo sviluppo di pagine web per la programmazione lato client e lato server. I framework vengono utilizzati per non far ripetere ad un programmatore più volte uno stesso codice tramite il principio DRY (don't repeat yourself). Quindi sono strumenti che rendono standard lavori.

Cosa si intende per architettura a tre strati (three tier)

Per architettura a tre strati si vuole intendere la forma più semplice di struttura di una qualsiasi applicazione web. Essa consiste nel primo strato dato dall'user agent (richiede i dati), la seconda parte dal server web (rende presentabile i dati all'utente tramite HTML, XML, ecc.) e l'ultima struttura Information Source (analizza l'affidabilità dei dati).

modello MVC (*)

il modello MVC è un tipo di architettura three tier. Essa è divisa in:

Model: dove risiedono i dati;

View: vengono visualizzati i dati;

Control: logica che s'interpone tra la visualizzazione dei dati e dove risiedono.

[JAVA] Cap.9

Si dice che Java è un linguaggio sicuro: quali sono le caratteristiche che lo rendono tale, in particolare in relazione alle applicazioni web?

Java è ritenuto un linguaggio sicuro per i seguenti motivi:

- Garbage collector;
- Non utilizza aritmetica dei puntatori;
- Fortemente tipato;
- Gestisce gli errori e l'eccezioni.

Qual è il ciclo di vita di una applet?

Init(): inizializza l'applet;

start(): viene utilizzato ogni qualvolta si fa uso della applet e può essere richiamato solo dopo init() oppure stop(). Avvia il thread;

stop(): ferma la applet, è un metodo che non dev'essere per forza richiamato esplicitamente;

destroy(): distrugge la applet;

cos'è un applet?

E' uno strumento utilizzato nella programmazione lato client. E' un programma contenuto in un altro programma.

[SERVLET] Cap.10

Ciclo di vita di una servlet.

Una servlet ha un ciclo di vita che si può definire simile alle applet: quando il client ha bisogno di una risorsa allora viene inizializzata la servlet tramite il metodo init(), una volta inizializzata viene utilizzato il metodo service() ogni qualvolta viene fatta una richiesta. Se le richieste si accavallano all'ora vengono attivati dei thread (qui avviene il risparmio rispetto alle CGI che attiva dei processi ad ogni richiesta). Infine quando tutte le richieste vengono soddisfatte si chiama il metodo destroy() per terminare e distruggere la servlet.

Quali sono i vantaggi e gli svantaggi dell'uso delle servlet rispetto al CGI?

Le servlet sono incredibilmente vantaggiose rispetto alla CGI in quanto, invece di attivare un processo ad ogni accesso al database come fa la CGI, le servlet attivano un solo processo ed utilizzano una moltitudine di thread che accedono in contemporanea al database.

Inoltre le servlet, utilizzando i thread, hanno accesso a variabili comuni e possono comunicare con programmi esterni. Tale comunicazione non avviene nelle CGI.

cos'è una servlet? (*)

Una servlet è una tecnologia lato server utilizzata per soddisfare le richieste del client.

dove si scrivono i messaggi di errore in una servlet?

Metodi dei servlet (*)

L'interfaccia HttpServletRequest e HttpServletResponse (*)

6.1.4 L'interfaccia HttpServletRequest

I metodi dichiarati da questa interfaccia provengono in parte dall'interfaccia ServletRequest. Un'implementazione dell'interfaccia è data dalla classe HttpServletRequestWrapper.

- `String getParameter(String name)`: torna il valore del parametro *name* passato nella get/post;
- `Enumeration getParameterNames()`: returns an enumeration of string objects containing the names of the parameters contained in this request. "
- `String[] getParameterValues(String name)`: se *name* ha valori multipli
- `Cookie[] getCookies()`: memorizzati dal server nel client;
- `HttpSession getSession(boolean create)`: se l'oggetto non esiste e *create* = *true*, allora lo crea.

6.1.5 L'interfaccia HttpServletResponse

- `void addCookie(Cookie cookie)`: nell'header della risposta passata al client; essa viene memorizzata a seconda dell'età massima e dell'abilitazione o meno ai cookie disposta nel client;
- `ServletOutputStream getOutputStream()`: stream basato su byte per dati binari;
- `PrintWriter getWriter()`: stream basato su caratteri per dati testuali;
- `void setContentType(String type)`: specifica il formato MIME.

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. <ul>
20. <%
21. for (int i = 0; i < authors.length; ++i) {
22. %>
23. <li><%= authors[i] %></li>
24. <%
25. }
26. %>
27. </ul>
28. <a href="<%= request.getRequestURI() %>">BACK</a>
29. <%
30. }
31. %>
32. </body>
```


33. </html>

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

E' un documento html. Il linguaggio come si può vedere è costituito da etichette e valori, le etichette sono chiuse tra le virgolette e la loro chiusura è indicata tramite la barra "/", mentre i valori sono racchiusi tra le etichette.

2. Si dettagli il funzionamento del codice dato sopra.

La pagina chiede all'utente di selezionare uno o più autori. Se l'utente seleziona un autore allora la pagina stamperà a video "You have selected autohr(s)" e dopodiché stamperà a video il nome degli autori selezionati, che sono salvati momentaneamente nella stringa.

3. Si costruisca una servlet equivalente.

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

public class ServletsRule extends HttpServlet {

    public void service(HttpServletRequest req, HttpServletResponse res) \
    throws IOException {

        res.setContentType("text/html");

        PrintWriter out = res.getWriter();

        out.print("<HEAD>");

        out.print("<TITLE> test </TITLE>");

        out.print("</HEAD>");

        out.print("\section{Servlets Rule! " + i++);

        out.print("</BODY>");

        out.close();

    }

}
```

....

[SESSION TRACKING] Cap.11

Cos'è e come funziona il session tracking mediante campi nascosti.

Durante la comunicazione client-server si utilizzano delle form non visibili all'utente in cui si scambiano informazioni.

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

I campi nascosti utilizzano le form per l'immissione dei dati, ed alcuni browser non accettano le form. Ciò significa che le variabili di sessione, sono più elastiche sotto questo punto di vista.

cosa si intende per "session tracking" e quali sono i modi per affrontarlo?

Per session tracking si vuole intendere il meccanismo che viene utilizzato in internet per il salvataggio di dati.

cosa sono, a cosa servono e come funzionano le variabili di sessione

Le variabili di sessione sono variabili utilizzati dal lato server che contengono username password dell'utente che ha effettuato l'accesso. Il server, dopo il primo accesso da parte dell'utente, invierà un identificativo unico al client e quest'ultimo lo utilizzerà ogni volta che vuole essere riconosciuto.

info contenute in un cookie

?

il meccanismo di autenticazione può essere implementato in due modi diversi: il primo sulle caratteristiche dell'HTTP, il secondo no. si spiegano questi due diversi approcci.

Il meccanismo di autenticazione può essere implementato o tramite le variabili di sessione, dove i dati vengono salvati nel lato server, o tramite il lato client: il browser mostra una pagina di pop-up ove inserire i dati e successivamente salvarli nel browser. Questi dati vengono inviati ad ogni richiesta.

per il meccanismo di session tracking mediante campi nascosti occorre supporto dell'HTTP? si spieghi nel dettaglio come funziona

?

meccanismo session tracking mediante autenticazione

Il meccanismo di autenticazione del session tracking funziona così: il server fa una richiesta HTTP (401) di autenticazione del client, l'utente è tenuto ad inserire i dati o tramite un pop-up o nella pagina principale.

riscrittura dell'URL (*)

Un altro metodo di session tracking che si usa nel web è quello della sovrascrittura dell'URL. Per

mantenere la sessione, l'URL originale viene modificato tramite l'aggiunta di un ID.

[JSP & Struts2] Cap.12

Cosa sono e come si presentano le JSP?

Le JSP (Java Server Pages) sono una metodologia di programmazione lato server che spesso si affianca e si confonde con le servlet. Le JSP sono pagine in Java tradotte in servlet, che fanno uso della JVM.

A proposito di JSP, cosa si intende per "oggetti impliciti"? e quali sono i loro possibili ambiti di definizione (scope)?

Gli oggetti impliciti delle JSP sono oggetti che il programmatore può utilizzare direttamente senza esplicitare la dichiarazione. Lo scope:

- applicazione (dove viene eseguita servlet. L'oggetto implicito application consente di accedere alle costanti dell'applicazione e di memorizzare oggetti a livello di applicazione e quindi accessibili da qualsiasi utente per un tempo che va dall'avvio del motore JSP alla sua chiusura, in pratica fino allo spegnimento del server. Gli oggetti memorizzati nell'oggetto application come appena detto sono visibili da ogni utente e ogni pagina può modificarli.),
- pagina (config, exception, out page, pageContext, response),
- richiesta (request, permette di accedere alle informazioni di intestazione specifiche del protocollo http, al momento della richiesta incapsula le informazioni del client e le rende disponibili tramite alcuni metodi)
- sessione (session)

Si descriva il funzionamento di una JSP.

Una JSP è una pagina di testo formata da un template statico espresso in qualsiasi linguaggio, con

istruzioni Java. La prima volta che viene chiamata viene tradotta in servlet (per indicare al server la rielaborazione si usa il formato .jsp), in seguito viene caricata la classe da un class loader da un qualsiasi servlet engine. Inoltre è contenuta in una sandbox che serve ad evitare che questa prenda informazioni del sistema.

Quali sono i modi per introdurre istruzioni Java in una JSP?

Per introdurre istruzioni di tipo Java in una JSP serve utilizzare come tag l'etichetta `<%scriptlet....%>`

Qual è la differenza tra dichiarazioni ed espressioni nelle JSP?

Le dichiarazioni sono sempre introdotte da `"<%! ... %>"` mentre le espressioni sono introdotte da `"<%= ... %>"`.

Quali sono le differenze tra scriptlet e dichiarazioni in una JSP?

Le scriptlet scritte `"<%...%>"` sono istruzioni che vengono eseguite durante l'elaborazione della richiesta HTTP ed al suo interno si possono fare dichiarazioni il cui scope è solamente interno alle etichette. Le dichiarazioni `"<%!...%>"` servono per la definizione di variabili ed il loro scope è utilizzabile in tutto il codice.

Cosa sono le Javabeans (java bean)

In informatica le JavaBean (letteralmente, chicchi di Java) sono classi scritte in linguaggio di programmazione Java secondo una particolare convenzione. Sono utilizzate per incapsulare più oggetti in un oggetto singolo (il bean), cosicché tali oggetti possano essere passati come un singolo oggetto bean invece che come multipli oggetti individuali.

Al fine di funzionare come una classe JavaBean, una classe di un oggetto deve obbedire a certe convenzioni in merito ai nomi, alla costruzione e al comportamento dei metodi. Queste convenzioni rendono possibile avere tool che possono usare, riusare, sostituire e connettere JavaBean.

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).

```
<! esercizio.jsp
```

```
<html>
```

```
<body>
```

```
    <% out.println("Ciao Francesco: sei il 100° visitatore da 12/02/19"); %>
```

```
</body>
```

```
</html>
```

[Web UML] Cap.13

Cos'è UML e L'estensione di Conallen? (*)

L'UML è il linguaggio utilizzato dal disegnatore durante la progettazione di un qualsiasi programma o sito. L'estensione di Conallen prevede che il progettista costruisca anche la "client page" e la "server page". La server page si occupa delle normali procedure da server (risorse, ecc) ed inoltre della costruzione della client page.

Come si indica una form in una site map nel formalismo di Conallen?

Con il termine "form" si vuole indicare un tipo di pagina aventi rispetto alle altre pagine delle proprietà aggiuntive.

Server Page e Client Page (*)

La server page è la pagina server che si occupa della gestione degli accessi da parte del client e della gestione delle risorse. La Client page a differenza della prima si occupa dell'interfaccia, il suo codice viene letto dallo user agent (come il browser), le operazioni corrispondono alle funzioni mentre gli attributi alle variabili.

