

sd1.md

DOMANDE PRIMO PARZIALE

- Nel contesto attuale delle tecnologie web, il browser:

- È semplicemente un sistema per la richiesta di pagine HTML, contenuti multimediali associati, e loro rendering sullo schermo dell'utente
- È un sistema per la richiesta di pagine HTML e contenuti multimediali associati tramite protocollo HTTP, che ne gestisce anche il rendering sullo schermo dell'utente. Le librerie per lo sviluppo di web application (ad esempio DOM) sono presenti ma non attive di default, per risparmiare memoria
- È un sistema modulare, con funzionalità base comuni (rendering delle pagine HTML, anche comprendendo file CSS) e altre funzionalità instabili su richiesta in stile plugin (in particolare la libreria DOM per lo sviluppo di web application)
- **È un ambiente composto ed estensibile, comprendente di norma un interprete JavaScript, librerie che implementano API quali il DOM, funzionalità di supporto allo sviluppo di web application, oltre la semplice rendering delle pagine HTML**

- Qual è la struttura di un elemento di base di un CSS?

- Essendo un elemento minimo permette di specificare che per un tag identificato da un selettore vale un elemento stilistico (in forma di coppia attributo-valore)
- Ogni elemento base di un file CSS permette di indicare uno stile (sotto forma di un insieme coppie attributo- valore) da associare a elementi specificate sulla base della posizione nell'albero DOM
- Un elemento base di un CSS è una regola che associa a un tag (o un elenco di tag) un insieme di elementi stilistici (nella forma di coppie attributo-valore)
- **Un elemento base di un CSS è una regola che specifica un selettore (una sorta di query per identificare un tag o un insieme di tag nella pagina) e una serie di coppie attributo-valore**

- Una media query in un CSS permette:

- **Di specificare regole di adattività dei contenuti (tanto in termini di stile che di struttura del documento HTML) sulla base delle caratteristiche del device usato per visualizzare la pagina**
- Esclusivamente di cambiare la posizione e dimensione di blocchi della pagina, in modo da poter adattare la presentazione a schermi più alti che larghi (come quelli degli smartphone).
- La definizione di regole statistiche (insiemi di coppie attributo-valore) selezionate in base alle caratteristiche del medium usato (che può essere lo schermo del device, oppure una stampante o altro)
- Di adattare i contenuti alle caratteristiche dello schermo del device usato, grazie alla presenza di variabili che possono considerare attributi come la larghezza dello schermo

- Qual è, in estrema sintesi, il ruolo del linguaggio HTML?

- HTML, nella sua versione 5, è un linguaggio per la programmazione di pagine web, compreso lo stile estetico e la capacità di reagire ad eventi generati dall'utente
- HTML, nella sua versione 5, è un linguaggio di markup per definire contenuti di una pagina web e relativo stile di presentazione
- HTML, nella sua versione 5, permette la specifica della struttura di contenuti web, tra i quali elementi semantici il cui uso scorretto viene identificato e segnalato dal browser
- **HTML, nella sua versione 5, è un linguaggio di markup per la specifica di contenuti e struttura di pagine web**

- JavaScript è:

- Un linguaggio di scripting interpretato derivante da una riduzione di Java adeguata all'esecuzione di semplici script all'interno di browser web
- Un linguaggio di scripting interpretato e dinamico pensato per scripting all'interno di un browser web, ma oggi estensibile grazie alla possibilità di importare librerie scritte in altri linguaggi
- Un linguaggio di scripting interpretato, dinamico e debolmente tipizzato esclusivamente usabile all'interno di browser per realizzare applicazioni, oggi anche piuttosto complicate
- **Un linguaggio di scripting interpretato, dinamico e debolmente tipizzato inizialmente pensato per l'esecuzione di semplici script all'interno del browser web**