Cosa si intende per "Redirect" e cosa si intende per "link" nell'estensione di Conallen? (*)

I collegamenti tra le pagine sono molteplici e tra questi vi sono il "redirect" che associa due pagine client o server che indica che il controllo possa passare da entrambe le parti. Il collegamento "link" invece

collega una pagina client ad un'altra pagina client, però da notare che ad ogni pagina client collegata implicitamente vi è collegata una pagina server in quanto una qualsiasi richiesta di risorse ad una pagina client corrisponde una richiesta alla pagina serve associata.

[XML e XHTML] Cap.14

Cosa si intende per "ben formatezza" e "validità"? In che contesto?

"Ogni documento XML e HTML sono anche documenti SGML, ma un documento HTML può non essere un documento XML. La sintassi del markup permette di dire se un documento è o meno ben formato. Un documento XML si dice valido se segue le regole di una DTD o di uno Schema che dicono quali informazioni devono esserci e in quale posizione."

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

Seppure entrambi i linguaggi sono linguaggi di markup le loro differenze sono sostanziali: HTML provvede all'interfaccia dei dati, mentre XML si occupa della descrizione dei dati. Prima di XML si utilizzavano programmi diversi ogni qualvolta si volessero descrivere i dati.

Cosa è una DTD e come si associa ad un documento?

Un DTD (Document Type Definition) è una descrizione formale della grammatica di un linguaggio di markup. Esso descrive gli elementi utilizzabili in un documento XML, la loro reciproca relazione nei confronti della struttura del documento e altre informazioni sugli attributi di ciascun elemento.

Si commenti la seguente istruzione:

<!ELEMENT paragrafo (#PCDATA | bold)+ >

di che strumento si tratta? Cosa si prefigge?

Si tratta di un documento di XML, e dichiara un elemento chiamato paragrafo che il contenuto del documento è testuale (#PCDATA è il content model) ed è in grassetto

Si dica di che strumento si tratta e cosa significa la seguente riga:

<!ATTLIST camicia taglia (S|M|L) #REQUIRED>

La prima parola "<!ATTLIST ... >" è della sintassi della XML e si prefigge di dichiarare degli attributi il nome è "camicia taglia" ed il suo valore è compreso tra "(S|M|L)". La parola #REQUIRED indica che il valore dell'attributo dev'essere specificato obbligatoriamente.

Cosa si intende per struttura fisica in XML?

Qualsiasi struttura logica di un documento XML viene tradotta in corrispondenza in una struttura fisica, questa può risiedere anche in parti hardware distanti fra loro.

cosa si intende per struttura logica in XML? (*)

Per struttura logica s'intende la struttura ad albero che si viene a formare dai vari elementi in quanto gli elementi sono caratterizzati in maniera gerarchica. Il primo elemento non ha padre ed è detto radice, mentre al di sotto si trovano i figli.

Sintatticamente ogni elemento è "aperto" e "chiuso" da delle etichette.

Cosa sono e come funzionano ID e IDREF?

Gli attributi ID sono attributi che identificano in maniera univoca un attributo, si può far riferimento ad esso tramite l'attributo IDREF. Se gli IDREF contenuti sono più di uno allora si utilizza l'attributo IDREFS.

Quali sono i diversi tipi di parser XML?

I parser derivano dalla parola inglese "parsing" il cui significato è "controllare flussi di". Nel caso dell'informatica il parser è un software che si occupa del controllo sintattico di vari codici, delle volte tramite l'utilizzo di alberi. I parser si dividono in parser DOM e parser ad eventi.

Come si dichiara un nuovo attributo in DTD?

Con la parola <!ATTLIST nomeattributo ... >

cos'è XHTML? vantaggi e svantaggi

che differenza c'è tra parser a DOM e parser a eventi?

Al parser basato a DOM si fornisce in input un documento e ne restituisce un oggetto di tipo albero. La pecca è che tale metodo non può essere utilizzato per oggetti di grandi dimensioni, per risolvere il problema si fa uso del metodo parser basato su eventi che, a differenza del primo, divide il documento in diverse parti detti "eventi". Tale metodo non è applicabile sempre perché perde la visione intera del documento poiché lo frammenta in tante porzioni di dati.

Cosa si intende per portlet?

I portlet sono delle "piccole finestre" che all'interno di una pagina web servono per ottimizzare le prestazioni della pagina, poiché funzionano in caso queste vengano attivate. Sono molto simili alle servlet ma a differenza delle prime non possono comunicare col browser od inviare redirect o scrivere markup.

[Strumenti legati a XML] Cap.15

Cosa sono, a cosa servono e come funzionano i namespaces?

Nel momento in cui si vuole utilizzare documenti XML con DTD, per evitare che si utilizzino elementi con lo stesso nome, si fa uso di una tecnica di nome “namespaces” (spazio dei nomi), ove si fa uso degli spazi per identificare elementi diversi.

Cos'è e a che serve XMLSchema (*)

L'XMLSchema è una pagina con estensione “.xml” che serve per la descrizione di altre pagine xml. Essa costruisce un albero gerarchico per la raffigurazione di una pagina xml.

Essa ha due tipi di dato:

- primitivi;
- derivati;

[XLS] Cap.16

A cosa serve e come è fatto XSL?

XSL (eXtensible Stylesheet Language) si riferisce ad una famiglia di linguaggi (XSLT, XpTh e XSL-FO) il cui compito è la rappresentazione dei dati. Viene fornito uno schema XML ed uno stile XSL, in seguito lo schema verrà:

- formattato: ovvero adattato al dispositivo su cui dovrà essere rappresentato;
- trasformato: l'albero di partenza verrà trasformato in un nuovo albero;

L'XSL è costituito da regole (come in CSS) di stile che si dividono in:

- pattern: indica da dove avverrà la modifica dell'albero;
- template: permette di ricostruire parte dell'albero risultante;

Dato un documento XML che rappresenta un libro con titolo e una serie di capitoli, XSL

permette di costruirne l'indice, CSS no. Non volendo usare nessuno di questi due strumenti,

come si può fare?

???

Ci sono risultati che posso ottenere con la DOM e JavaScript e non con XSL? E risultati che posso

ottenere con la DOM e JavaScript e non con CSS? Si spieghi perché.

[PHP] Cap.17

Cosa si intende per variabili superglobali in PHP?

Per variabili "superglobali" si vuole intendere delle variabili che sono visibili in tutto lo script (funzioni, fuori dalle funzioni, ecc.)

In PHP ad un certo punto sono state introdotte delle caratteristiche orientate agli oggetti: in che modo e con quali limitazioni.

PHP5 è la versione di PHP che introduce la programmazione orientata ad oggetti. Come tutti i linguaggi orientati ad oggetti introduce classi, ereditarietà, ecc. Una limitazione è che il nome della classe deve corrispondere al nome del file, inoltre a differenza degli altri linguaggi gli oggetti appartenenti ad una stessa classe potranno accedere reciprocamente ai membri privati e protetti anche se non sono della stessa istanza.

Quali sono gli ambiti di definizione (scope) delle variabili in PHP? Come sono definiti?

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 funzioni in cui sono definite).

Si consideri il seguente frammento di codice:

<?

```
$assigned = $instance;
```

```
$reference =& $instance;
```

```
$instance->var = '$assigned will have this value';
```

```
$instance = null; // $instance and $reference become null
```

?>

Di quale linguaggio si tratta? Quali valori avranno alla fine le variabili

\$assigned, \$reference e \$instance?

???

Si consideri il seguente codice a cui fanno riferimento le domande da 1 a 5:

```
if ($_GET['set'] != 'yes') {  
  
    setcookie ('test', 'test', time() + 60);  
  
    header ("Location: checkcookies.php?set=yes");  
}  
else {  
  
    if (!empty(['test'])) {  
  
        echo "Cookies are enabled on your browser";  
    } else {  
  
        echo "Cookies are NOT enabled on your browser";  
    }  
}
```

1. Di che linguaggio si tratta? Da cosa si capisce? Si tratta di codice lato server o lato client?

Si tratta del linguaggio PHP e lo si capisce poiché la variabile "GET" è preceduta dal simbolo "\$" tipico del linguaggio PHP. Si tratta di un codice lato client

2. Cosa si prefigge questo codice e in che modo lo realizza?

Questo codice si prefigge di vedere se il server accetta i cookies. Utilizza il metodo setcookie() dove invia un ping ed attende un tempo prestabilito. Se 'test' è diverso da vuoto allora non sono abilitati i cookie e viceversa

3. Che tipo di variabili sono \$_GET[] e \$_COOKIE[]? Qual è il loro ambito di definizione?

Sono delle variabili superglobali. La variabile GET utilizza parametri passati per via get, mentre la variabile COOKIE è un array di cookie

4.

Si commenti la seguente istruzione:

```
setcookie ('test', 'test', time() + 60);
```

Tale metodo serve a settare i cookie

5.

L'istruzione

```
header ("Location: checkcookies.php?set=yes");
```

coinvolge un header: di quale header si tratta? Qual è lo scopo di questa riga?

Lo scopo di questa riga è ricaricare la pagina.

Quali sono gli aspetti che devo guardare per scegliere tra Java e Php per un'applicazione

web lato server? (*)

???

[AJAX] Cap.18

Cos'è e per cosa viene usato l'oggetto XMLHttpRequest?

L'oggetto XMLHttpRequest è utilizzato per permettere lo scambio asincrono dei dati tra client e server.

1. A seguito di un evento generato dall'utente, che ad esempio preme un bottone, viene eseguita una chiamata JavaScript; 2. Viene creato e configurato un oggetto XMLHttpRequest; in particolare viene impostato un parametro di richiesta che include l'identificativo del componente che ha generato l'evento e qualsiasi valore immesso dall'utente; 3. L'oggetto XMLHttpRequest manda al server una richiesta HTTP in modo asincrono rispetto all'azione dell'utente. Il server web elabora la richiesta come farebbe normalmente (ad esempio, attraverso un CGI, o una servlet) e, se necessario, salva i dati passati nella richiesta in modo persistente; 4. Il server web prepara quindi una risposta HTTP nel cui body viene inserito un documento XML con tutti gli aggiornamenti che sono stati richiesti dal client; 5. L'oggetto XMLHttpRequest riceve i dati XML, li elabora e aggiorna opportunamente la DOM della pagina (X)HTML visualizzata dal browser

AJAX si fonda su un certo numero di strumenti, alcuni obbligatori, altri tipicamente utilizzati in questo framework: si spieghi quali sono e che funzione hanno in AJAX, e cos'è AJAX

AJAX (Asynchronous JavaScript and XML) è un approccio utilizzato per l'aggiornamento delle pagine web. Esso comprende obbligatoriamente linguaggi come XHTML, CSS, JavaScript, DOM, XML, XSLT e l'oggetto XMLHttpRequest. Grazie all'utilizzo di queste tecnologie la pagina web può essere aggiornata

in maniera rapida e se è il caso solo in nelle parti in cui necessita di un aggiornamento e non l'intera pagina.

[BOTS, SPIDERS & CRAWLERS] Cap.19

Quali sono le differenze tra bot, spider e crawler?

Tutti e tre sono software che si occupano dell'analisi delle pagine web ma tutti e tre in maniera differente. Il bot cerca delle specifiche cose all'interno della pagina, lo spiders si occupa in genere di rimuovere link obsoleti da enormi pagine, mentre il crawler si occupa d'indicizzare i vari link.

Cosa è e a che serve il file robots.txt?

il file robots.txt è un file ideato per i crawler, esso descrive i limiti che i crawler devono avere durante la loro operazione.

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 = "PIPP0";
8. }
9. maiuscolo();
10. document.write(\$pippo);
11. </script>
12. </p>
13. </body>
14. </html>

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

I linguaggi coinvolti sono: HTML come s'intuisce dalla riga 1 e JavaScript come fa presente la riga 4. Una volta messo in esecuzione il codice, verrà stampato a video il valore della variabile \$pippo che nello specifico è la stringa "pippo" in quanto entrambe le variabili dichiarate sono delle variabili locali quindi pur avendo lo stesso nome (ma scope diverso) assumono due valori diversi.

[SERVIZI WEB & CLOUD COMPUTING] Cap.20

Cosa si intende per servizio web? In quale/i linguaggio/i si implementa?

Con il termine "servizio web" si vuole intendere un insieme di funzionalità che possono esser innescate da software od hardware. I linguaggi utilizzati possono essere anche più di uno in un stesso servizio web, però in genere si preferisce utilizzare l'XML.

Cosa è SOAP e a cosa serve? nei servizi web?

La SOAP (Simple Object Access Protocol) è un protocollo per lo scambio di messaggi, i servizi web utilizzano la SOAP per essere abilitati.

Qual è la differenza tra scoperta manuale e autonoma di un servizio web?

Per scoperta (discovery) si intende la localizzazione di una descrizione elaborabile in modo automatico di un servizio Web che, pur sconosciuto fino a quel momento, risponde a criteri funzionali dati. Per servizio di scoperta (discovery service) si intende un servizio che aiuta nel processo di scoperta; può essere implementato sia dall'agente provider che dall'agente requester o da un altro agente.

Quali sono le differenze tra un web service e CGI?

Cos'è e a cosa serve WSDL?

WSDL permette di descrivere il servizio e rappresenta un accordo che sta alla base dei meccanismi di interazione con quel servizio. Le definizioni dei servizi Web possono poi venir mappate verso un qualsiasi linguaggio di programmazione, piattaforma, modello di oggetti o sistemi di messaggi

Cosa si intende per "Discovery service" nei servizi web?

Per servizio di scoperta (discovery service) si intende un servizio che aiuta nel processo di scoperta; può essere implementato sia dall'agente provider che dall'agente requester o da un altro agente.

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

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

Cos'è un 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.

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.

In CSS si possono utilizzare delle istruzioni come per indicare uno per uno gli elementi di una

qualsiasi lista. Le liste possono essere ordinate () o non ordinate (). Una possibile soluzione è:

```
<body>
```

```
<h1> lista registi </h1>
```

```
<ul>
```

```
    <li> Regista 1 </li>
```

```
    <li>Regista 2 </li>
```

```
</ul>
```

```
<h2> lista film </h2>
```

```
<ul>
```

```
    <li> Film 1 </li>
```

```
    <li>Film 2 </li>
```

```
</ul>
```

```
</body>
```

[Semantic Web] Cap.21

Cosa si intende per "web semantico" (*)

Con il termine "web semantico" s'intende un utilizzo standard del linguaggio in internet così da poter essere comprensibile ai più.

Cosa si intende per Linked data, Open data e Linken open data (*)

Con il termine "open data" s'intendono dati che sono accessibili a tutti in quanto liberi. Mentre per "linked data" s'intende la possibilità di utilizzare i link per viaggiare su diverse sorgenti, ciò fa pensare ad internet come un enorme database accessibile a tutti.

Cosa si intende per RDF e a cosa serve?

RDF (Resource Description Framework) serve per la descrizione delle risorse, lo scopo del RDF è quello di rappresentare l'informazione in modo che sia facile da leggere ed elaborare via calcolatore

Esso è suddiviso in nodi, ogni nodo è detto elemento e contiene delle risorse. il grafo non è assolutamente gerarchico infatti non la posizione degli elementi non è indice di nulla gerarchico. Ecco un esempio di rdf statments (elementi di rdf):


```

<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">

  <rdf:Description
rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
    <cd:artist rdf:resource="http://www.somesite.com/Bob Dylan"/>
    <cd:country>USA</cd:country>
    <cd:company>Columbia</cd:company>
    <cd:price>10.90</cd:price>
    <cd:year>1985</cd:year>
  </rdf:Description>

  <rdf:Description
rdf:about="http://www.recshop.fake/cd/Hide your heart">
    <cd:artist rdf:resource="http://www.someothersite.com/Bonnie Tyler"/>
    <cd:country>UK</cd:country>
    <cd:company>CBS Records</cd:company>
    <cd:price>9.90</cd:price>
    <cd:year>1988</cd:year>
  </rdf:Description>
</rdf:RDF>

```

Essi sono costituiti da soggetti, predicati ed oggetti, esempio:

```

<rdf:Description rdf:about="subject">
  <predicate rdf:resource="object"/>
  <predicate>literal value</predicate>
</rdf:Description>

```

Esempi:

- PROTOCOLLO HTTP:
Accediamo al sito Wikipedia.

Request:

GET <http://www.wikipedia.it> http/1.1

Host: www.wikipedia.it

Connection: Keep Alive

User agent: Chromium

In questo caso, il corpo del messaggio è vuoto in quanto stiamo richiedendo semplicemente la pagina html di wikipedia.

Response:

http/1.1 200 OK

Connection: close

Content-Encoding: gzip

Content-Length: 15999

Content-Type: text/html; charset=UTF-8

Date: Mon, 21 Dec 2020 09:29:23 GMT

- PROTOCOLLO HTTP CON MECCANISMO DI CACHE

Request:

GET <http://www.wikipedia.it> http/1.1

Host: www.wikipedia.it

Connection: Keep Alive

User agent: Chromium

If-Modified-Since: Mon, 21 Dec 2020 09:29:23 GMT

Response:

http/1.1 304 Not Modified

Connection: close

Content-Encoding: gzip

Content-Length: 15999

Content-Type: text/html; charset=UTF-8

Date: Mon, 21 Dec 2020 09:29:23 GMT

Possibili domande:

Cos'è http?

È un protocollo di tipo client-server che serve a permettere la comunicazione tra essi. Http prevede due tipi di messaggi: Richiesta dal client al server e Risposta dal server al client. La richiesta è formata da una prima riga Request Line, un certo numero di righe di intestazione, una riga vuota e il corpo del messaggio; per la risposta invece, si ha una prima riga chiamata StatusLine e il corpo del messaggio che conterrà la pagina richiesta. Inoltre, questo protocollo è di tipo stateless, ovvero non prevede il mantenimento delle informazioni sullo stato di comunicazione tra client e server; infatti verranno adottati diversi meccanismi per la risoluzione del problema.

Tipi di messaggi http:

Si hanno due tipi: Richiesta e Risposta

La richiesta è formata da: una prima riga detta RequestLine e ha tre campi (Metodo:GET,POST,HEAD ecc..., URL e Versione Http); un certo numero di righe di intestazione (Host, Connection, UserAgent); una riga vuota; il body del messaggio.

La risposta è formata da: una prima riga detta StatusLine e ha tre campi (Versione http, Codice di stato: 200 OK,301 MOVED PERMANENTLY,404 NOT FOUND,505 HTTP VERSION NOT SUPPORTED) e Messaggio di stato; numero di righe di intestazione ed infine il body del messaggio.

Come adotta il meccanismo di cache http?

Occorre che il client sia in grado di controllare se la pagina è stata modificata; per farlo http prevede il campo If-Modified-Since inserito nell'header della richiesta e il codice 304Not Modified nella status line.

-HTML

Semplice pagina di benvenuto:

```
<!DOCTYPE html>
<html>
  <head><title>Pagina di benvenuto</title></head>
  <body>
    <h1>BENVENUTO UTENTE</h1>
  </body>
</html>
```

Ecco come appare:



-FORM IN UN DOCUMENTO HTML

Supponiamo che l'utente invii il proprio nome ad una servlet `servletDati` (la servlet è stata impostata in modo tale da ricevere l'argomento inserito dall'utente).

```
<!DOCTYPE html>
<html>
  <head><title>Inserire dati</title></head>
  <body>
    <form action="servletDati" method="POST">
      NomeUtente: <input type="text" name="user">
      <input type="submit" value="Invio">
    </form>
  </body>
</html>
```

Ecco come appare:



Possibili domande:

Cos'è html e da cosa è formato?

Html (HyperText Markup Language) è un linguaggio di markup per ipertesti. Esso è utilizzato per la rappresentazione dei documenti. Il documento infatti è formato da un insieme di elementi ed un insieme di etichette. L'elemento è quella parte di documento compreso fra etichette di apertura e chiusura.

Cos'è xml?

Xml (extensible markup language) è un linguaggio di markup come html, ma con la caratteristica di essere estensibile, nel senso che le etichette non sono prefissate ma scelte dall'utente.

Cos'è html5?

È un'evoluzione dell'html. In questa nuova evoluzione sono stati seguiti alcuni principi come: riduzione plugin esterni, gestione degli errori, indipendenza dal device, trasparente al pubblico ed infine se non quello più importante la possibilità di memorizzare i dati localmente (sessionStorage o localStorage).

Java

Possibili domande:

Cos'è Java?

Java, è un linguaggio di programmazione orientato agli oggetti. Si tratta di un linguaggio fortemente tipato, non ammette costrutti insicuri e viene eseguito su una JVM. Inoltre, ha una gestione automatica della memoria (Garbage Collection) che evita l'introduzione di errori legati alla deallocazione della memoria.

Cosa rende sicuro Java?

È l'assenza dell'aritmetica dei puntatori (questo non significa vietare altri tipi di puntatori, come i references). Oltre all'assenza dei puntatori, java definisce chiaramente il comportamento di variabili non inizializzate come ad esempio: variabili sullo heap vengono inizializzate automaticamente, ne consegue che variabili di classi e istanze non possono avere valori non definiti; variabili sullo stack non vengono automaticamente inizializzate, quindi tale operazione deve avvenire prima dell'uso.

Cos'è un'applet e quali sono le sue caratteristiche?

Un'applet è uno strumento lato client basato sul linguaggio Java. Le caratteristiche sono quelle di essere portabili (lo sviluppatore non sa nulla della piattaforma su cui viene eseguita l'applet) e sicure.

Ciclo di vita di un'applet?

Init() chiamato automaticamente quando l'applet è in esecuzione; start() chiamato dopo init() e può essere richiamato più volte; stop() chiamato ogni volta che l'utente esce dalla pagina su cui viene eseguita l'applet; destroy() chiamato quando il browser viene chiuso.

CGI:

Mostriamo un primo esempio di CGI. Supponendo di aver già impostato la configurazione di Apache2, creiamo uno script Bash:

firstCGI.cgi:

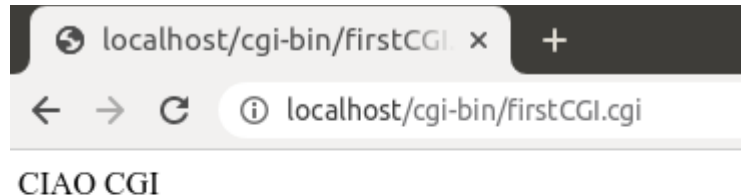
```
#!/bin/bash

echo "Content-type: text/html"
echo ""
echo "CIAO CGI"
exit 0
```

Risultato:



http://localhost/cgi-bin/firstCGI.cgi



Vediamo adesso come funziona il passaggio dei parametri:

In questo esempio, creo uno script cgi che leggerà una o più parole inserite in un form.