- **Quali affermazioni sono VERE? L'invocazione da parte di un server di un accept su una socket s1 determina:**
 - L'associazione di una porta alla socket s1 per poter ricevere dal client
 - La lettura del primo messaggio di richiesta da parte del client dalla socket s1
 - L'abilitazione della socket s1 a colloquiare con il client
 - **La creazione di una nuova socket s2 per colloquiare con il client**
- **A un server multi-threaded sono collegati 3 processi client. Quante sono le socket aperte per il processo che esegue il server?**
 - 2
 - 1
 - 3
 - **Più di 3 (del dettaglio 4)**
- **Un sender esegue l'istruzione (in pseudo codice): write(socket, buffer, N) per scrivere sulla socket gli N byte contenuti nel buffer. QUALI di queste affermazioni sono FALSE (2 risposte) [non ho capito quale dovrebbe essere la seconda vera]?**
 - Il receiver può leggere l'intero buffer con un ciclo di istruzioni: read(socket, buffer, N/2)
 - **Il receiver può leggere l'intero buffer con una istruzione: read(socket, buffer, 2*N)**
 - Il receiver può leggere l'intero buffer con un ciclo di istruzioni: read(socket, buffer, N)
 - **Il receiver può leggere l'intero buffer con una istruzione: read(socket, buffer, N)**
- **Dato un processo P1 che apre una socket s1 e poi esegue una fork() generando un processo P2. Cosa succede se i due processi leggono entrambi sulla stessa socket s1? [forse secondo parziale]**
 - **leggono in modo concorrente lo stesso canale**
 - P2 riceve un messaggio di errore perchè il canale è riservato a P1
 - P1 riceve un messaggio di errore perchè il canale è riservato a P2
 - Leggono dati diversi perchè ciascuno ha un proprio canale
- **Le socket permettono a due pari (client e server) di comunicare tramite il protocollo:**
 - HTML
 - HTTP
 - FTP
 - **TCP/IP**
- **Quale di queste affermazioni è FALSA:**
 - La system call bind serve ad associare una porta nota ad un server socket;
 - Per leggere tutti i dati da una socket serve sempre un ciclo di lettura;
 - Se il client effettua una system call connect verso un server socket occupata viene messo in coda;
 - **Non è possibile associare una porta nota (well-known port) ad un client socket.**
- **La trasmissione di messaggi tramite socket in Java avviene attraverso:**
 - **TCP/IP attraverso flussi di byte (byte stream) dopo una connessione esplicita tramite normali system call read/write (sono suspensive/bloccanti e utilizzano buffer per garantire flessibilità)**
 - ...
- **L'invocazione da parte di un server di una accept su una socket determina:**
 - la definizione del numero di richieste che il server è disposto ad accettare
 - la creazione di una nuova socket su cui accettare richieste di connessione
 - **la sospensione del server finché non arriva una richiesta di connessione**
 - l'accettazione di un nuovo messaggio di richiesta
- **Quando il client e il server inizializzano le socket, che tipo di socket devono inizializzare?**
 - Il server inizializza una Socket, il client inizializza una ServerSocket.
 - *** Il server inizializza una ServerSocket, il client inizializza una Socket***
 - Il server e il client devono entrambi creare una Socket e una ServerSocket altrimenti non possono leggere e scrivere.