ProgrammaCGI.cgi:



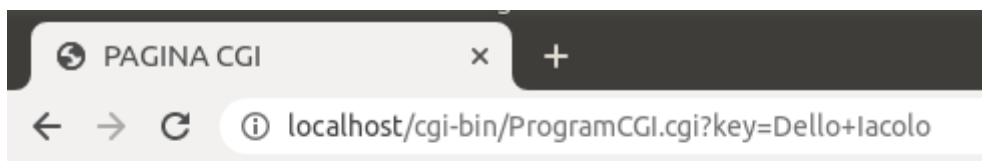
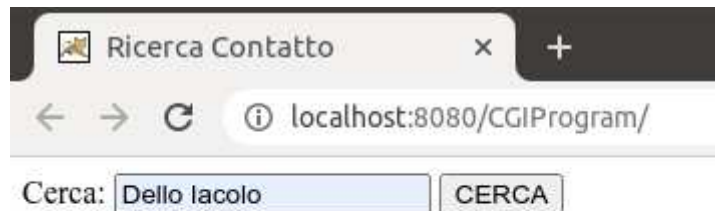
```
#!/bin/bash
echo "Content-type: text/html"
echo ""
name=$(QUERY_STRING)
nome=$(echo $name | sed 's/.*key=//' | sed 's/\+/ /g')
echo "<html>"
echo "<head>"
echo "<title>PAGINA CGI</title>"
echo "</head>"
echo "<body><h1>Hai cercato la parola: $nome </h1></body></html>"
```

NB: ho inserito le operazioni sed perché se nella form inseriamo uno spazio tra una parola e l'altra, la variabile d'ambiente QUERY_STRING darà come risultato: key=parola1+parola2+....+parolaN.

Index.html:

```
<html>
  <head>
    <title>Ricerca Contatto</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form action="http://localhost/cgi-bin/ProgramCGI.cgi" method="GET">
      Cerca: <input type="text" name="key">
      <input type="submit" value="CERCA">
    </form>
  </body>
</html>
```

Risultato:



Hai cercato la parola: Dello Iacolo

Come ultimo esempio, svolgo l'esercizio presente sulle slide della prof Corazza in cui tramite un form inserisco un numero di telefono e lo script avviato ricerca quest'ultimo all'interno di un file .txt:

index.html:

```
<html>
  <head>
    <title>Ricerca Contatto</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form action="http://localhost/cgi-bin/ProgramCGI.cgi" method="GET">
      | Cerca: <input type="text" name="key">
      <input type="submit" value="CERCA">
    </form>
  </body>
</html>
```

rubrica.txt:

Nome	Telefono
Umberto	66633322
Carmine	88877766
Marco	66644112
Vincenzo	44455599
Mario	11223333
Antonio	66622211

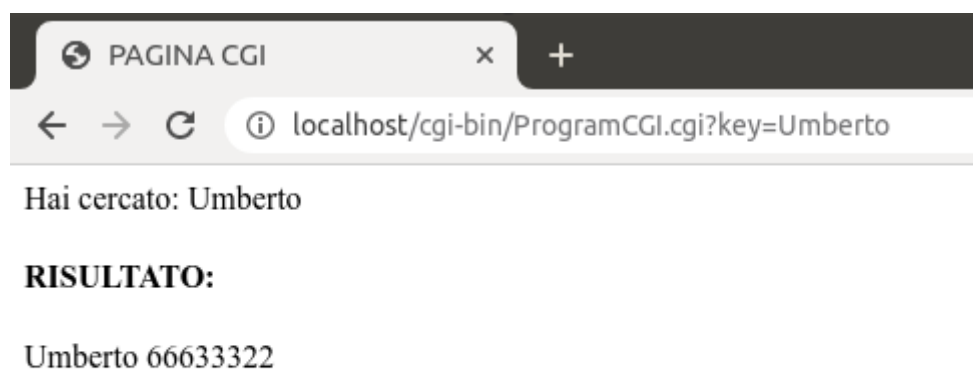
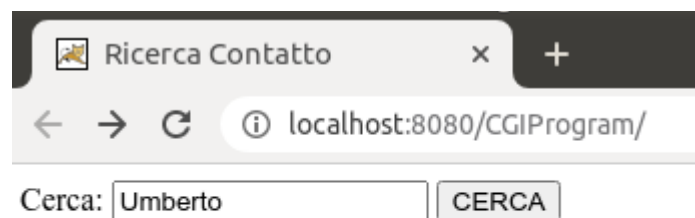
ProgrammaCGI.cgi:

```
#!/bin/bash

echo "Content-type: text/html"
echo ""
name=$(QUERY_STRING)
nome=$(echo $name | sed 's/.*key=//' | sed 's/\+/ /g')
result=$(cat rubrica.txt | grep $nome)

echo "<html>"
echo "<head>"
echo "<title>PAGINA CGI</title>"
echo "</head>"
echo "<body><p>Hai cercato: $nome </p>"
echo "<h4>RISULTATO:</h4>"
echo "<p>$result</p></body></html>"
```

Risultato:



Servlet:

Scriviamo una prima servlet che stampa HELLO WORLD al click di un utente da una pagina html.

NB: la servlet è stata scritta col programma NetBeans e con supporto di ApacheTomcat.

Dopo aver creato il progetto e la servlet, la pagina index.html sarà di questo tipo:

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
<head>
<title>Pagina</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<h1>CLICCA SUL PULSANTE</h1>
<form action = "helloworld" method="POST">
<input type="submit" value="Clicca qui">
</form>
</body>
</html>
```

mentre la nostra servlet:

```

public class HelloWorld extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        PrintWriter out = response.getWriter();

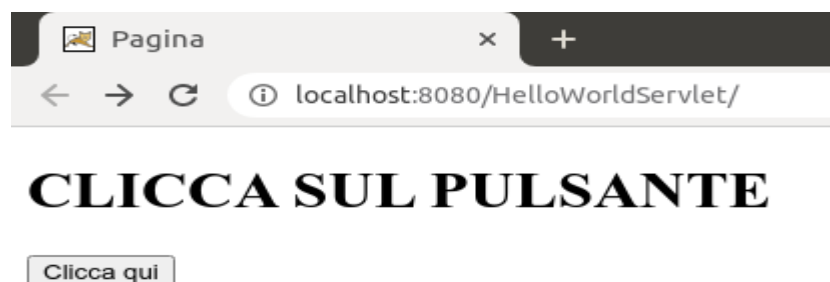
        out.println("<html>"
            + "<head><title>Pagina di benvenuto</title></head>"
            + "<body>"
            + "<h1>HELLO WORLD</h1>"
            + "</body></html>");
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}

```

Il risultato sarà così:



Un altro esempio che possiamo fare è vedere come una servlet prende i parametri in ingresso; possiamo fare riferimento all'esempio che abbiamo fatto all'inizio sulla form:

index.html

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <title>Inserire Dati</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form action="servletDati" method="POST">
      NomeUtente: <input type="text" name="username">
      <input type="submit" value="Invio">
    </form>
  </body>
</html>
```

servletDati.java

```
public class ServletDati extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        String user = request.getParameter("username");

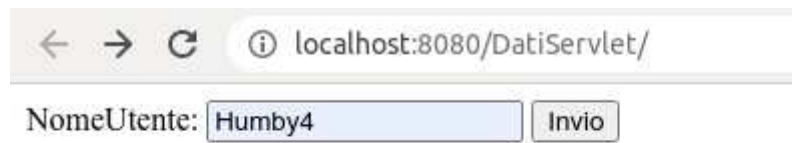
        out.println("BENVENUTO " + user);
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

    }

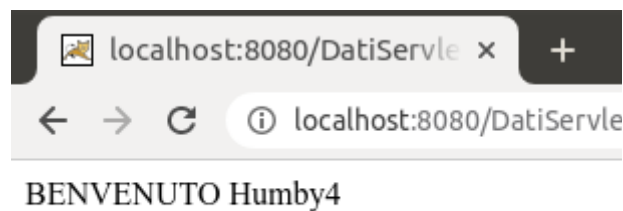
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}
```

risultato:



← → ↻ ⓘ localhost:8080/DatiServlet/

NomeUtente:



Vediamo adesso come una servlet accede ad una base di dati.
NB: Ho impostato il database MySql e l'ho connesso a Netbeans.

Index.html:

```
to change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <title>Cerca nel database</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form action="search" method="POST">
      <p>CERCA FAMIGLIA: <input type="text" name="key"></p>
      <p><input type="submit" value="Invia"></p>
    </form>
  </body>
</html>
```

search.java

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException, SQLException, ClassNotFoundException, InstantiationException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    String key = request.getParameter("key");

    Class.forName("com.mysql.jdbc.Driver");
    String url = "jdbc:mysql://localhost:3306/Numero10";
    Connection con = DriverManager.getConnection(url, "root", "password");

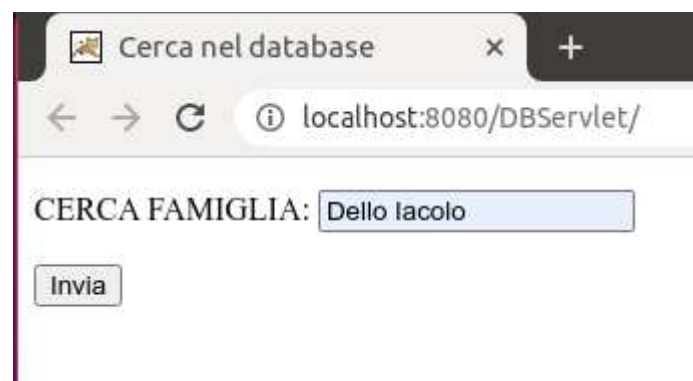
    String query = "SELECT * FROM Famiglia WHERE Cognome='"+key+"'";

    PreparedStatement pst = con.prepareStatement(query);
    ResultSet st = pst.executeQuery();

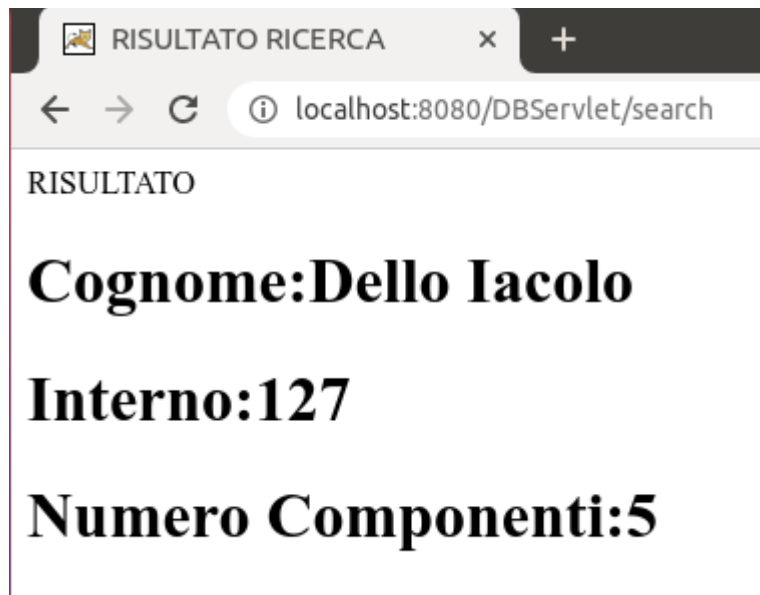
    while(st.next()) {
        String cog = st.getString("Cognome");
        int id = st.getInt("id");
        int ncom = st.getInt("Numero_Componenti");

        out.println("<html><head><title>RISULTATO RICERCA</title></head>"
            + "<body>"
            + "<h1>Cognome:"+cog+"</h1>"
            + "<h1>Interno:"+id+"</h1>"
            + "<h1>Numero Componenti:"+ncom+"</h1>"
            + "</body></html>");
    }
}
```

Risultato:



The screenshot shows a web browser window with a single tab titled 'Cerca nel database'. The address bar displays 'localhost:8080/DBServlet/'. The main content area shows the text 'CERCA FAMIGLIA:' followed by a text input field containing 'Dello Iacolo'. Below the input field is a button labeled 'Invia'.



Domande possibili:

Cos'è una servlet?

Sono un esempio dell'uso di Java server-side per la programmazione Web. Sono principalmente classi Java caricate dinamicamente per estendere la funzionalità del server. Esse operano solo sul server e non richiedono condizioni speciali da parte del browser.

Le servlet definiscono due classi astratte che implementano l'interfaccia servlet: GenericServlet e HttpServlet. Ogni sottoclasse di HttpServlet deve fare Override di almeno un metodo come: doGet, doPost, doPut, doDelete e getServletInfo.

Descrivere i metodi più importanti di una servlet.

L'interfaccia di una servlet dichiara 5 metodi come init() che inizializza la servlet una volta sola durante il ciclo di esecuzione; getServletConfig() che ritorna un oggetto che implementa l'interfaccia ServletConfig e fornisce l'accesso alla configurazione della servlet; getServletInfo() che restituisce una stringa di informazioni; service(request, response) che viene mandato in esecuzione in base alla richiesta di un client; destroy() quando la servlet viene terminata.

Differenza tra servlet e CGI.

Rispetto alle CGI, le servlet sono molto più manutenibili, inoltre permettono di condividere informazioni sia all'interno di un'applicazione che col server. Nelle CGI, l'unico ambito di definizione è quello di richiesta, ad ogni richiesta corrisponde un processo che termina non appena la richiesta viene soddisfatta. Nella servlet invece ogni richiesta corrisponde ad un thread all'interno di una servlet engine.

CMS e WEB FRAMEWORKS

Domande possibili:

cos'è un CMS?

Un CMS è un sistema per la gestione dei contenuti. Esso può essere un database relazionale per immagazzinare i suoi contenuti, questi ultimi vengono memorizzati in XML. Inoltre, CMS mette a disposizione di chi crea il documento, uno o più template. Un'ulteriore funzionalità è quella del retrieval dei documenti; i documenti immessi nel CMS, sono memorizzati in una repository opportunamente indicizzata per effettuare ricerche semplificate.

Cos'è un WebFramework?

È un framework per lo sviluppo di applicazioni web mettendo a disposizione sia strumenti lato server che lato client.

Descrivere il modello MVC per i web frameworks.

Il modello MVC per web è un po' diverso da quello standard in quanto non vi è alcuna relazione diretta tra Model e View a causa del fatto che la comunicazione tra essi è veicolata da richieste http. Il model rende fruibile l'accesso e la gestione dei dati; lo scambio dei messaggi tra model e controller è veicolata da un ORM responsabile del mapping tra modello dei dati e modello ad oggetti. La View è responsabile della visualizzazione dei dati, tipicamente in pagine HTML. Il controller gestisce la logica del programma, esso si occupa di ricevere le richieste http e gestire la comunicazione col modello dei dati ed infine trasferirli alla vista.

Cos'è django?

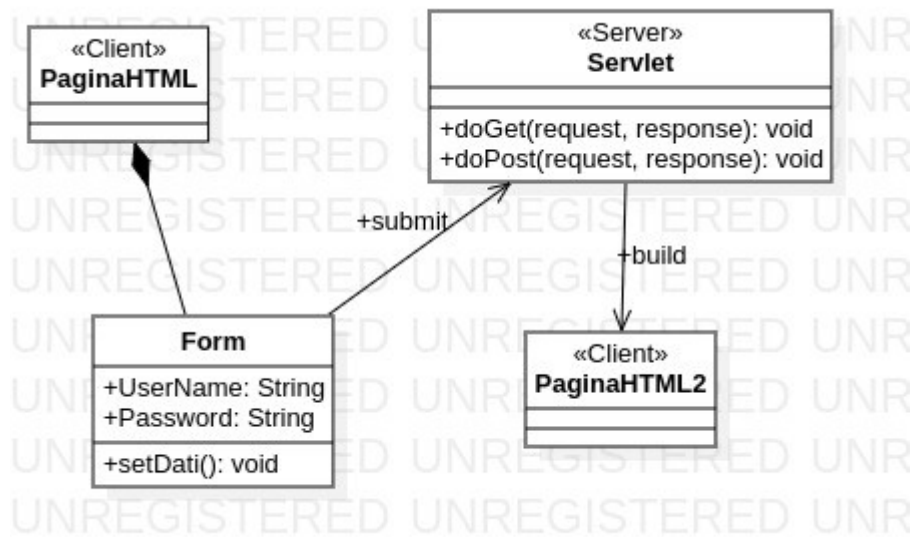
È un web framework basato su python, offre supporto per una serie di necessità come: gestione utenti, controllo accessi, forms ecc...

django è basato sul modello MVT in cui il controller è la view mentre la view è il Template. Per la parte di modello, django propone l'uso di ORM in grado di specificare i tipi dei campi da memorizzare; in questo modo è possibile applicare una validazione dell'informazione introdotta a livello di campo. Un altro aspetto molto interessante è che django realizza il routing, ovvero mette in corrispondenza l'url col codice che va messo in esecuzione per produrre la risposta http.

WEBUML

Esempio:

Creare tramite Conallen il caso in cui un form invia dei dati di accesso.



Domande possibili:

Cosa propone l'estensione di Conallen per il web?

Quest'estensione propone due stereotipi: Pagina server e pagina client. La pagina server contiene metodi e variabili relativi allo scripting server side mentre la pagina client contiene elementi relativi alla formattazione, in pratica allo scripting client side.

JSP e STRUTS2

Esempio:

Realizziamo una semplice JSP che stampa HELLO WORLD!

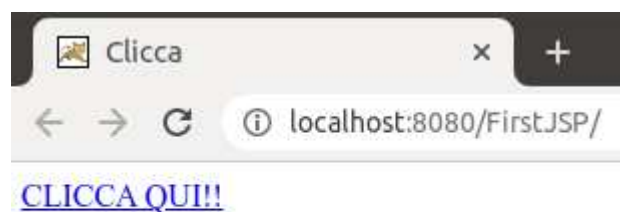
Index.html

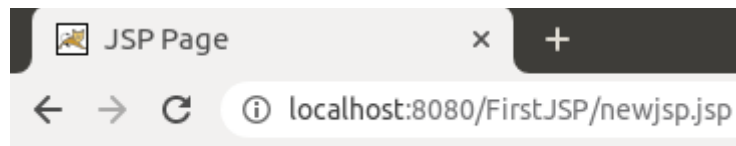
```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <title>Clicca</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <a href="newjsp.jsp">CLICCA QUI!!</a>
  </body>
</html>
```

newjsp.jsp

```
<!--
Document   : newjsp
Created on : 14-dic-2020, 14.06.51
Author    : umberto
-->
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

Risultato:





Hello World!

-Vediamo adesso come funziona una jsp riguardo al passaggio dei parametri tramite un form:

index.html

```
<html>
  <head>
    <title>Effettua Login</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form action="newjsp.jsp" method="post">
      User: <input type="text" name="user">
      Pass: <input type="text" name="pass">
      <input type="submit" value="Invio">
    </form>
  </body>
</html>
```

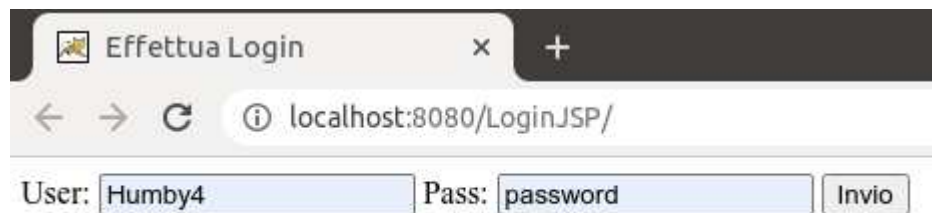
newjsp.jsp:

```
<%--
    Document    : newjsp
    Created on  : 14-dic-2020, 14.21.26
    Author      : umberto
--%>

<%@page import="java.lang.String"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%! String username,password; %>

<%
    username = request.getParameter("user");
    password = request.getParameter("pass");
    if (username.equals("humby4") && password.equals("password")) {
        out.println("BENVENUTO "+username);
    }
%>
```

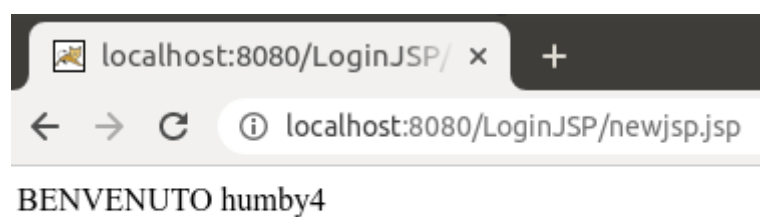
Risultato:



Effettua Login

localhost:8080/LoginJSP/

User: Pass:



localhost:8080/LoginJSP/

localhost:8080/LoginJSP/newjsp.jsp

BENVENUTO humby4

-Inoltre, ecco un altro esempio sull'utilizzo di JavaBean:

index.html:

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <title>Settaggio dati</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <h4>INSERISCI I DATI</h4>
    <form action="newjsp.jsp" method="post">
      Nome: <input type="text" name="name">
      Cognome: <input type="text" name="surname">
      Et : <input type="text" name="age">
      <input type="submit" value="INVIA DATI">
    </form>
  </body>
</html>
```

newjsp.jsp:

```
<%--
Document   : newjsp
Created on : 14-dic-2020, 15.38.54
Author    : umberto
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>PAGINA JSP</title>
  </head>
  <body>
    <jsp:useBean id="student" class="com.JSPBeans.StudentB">
      <jsp:setProperty name="student" property="name" value = '<%=request.getParameter("name") %>' />
      <jsp:setProperty name="student" property="surname" value = '<%=request.getParameter("surname") %>' />
      <jsp:setProperty name="student" property="age" value = '<%=request.getParameter("age") %>' />
    </jsp:useBean>
    <p>
      First Name Student: <jsp:getProperty name="student" property="name"/>
      Second Name Student: <jsp:getProperty name="student" property="surname"/>
      Age: <jsp:getProperty name="student" property="age"/>
    </p>
  </body>
</html>
```

StudentB.java

```
package com.JSPBeans;

public class StudentB {
    private String name;
    private String age;
    private String surname;

    public StudentB () {
        name=surname=age=null;
    }
    public String getName() {
        return name;
    }
    public String getSurname() {
        return surname;
    }
    public String getAge() {
        return age;
    }
    public void setName(String n) {
        this.name=n;
    }
    public void setSurname(String s) {
        this.surname=s;
    }
    public void setAge(String a) {
        this.age=a;
    }
}
```

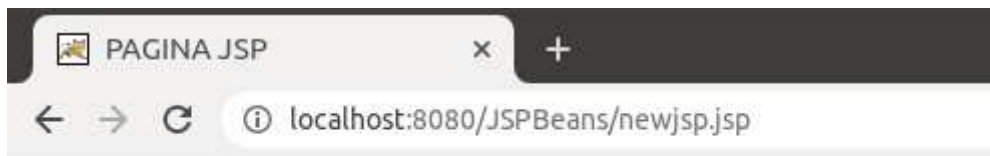
NB: c'è un errore; StudentB deve implementare java.io.Serializable

Risultato:



INSERISCI I DATI

Nome: Cognome: Et :



First Name Student: Umberto Second Name Student: Dello Iacolo Age: 23

Vediamo adesso come scrivere un'applicazione web utilizzando il framework Struts2: scriveremo un programma di login.

index.html:

```
<html>
  <head>
    <title>Login</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form action="login" method="post">
      Username: <input type="text" name="username">
      Password: <input type="text" name="password">
      <input type="submit" value="Login">
    </form>
  </body>
</html>
```


struts.xml:

```
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
  <!-- Configuration for the default package. -->
  <package name="default" extends="struts-default">
    <action name="login" class="LoginAction">
      <result name="success">
        success.jsp
      </result>
      <result name="no">
        error.jsp
      </result>
    </action>
  </package>
</struts>
```

LoginAction.java


```

import com.opensymphony.xwork2.ActionSupport;

public class LoginAction extends ActionSupport {
    private String username,password;

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String execute() {
        if(username.equals("Humby4") && password.equals("password")) {
            return "success";
        }
        else {
            return "no";
        }
    }
}

```

error.jsp:

```

<%--
Document      : error
Created on    : 10-gen-2021, 12.04.37
Author       : umberto
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri = "/struts-tags" prefix="s"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Error</title>
    </head>
    <body>
        <h1>Le credenziali non corrispondono!!</h1>
    </body>
</html>

```

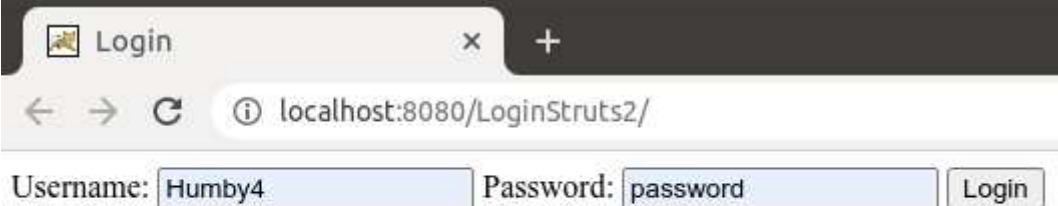
success.jsp:

```
<%--
Document    : success
Created on  : 10-gen-2021, 12.04.26
Author      : umberto
--%>

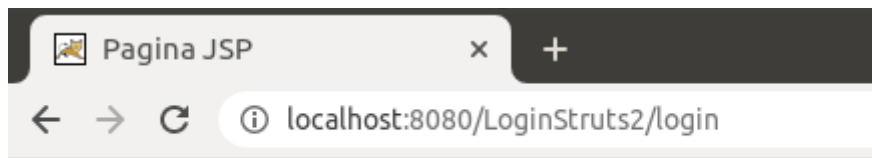
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri = "/struts-tags" prefix="s"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Pagina JSP</title>
  </head>
  <body>
    <h1>Login eseguito correttamente</h1>

  </body>
</html>
```

Risultato:



The screenshot shows a web browser window with a single tab titled "Login". The address bar displays "localhost:8080/LoginStruts2/". The page content includes a form with two input fields: "Username:" containing the text "Humby4" and "Password:" containing the text "password". To the right of these fields is a button labeled "Login".



Login eseguito correttamente

Domande Possibili:

Differenza tra servlet e JSP.

Il principale problema delle servlet è che la preparazione di una pagina da mandare al client richiede istruzioni per la presentazione risultano immerse nelle istruzioni di stampa; per le JSP invece, template statico e codice Java convivono ma in modo chiaro e semplice. Inoltre per la servlet, bisognava compilarla ad ogni modifica, mentre questo non vale per la jsp nel caso si modifichi il template.

Descrivere gli elementi di una JSP.

Una pagina jsp possiede i seguenti elementi: Direttive che servono a specificare le impostazioni della pagina, includere altri contenuti e specificare delle librerie; Scriptlet in cui vengono eseguite le istruzioni durante l'elaborazione della richiesta http; Dichiarazioni per la definizione di variabili e metodi ed infine le Espressioni che contengono espressioni java che il contenitore converte sempre in un oggetto String.

Quali sono i principali problemi di una JSP?

I principali problemi di una jsp sono: non favoriscono il riuso del codice; presentano problemi di leggibilità che manutenzione; si allontana dalla programmazione ad oggetti in quanto non si può usare l'ereditarietà e rende difficile sia lo sviluppo del codice che il debugging.

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.

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.

Session Tracking

Esempi:

Realizziamo delle servlet che si occupano di salvare le credenziali di un utente tramite l'utilizzo dei cookies.

Index.html

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <title>Effettua il login</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form action="login" method="post">
      Utente: <input type="text" name="user">
      Pass: <input type="text" name="pass">
      <input type="submit" value="INVIO">
    </form>
  </body>
</html>
```

Login.java

```

public class Login extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String id = request.getParameter("user");
        String pas = request.getParameter("pass");
        if (id.equals("Humby4") && pas.equals("password")) {
            out.println("BENVENUTO " + id);
            out.println("HAI LA PASSWORD: " + pas);
            Cookie c1 = new Cookie("Username", id);
            Cookie c2 = new Cookie("Password", pas);
            response.addCookie(c1);
            response.addCookie(c2);
            out.println("<a href= 'profilo' >Visualizza il profilo</a>");
        }
        out.close();
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}

```

Profilo.java

```

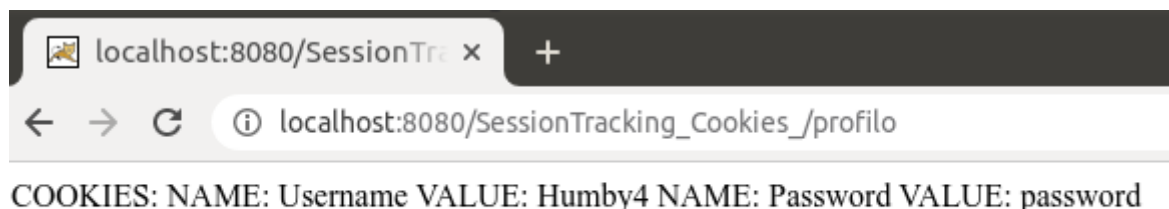
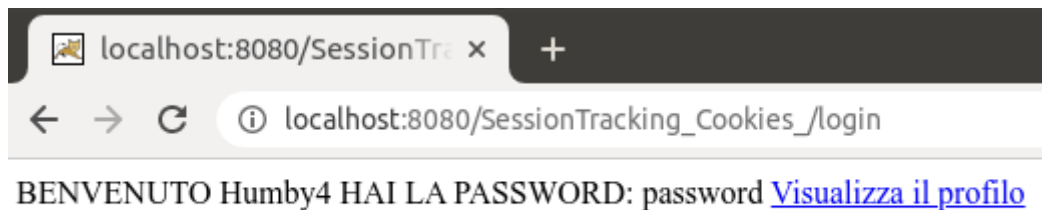
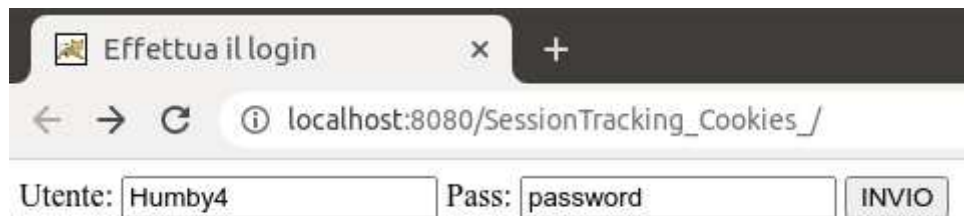
public class Profilo extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        Cookie []c = request.getCookies();
        out.println("COOKIES: ");
        if (c != null) {
            for (int i = 0; i < c.length; i++) {
                String name = c[i].getName();
                String value = c[i].getValue();
                out.println("NAME: " + name);
                out.println("VALUE: " + value);
            }
        }
        out.close();
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}

```

Risultato



-Session Tracking tramite l'utilizzo delle variabili di sessione:

index.html:

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <title>LOGIN</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form action="login" method="post">
      User: <input type="text" name="user">
      Password: <input type="text" name="pass">
      <input type="submit" value="INVIO">
    </form>
  </body>
</html>
```

login.java


```

public class Login extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String id = request.getParameter("user");
        String pass = request.getParameter("pass");
        if (id.equals("humby4") && pass.equals("password")) {
            out.println("BENVENUTO " + id);
            HttpSession session = request.getSession();
            session.setAttribute("Id", id);
            session.setAttribute("Pass", pass);
            out.println("<a href='profilo'>CLICCA QUI</a>");
        }
        else {
            out.println("ERRORE! <a href='index.html'>Ritenta</a>");
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}

```

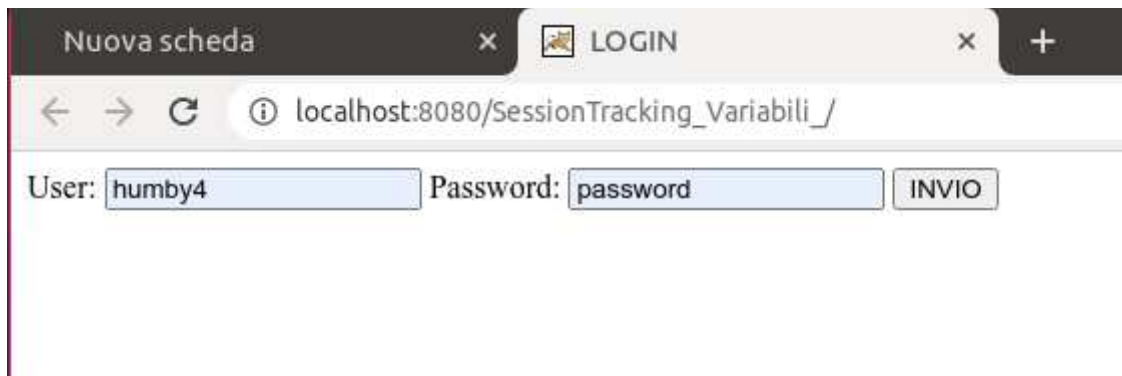
profilo.java

```

public class Profilo extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        HttpSession session = request.getSession();
        Enumeration valueNames;
        if (session != null) {
            valueNames = session.getAttributeNames();
        }
        else {
            valueNames = null;
        }
        if (valueNames != null) {
            out.println("BENVENUTO, I DATI INSERITI PRIMA:\n");
            String name, value;
            while (valueNames.hasMoreElements()) {
                name = valueNames.nextElement().toString();
                value = session.getAttribute(name).toString();
                out.println(name + ": " + value);
                out.println("\n");
            }
        }
        else {
            out.println("SESSIONE SCADUTA!\n");
        }
        out.close();
    }
}

```

Risultato:



Domande possibili:

Descrivere la differenza tra session tracking per autenticazione e campi nascosti.

La session tracking per autenticazione si basa sulla memorizzazione di username e password sul client in modo che l'utente non debba digitarli tutte le volte che viene inoltrata una richiesta http. Inoltre è facile da implementare, funziona anche se l'utente accede al sito da macchine diverse e continua a funzionare anche se l'utente esce dal sito o addirittura dal browser. Per quanto riguarda i campi nascosti nelle form, risulta utile nel caso in cui l'informazione da passare è di grossa dimensione, salvandola direttamente sul server e passare solo l'identificativo.

Cos'è un cookies? Come funziona?

È un pezzo di codice che viene scambiato tra client e server. Esso contiene alcune informazioni: host e path dell'applicazione, expiration date, nome e valore del cookie. Ogni volta che lo user agent prepara una richiesta http identificata da host e path, inserisce nell'header tutti i cookies che ha memorizzato e che corrispondono a quella combinazione. A questo punto il server riceve il cookie e lo inserisce nel campo set-cookie. Inoltre se vuole che il cookie venga cancellato, ne invia un altro con lo stesso nome in modo tale da sovrascriverlo e poi cancellarlo.

CSS

Prima pagina integrando CSS:

```
<!DOCTYPE html>
<html>
  <head><title>PAGINA CSS</title>
    <style type="text/CSS">
      body {
        background-color: green;
      }
    </style>
  </head>
  <body>
    <h1>HELLO WORLD!</h1>
  </body>
</html>
```

Risultato:



Domande Possibili:

cos'è CSS?

È un linguaggio dichiarativo che serve per associare ad un documento HTML o XML uno stile di presentazione tramite dei fogli di stile. Il CSS permette di formulare delle regole per associare ad ogni elemento del documento in ingresso; le regole sono divise in due parti: un selettore e una dichiarazione. La dichiarazione a sua volta si divide in proprietà e valore.

JAVASCRIPT

Somma di due numeri

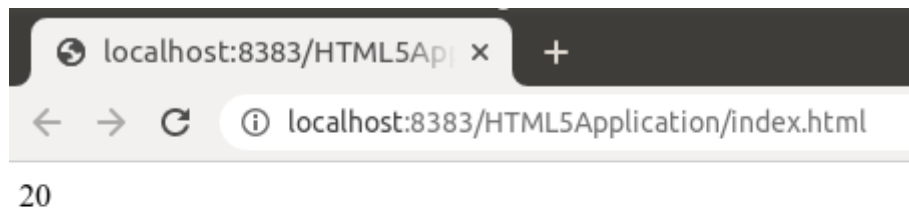
index.html

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <title>Pagina di inserimento</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script type="text/javascript" src="somma.js"></script>
  </head>
  <body>
    <form method="post" name="formDati">
      Input 1: <input type="text" name="in1">
      Input 2: <input type="text" name="in2">
      <input type="button" onClick="sum()" value="Premi">
    </form>
  </body>
</html>
```

somma.js

```
function sum() {
    var a=parseInt(document.formDati.in1.value);
    var b=parseInt(document.formDati.in2.value);
    var c = a + b;
    document.writeln(c);
}
```