- Nessuna delle risposte precedenti.
- **Quale delle seguenti affermazioni è corretta a proposito della comunicazione tra client e server?**
 - Il server usa un InputStream per ricevere dati dal client, e il client usa un OutputStream per dare informazioni al server
 - Il server e il client hanno "due connessioni", una per leggere e l'altra per scrivere
 - Il server usa un OutputStream per ricevere dati dal client, e il client usa un InputStream per dare dati al server
 - **Il server e il client hanno entrambi un InputStream e un OutputStream**
- **Qual è il vantaggio di utilizzare Socket Multicast? [forse non in programma]**
 - Le socket Multicast sono più facili da implementare rispetto a quelle Broadcast e Unicast
 - **È generalmente più efficiente usare one socket Multicast che usare socket Unicast multiple**
 - Le socket Multicast, a differenza di quelle Broadcast, toccano tutti i nodi/destinazioni
 - Le socket Multicast utilizzano più banda larga/risorse rispetto alle socket Broadcast e Multicast
- **Perché i servizi REST sono più conformi al modello Web di quelli WSDL/SOAP?**
 - Perché permettono di definire API personalizzate
 - Perché permettono l'utilizzo di interfacce AJAX
 - **Perché usano il protocollo HTTP in modo nativo**
 - Perché usano il protocollo HTTP come vettore per le chiamate di procedura
- **Perché il ciclo di vita delle HTTPServlet può essere gestito da un engine?**
 - **Perché hanno un'interfaccia nota**
 - Perché usano il protocollo HTTP
 - Perché Java è un linguaggio interpretato
 - Perché generano risposte HTML
- **REST (due risposte):**
 - **Rende centrale il concetto di risorsa**
 - È uno standard codificato
 - **È un insieme di guidelines e best practices**
 - Non è in grado di appoggiarsi e sfruttare HTTP
- **Dopo una prima richiesta HTTP GET viene inviata una richiesta HTTP POST. Secondo il modello Java Servlet, sarà invocato per primo:**
 - Il metodo init() **forse giusta**
 - **Il metodo service(...)**
 - Il metodo destroy()
 - Il metodo doPost()
- **In HTTP, quale è una funzione primaria delle coppie nome-valore contenute nell'header?**
 - **Indicare il formato dei dati del payload**
 - Trasferire i parametri passati con le richieste
 - Indicare la dimensione dell'intero messaggio HTTP
 - Definire il valore dei cookie scambiati tra client e server
- **In HTTP, che cosa significa che una operazione è "safe"?**
 - Che l'esecuzione multipla della stessa richiesta produce lo stesso effetto di una singola esecuzione
 - **Che l'esecuzione della richiesta non modifica lo stato della risorsa coinvolta.**
 - Che l'esecuzione multipla della stessa richiesta produce lo stesso effetto di un'altra operazione.
 - Che l'esecuzione della richiesta modifica lo stato della risorsa coinvolta.
- **Quale è la definizione corretta di middleware?**
 - È uno strato software che virtualizza l'hardware e fornisce servizi alle applicazioni soprastanti
 - **È uno strato software che sta sopra i sistemi operativi di rete e fornisce servizi alle applicazioni soprastanti**