Risultato:



Vediamo adesso come html5 insieme a JavaScript usano Session e Local Storage.
Partendo dall'esempio di prima, aggiungiamo metodi relativi al session e al local storage:

index.html:

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <title>Pagina di inserimento</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script type="text/javascript" src="somma.js"></script>
  </head>
  <body>
    <form method="post" name="formDati">
      Input 1: <input type="text" name="in1">
      Input 2: <input type="text" name="in2">
      <input type="button" onClick="sum()" value="Premi">
      <input type="button" onClick="clearLocalS()" value="Clear Storage">
    </form>
  </body>
</html>
```

somma.js

```
function sStorage() {
  if (sessionStorage.hits) {
    sessionStorage.hits=Number(sessionStorage.hits)+1;
  }
  else {
    sessionStorage.hits=1;
  }
  document.writeln("TOTAL HITS (Session Storage): " + sessionStorage.hits);
}
function lStorage() {
  if(localStorage.getItem("hits")==null) {
    localStorage.hits=1;
  }
  else if (localStorage.hits) {
    localStorage.hits=Number(localStorage.hits)+1;
  }
  else {
    localStorage.hits=1;
  }
  document.writeln("TOTAL HITS (Local Storage): " + localStorage.hits);
}

function clearLocalS() {
  localStorage.clear();
}
```

Risultato:

A screenshot of a web browser window. The address bar shows 'localhost:8383/HTML5Application/index.html'. The page has two input fields: 'Input 1:' with the value '5' and 'Input 2:' with the value '56'. To the right of the inputs are two buttons: 'Premi' and 'Clear Storage'.

localhost:8383/HTML5Ap x +
localhost:8383/HTML5Application/index.html
61 TOTAL HITS (Session Storage): 1 TOTAL HITS (Local Storage): 1

NB: Chiudendo il browser, il session storage si resetta mentre il local storage rimane (infatti serve una procedura per resettare quest'ultimo).

A screenshot of a web browser window. The address bar shows 'localhost:8383/HTML5Application/index.html'. The page has two input fields: 'Input 1:' with the value '500' and 'Input 2:' with the value '66'. To the right of the inputs are two buttons: 'Premi' and 'Clear Storage'.

localhost:8383/HTML5Ap x +
localhost:8383/HTML5Application/index.html
566 TOTAL HITS (Session Storage): 1 TOTAL HITS (Local Storage): 2

NB: Clear Storage resetta il numero di Hits associati al local storage!

Vediamo adesso qualche esempio sull'utilizzo della libreria jQuery: aggiungo la funzione printHello che conterrà istruzioni jquery.

Index.html:

```
<html>
  <head>
    <title>Pagina di inserimento</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script type="text/javascript" src="jquery-3.5.1.js"></script>
    <script type="text/javascript" src="somma.js"></script>
  </head>
  <body>
    <form method="post" name="formDati">
      Input 1: <input type="text" name="in1">
      Input 2: <input type="text" name="in2">
      <input type="button" onClick="sum()" value="Premi">
      <input type="button" onClick="clearLocalS()" value="Clear Storage">
      <input type="button" onClick="printHello()" value="Stampa ciao mondo">
    </form>
  </body>
</html>
```

somma.js

```
function printHello() {
  $(document).ready(function(){
    alert("CIAO MONDO");
  });
}
```

Risultato:



Domande possibili:

Cos'è JavaScript?

JavaScript è un linguaggio di scripting client e server side basato su oggetti. Per linguaggio di scripting si intende un linguaggio usato per manipolare le caratteristiche di un sistema esistente. Inoltre le istruzioni js possono stare in ogni parte del documento e non c'è nessun limite al numero di script di una pagina; infine, è un linguaggio interpretato e debolmente tipato.

Differenza tra Java e JavaScript.

JavaScript è completamente distinto da Java in quanto: java è usato per scopi diversi, è un linguaggio compilato, è basato su classi, gerarchia sulle classi e controllo dei tipi delle variabili forte. Per javascript invece: è utilizzato solo per il web, non è compilato, gerarchie delle istanze, basato su prototipi e infine tutti gli oggetti sono istanze.

Come sono gestite le variabili in JavaScript?

In JS, le variabili si dividono in Globali e Locali. Le globali vengono tipicamente inizializzate dalle dichiarazioni var, eseguite quando il documento viene caricato e distrutto quando il documento viene eliminato; mentre quelle locali vengono inizializzate all'interno di una funzione e restano attive solo al termine di essa.

Spiegare l'ereditarietà di JavaScript.

L'ereditarietà è basata su prototipi e non sulle classi. In JS, attributi e metodi vengono ereditati solo dal costruttore. Quando viene creato un oggetto, viene associato ad esso un riferimento al prototipo del suo costruttore attraverso l'espressione `<costruttore>.prototipo`. A sua volta, un prototipo può avere un riferimento implicito diverso da null al suo prototipo, creando così una catena di prototipi che viene risalita per disambiguare i riferimenti alle proprietà.

Che differenza c'è tra Session Storage e Local Storage?

Il Session Storage ha uno scope limitato alla singola finestra, inoltre se la finestra viene chiusa, la sessione termina ed i dati memorizzati vengono persi. Invece per Local Storage, non è limitato alla singola finestra e salva la sessione in modo persistente. Essa risulta efficace nel caso in cui bisogna salvare grosse quantità di dati. Un problema è che i dati non vengono automaticamente rimossi, per far ciò bisogna ripulirlo tramite `localStorage.removeItem(key)` oppure `localStorage.clear()`.

Cos'è la libreria JQuery?

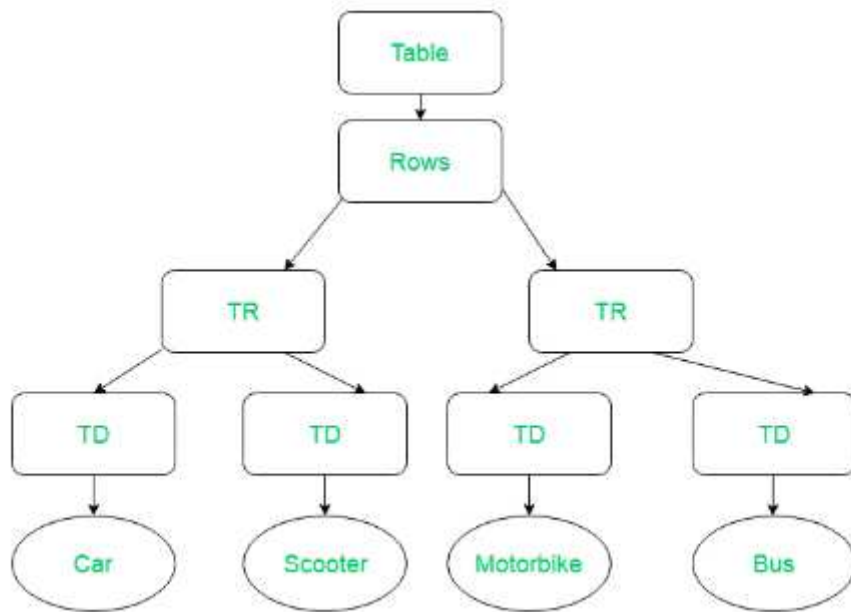
È una libreria di JavaScript che facilita l'uso della Document Object Model per muoversi all'interno di documenti HTML, gestione degli eventi e interazione con AJAX. Le istruzioni JQuery sono segnalate dalla presenza del carattere `$` all'inizio.

DOM:


```

<Table>
  <ROWS>
    <TR>
      <TD>Car</TD>
      <TD>Scooter</TD>
    </TR>
    <TR>
      <TD>MotorBike</TD>
      <TD>Bus</TD>
    </TR>
  </ROWS>
</Table>

```



Vediamo adesso come funziona il parserXML usando la libreria jquery ed una richiesta ajax: in questo esempio, ho creato un file xml 'Rubrica' che contiene alcuni nomi e numeri di persone. Inviando la parola chiave tramite un form ed utilizzando le istruzioni jquery, posso risalire al nome ed il numero nel documento. NB: ho usato la jquery perché facilita l'uso della DOM.

index.html

```
<html>
  <head>
    <title>Cerca una persona</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script type="text/javascript" src="jquery-3.5.1.js"></script>
    <script type="text/javascript" src="searchDoc.js"></script>
  </head>
  <body>
    <div id="search">
      <form name="InvioDati" method="GET">
        Cerca: <input type="text" name="key">
        <input type="button" value="CERCA" onclick="search()">
      </form>
    </div>
  </body>
</html>
```

Rubrica.xml

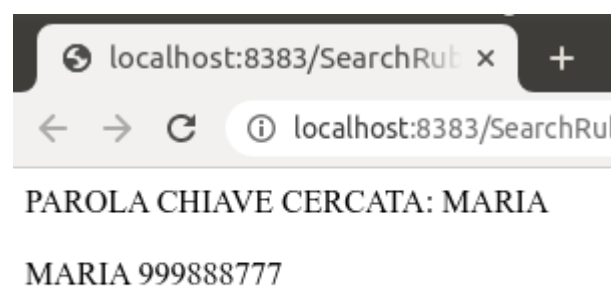
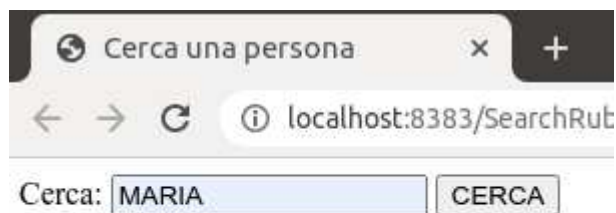
```
<?xml version="1.0" encoding="UTF-8"?>

<root>
  <Rubrica>
    <list>
      <id name='1'>
        <persona>ROBERTO</persona>
        <numero>555666333</numero>
      </id>
      <id name='2'>
        <persona>MARIA</persona>
        <numero>999888777</numero>
      </id>
      <id name='3'>
        <persona>GIACOMO</persona>
        <numero>444222111</numero>
      </id>
      <id name='4'>
        <persona>RODOLFO</persona>
        <numero>222333555</numero>
      </id>
      <id name='5'>
        <persona>MARIO</persona>
        <numero>888555222</numero>
      </id>
    </list>
  </Rubrica>
</root>
```

searchDoc.js

```
function search() {  
    var parola = document.InvioDati.key.value;  
    document.writeln("PAROLA CHIAVE CERCATA: "+parola);  
  
    $.ajax({  
        type: "GET",  
        url: "Rubrica.xml",  
        dataType: "xml",  
        success: function(xml) {  
            $(xml).find('id').each(function(){  
                var name = $(this).find('persona').text();  
                if(name==parola) {  
                    document.write("<br><br> "+parola + " "+$(this).find('numero').text());  
                }  
            });  
        }  
    });  
}
```

Risultato:



Vediamo adesso un semplice parserXML DOM:

index.html

```
<html>
  <head>
    <title>ParserDOM</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script type="text/javascript" src="Leggi.js"></script>
  </head>
  <body>
    <input type="button" value="Leggi XML" onclick="readDoc()">
    <p id="demo"></p>
  </body>
</html>
```

Documento.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<root>
  <list>
    <persona>
      <nome>UMBERTO</nome>
      <id>1</id>
    </persona>
    <persona>
      <nome>LUIGI</nome>
      <id>2</id>
    </persona>
    <persona>
      <nome>CARLO</nome>
      <id>3</id>
    </persona>
  </list>
</root>
```

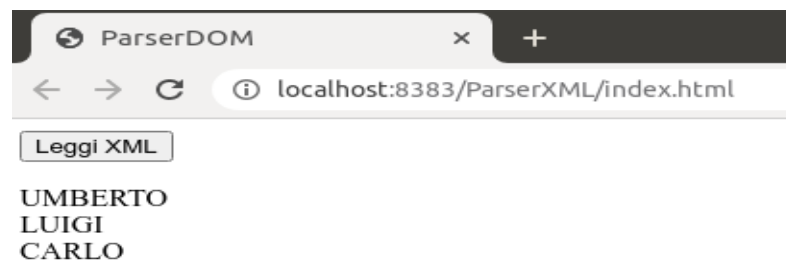
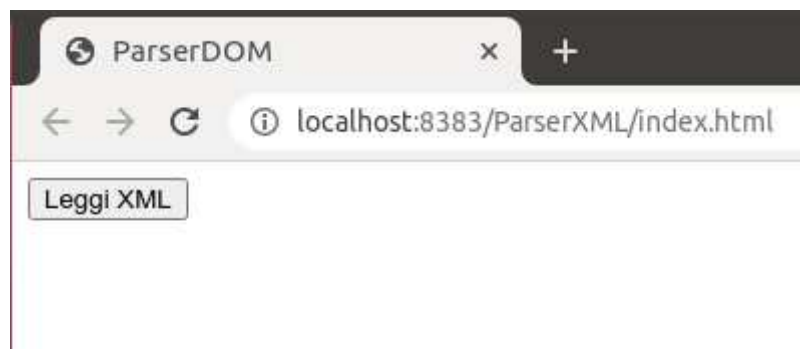
Leggi.js

```
function readDoc() {
    var xmlHttp = new XMLHttpRequest();
    xmlHttp.onreadystatechange=function() {
        if(this.readyState==4 && this.status==200) {
            RichiestaDoc(this);
        }
    };

    xmlHttp.open("GET","Documento.xml",true);
    xmlHttp.send();
}

function RichiestaDoc(xml) {
    var xmlDoc = xml.responseXML;
    var txt="";
    var x = xmlDoc.getElementsByTagName("nome");
    for (i=0;i<x.length;i++) {
        txt += x[i].childNodes[0].nodeValue + "<br>";
    }
    document.getElementById("demo").innerHTML = txt;
}
```

Risultato:



Domande possibili:

Cos'è DOM?

Si tratta di un API che permette al linguaggio di accedere a diverse parti del documento. Tale interfaccia è indipendente sia dalla piattaforma che dal linguaggio. La sua architettura è divisa in moduli, ciascuno dei quali si riferisce ad un dominio. Nelle dom, i documenti hanno una struttura logica ad albero; si tratta quindi di un modello logico, che può venir implementato in qualsiasi modo risulti più conveniente.

XML e XHTML

Come viene rappresentato una tracklist di un cd tramite xml:

```
<Music>
  <cd>
    <album name=BORN IN THE USA/>
      <artist>Name=Bruce Springsteen</artist>
      <song title="Dancing in the dark"/>
      <song title="No Surrender"/>
    </album>
  </cd>
</Music>
```

Domande possibili:

Cos'è XML?

È un linguaggio di markup usato per descrivere i dati. Esso abbandona completamente ASCII e le codifiche ad 1 byte e usa UNICODE; possiamo dire quindi che un documento XML è una stringa di caratteri e contiene tre sezioni: - Una dichiarazione che si tratta di un documento XML; - una dichiarazione opzionale del tipo di documento; - il contenuto del documento con i relativi tag.

Descrivi i due aspetti dei documenti XML.

Partiamo dal punto di vista logico: un documento XML corrisponde ad una gerarchia di informazioni, i dati sono appesi in singoli nodi di un albero; infine abbiamo un punto di vista fisico in cui un parser XML elabora il documento dall'inizio alla fine, ricostruendone la struttura. Ogni pezzo costituisce un'entità; le entità hanno dei nomi e per richiamarla si usano i riferimenti a quest'ultime.

XSL

Domande possibili:

cos'è XSL e come funziona?

Si riferisce ad una famiglia di documenti XML e consiste di 3 parti: XSLT(per trasformazioni su documenti xml), Xpath(usato da xslt per accedere a parti di un documento) e XSL-FO(per specificare indicazioni di formattazione).

In pratica, XSL prende in ingresso un documento XML e un foglio di stile XSL e produce la presentazione dei contenuti del documento XML descritta dal foglio di stile. L'elaborazione prevede

due parti: 1)trasformazione dell'albero in cui ogni albero corrisponde ad un documento xml, viene quindi costruito un altro albero secondo le indicazioni date; 2) partendo dall'albero creato, viene prodotto un risultato per essere rappresentato sul mezzo di destinazione.

PHP

Vediamo come in PHP si inviano dei dati:

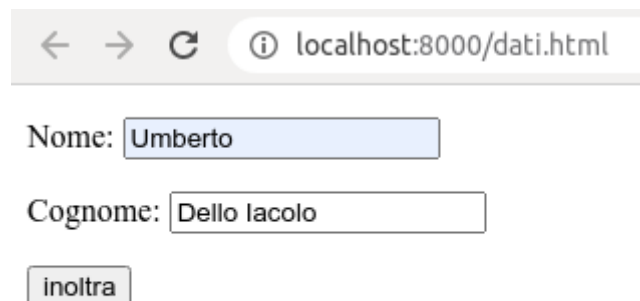
dati.html

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <title>Inserimento dati</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form action="index.php" method="post">
      <p>Nome: <input type="text" name="In1" id="In1"></p>
      <p>Cognome: <input type="text" name="In2" id="In2"></p>
      <input type="submit" name="Submit" value="inoltra">
    </form>
  </body>
</html>
```

index.php

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <meta charset="UTF-8">
    <title>Pagina PHP</title>
  </head>
  <body>
    <?php
    echo "hello world ";
    if( isset($_POST['Submit']) && !empty($_POST['In1']) && !empty($_POST['In2']) ){
      $nome = $_POST['In1'];
      $cognome = $_POST['In2'];
      echo $nome;
      echo " ";
      echo $cognome;
    }
    else{
      echo "The args was not received.";
    }
    ?>
  </body>
</html>
```

Risultato:



The screenshot shows a web browser window with the address bar displaying "localhost:8000/dati.html". Below the address bar, there is a form with two input fields. The first field is labeled "Nome:" and contains the text "Umberto". The second field is labeled "Cognome:" and contains the text "Dello Iacolo". Below these fields is a button labeled "inoltre".



Vediamo come funziona l'accesso alla base di dati:

Dati.html:

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <title>Parola chiave</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form action="index.php" method="post">
      Ricerca: <input type="text" name="key">
      <input type="submit" value="invia">
    </form>
  </body>
</html>
```

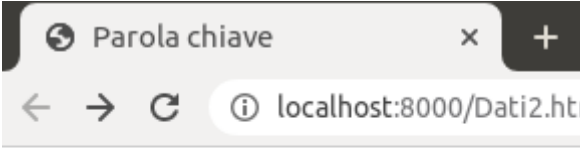
index.php:

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Risultati</title>
  </head>
  <body>
    <?php
      $user="root";
      $password="password";
      $dbName="Numerol0";
      $server="localhost";
      $con = new mysqli($server,$user,$password,$dbName);
      $key = $_POST['key'];
      $query="SELECT * FROM Famiglia WHERE Cognome='$key'";

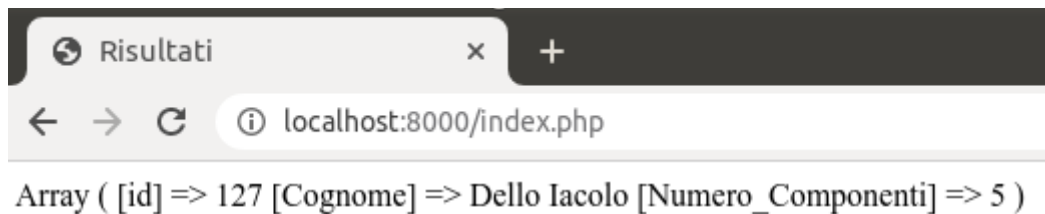
      $result=mysqli_query($con,$query);

      if (!$query) {
        echo "NESSUNA QUERY!";
      }
      else {
        while ($row = mysqli_fetch_array($result,MYSQLI_ASSOC)) {
          print_r($row);
        }
      }
    ?>
  </body>
</html>
```

Risultato:



The screenshot shows a web browser window with a single tab titled "Parola chiave". The address bar displays "localhost:8000/Dati2.htm". Below the address bar, there is a search bar with the text "Ricerca:" followed by an input field containing "Dello Iacolo" and a button labeled "invia".



Domande possibili:

cos'è PHP?

È un linguaggio di scripting lato server. Come per le jsp, PHP contiene un template e codice PHP. Quest'ultimo viene elaborato sul server e quindi al client verrà passato il template. Php è stato progettato per spedire posta, accesso alle basi di dati, scaricare documenti http e ftp ecc.

Quale novità è stata introdotta a partire da PHP5?

C'è il supporto alla programmazione ad oggetti; il nome della classe deve coincidere col nome del file. Ovviamente classi e oggetti hanno proprietà che ci aspettiamo da un linguaggio orientato agli oggetti. Come al solito, gli oggetti vengono creati con l'operatore new e si fa accesso a metodi e attributi attraverso l'operatore → . anche in questo caso, attributi e metodi possono avere i classici livelli di visibilità come public, private o abstract. Infine, php ammette ereditarietà e metodi astratti.

AJAX

Domande possibili:

Cos'è AJAX?

È una tecnica di sviluppo software che permette di programmare un'interazione asincrona rispetto alle azioni dell'utente, dove una richiesta http viene preparata e inoltrata solo quando risulta necessario. In generale, AJAX permette di ottimizzare lo scambio di dati tra client e server andando a richiedere solo i dati necessari a costruire una parte della pagina che va modificata. Il modello ajax si basa su alcuni strumenti come: JavaScript(per permettere la comunicazione col browser), DOM(per accedere e modificare la struttura della pagina), XML(per rappresentare i dati che vengono scambiati) e l'oggetto XMLHttpRequest(per permettere lo scambio asincrono tra client e server).

SEMANTIC WEB

Domande possibili:

Cos'è il semantic web?

Può essere visto come il tentativo di collegare i dati presenti nelle diverse basi di dati. L'idea è quella di raccogliere tutti i dati disponibili sul web in un'enorme base di dati.

Cos'è SOAP?

È uno strumento utilizzato per impacchettare e scambiare messaggi XML. Questi messaggi vengono trasmessi con opportuni protocolli come http, ftp ecc...

Qual è la differenza tra Registro, Indice e P2P?

Il registro, rappresenta un deposito di descrizioni di servizi; l'indice pur essendo una raccolta di informazioni esterna, non ha né controllo centrale, né autorevolezza, infatti la pubblicazione è passiva; infine il P2P non ha repositories centralizzate, sono i servizi che si scoprono vicendevolmente in modo dinamico passandosi richieste sulla rete. È più robusto perché non dipende da un solo punto centrale. Infatti ogni nodo può contenere un proprio indice dei servizi web esistenti, inoltre la latenza è minore in quanto i nodi vengono contattati direttamente.

Come funziona il parser a DOM?

Un parser a DOM, accede ad un documento XML utilizzando una struttura logica ad albero, infatti restituisce un oggetto di tipo albero in cui ogni nodo rappresenta l'elemento di un documento. L'approccio diventa rischioso al crescere delle dimensioni del documento.

Come funziona il parser a EVENTI?

Il parser a eventi, ad ogni passo, rende disponibile all'applicazione il frammento di dati che corrisponde ad un singolo evento nel documento di ingresso.