- È uno strato software che sta sopra il sistema operativo di rete per simulare un sistema operativo distribuito
- È uno strato software che sostituisce i sistemi operativi di rete e fornisce servizi alle applicazioni soprastanti
- **In riferimento all'architettura a 3-tier, la tecnologia Servlet e JSP è utile per (2 risposte)**
 - **Il tier di presentazione**
 - **Il tier della logica applicativa**
 - Non è una tecnologia adeguata a questo genere di architettura
 - Il tier dei dati persistenti
- **Cosa significa il termine stateless nel contesto della Service Oriented Architecture (SOA)?**
 - Che non si possono realizzare sessioni di lavoro
 - **Che lo stato di un servizio non dipende dallo stato di un altro servizio**
 - Che i client di un servizio non possono avere stato
 - Che i servizi non possono avere stato
- **Il concetto di trasparenza nei sistemi distribuiti significa che:**
 - Occorre utilizzare meccanismi di base per accedere alle funzionalità desiderate
 - È possibile accedere alle funzionalità di più basso livello rispetto alle funzionalità utilizzate
 - **Non è necessario conoscere i dettagli con cui vengono realizzate le funzionalità utilizzate**
 - Sono visibili i dettagli con cui vengono realizzate le funzionalità utilizzate
- **Il termine Failure Transparency sta ad indicare che...**
 - L'utente può riconoscere un fallimento parziale e porvi rimedio
 - Il sistema operativo può far ripartire un server dopo un fallimento
 - Un sistema è in grado di effettuare un abort per riportare ad uno stato consistente dopo un fallimento
 - **Un sistema è in grado di portare a termine un compito anche in presenza di fallimenti parziali**
- **La realizzazione di un server basato su socket TCP/IP può essere critica. Quale delle seguenti affermazioni è FALSA?**
 - La connessione potrebbe non essere chiusa correttamente generando problemi di blocco del servizio
 - L'invio di stream di bytes rende arbitrario il tempo di ricezione dei messaggi
 - La necessità di una connessione per client può comportare ritardi nel servizio
 - **L'invio di stream di bytes limita il tipo di messaggi da inviare**
- **Il termine Persistence Transparency sta ad indicare che...**
 - È possibile memorizzare i dati via rete;
 - È possibile memorizzare i dati su disco;
 - Una risorsa (software o dati) viene conservata in memoria;
 - **Una risorsa (software o dati) può essere in memoria o su disco.**
- **Il termine Migration/Mobility Transparency sta ad indicare che...**
 - È possibile portare il software su macchine diverse senza modifiche
 - Lo spostamento di dati e/o software comporta modifiche nei programmi
 - **Lo spostamento di dati e/o software non comporta modifiche nei programmi utente**
 - È possibile accedere a dati e/o software da qualsiasi punto della rete
- **Quali vantaggi o svantaggi comporta un'organizzazione a messaggi di lunghezza fissa?**
 - **Semplificazione della gestione dei buffer**
 - Minore velocità di trasmissione
 - Limitazione nella lunghezza di invio
 - Maggiore velocità di trasmissione
- **Con il termine Asynchronous RCP si intende che:**
 - il server restituisce il controllo alla ricezione della richiesta;
 - il server non si blocca quando riceve una chiamata;

- *il client non si blocca quando esegue una chiamata;*
- il server restituisce il controllo al completamento della richiesta;
- **Perché è necessario usare la "select" per realizzare server concorrenti?**
 - Perché le operazioni di read() e write sono bloccanti
 - **Perché permette di utilizzare più canali di comunicazione**
 - Perché è possibile selezionare il canale da cui leggere e/o scrivere
 - Perché non è più necessario eseguire la "accept" per le connessioni
- **Se N client inviano una richiesta GET alla stessa risorsa quante istanze della servlet che la gestisce vengono attivate?**
 - **Una, cui accedono tutti i client in concorrenza**
 - N, una per ogni client
 - Una o N, come stabilito dal codice della servlet
 - Una, cui accedono tutti i client per mutua esclusione
- **Un sender esegue l'istruzione (in pseudo codice): write(socket, buffer, N) per scrivere sulla socket gli N byte contenuti nel buffer. QUALE è falsa? [bho]**
 - Il receiver può leggere l'intero buffer con un ciclo di istruzioni: read(socket, buffer, N/2)
 - Il receiver può leggere l'intero buffer con un ciclo di istruzioni: read(socket, buffer, 1)
 - Il receiver può leggere l'intero buffer con un ciclo di istruzioni: read(socket, buffer, N)
 - **Il receiver può leggere l'intero buffer con una istruzione: read(socket, buffer, N)**
- **In quale scope è necessario mettere una variabile perché sia letta globalmente da diversi utenti e da diverse servlet?**
 - Session
 - Page
 - **Application**
 - Request
- **Quali metodi dell'interfaccia Serializable è obbligatorio implementare?**
 - **Nessuno, Serializable è una classe marker, quindi non è obbligatorio implementare alcun metodo**
 - È obbligatorio implementare onSerialization e onDeserialization
 - È obbligatorio implementare i metodi readObject e writeObject
- **Perché nel contesto SOA è fondamentale definire delle interfacce, come WSDL, indipendenti dal linguaggio per la realizzazione del servizio? [?]**
 - Perché è più semplice automatizzare la gestione dei servizi
 - **Per consentire la composizione di servizi realizzati da terze parti**
 - Per separare meglio i compiti assegnati ai servizi di una applicazione
 - Per migliorare le performance dei servizi realizzati