

---

Domande utili per verificare  
la propria preparazione

---

---

# Domande di verifica (1)

- Descrivere l'architettura di una web application a due livelli, contenente cioè un web server e un database server. Spiegare le modalità di interazione e i protocolli coinvolti.
  - Descrivere come l'interazione client-server possa scatenare l'esecuzione di programmi java da parte del server (protocolli e oggetti coinvolti)
  - Definire il concetto di sessione di navigazione, spiegare perché la sessione non sempre debba seguire delle operazioni di autenticazione. Dare opzionalmente un esempio.
  - Spiegare perché il protocollo HTTP sia definito stateless, e elencare tutte le tecniche note per il mantenimento di informazioni relative alla sessione di navigazione di un utente
  - Spiegare possibili utilizzi della tecnica di URL rewriting.
-

---

## Domande di verifica (2)

- Descrivere tecniche di programmazione lato server basate sull'uso di oggetti persistenti per il mantenimento di informazioni relative alla sessione di navigazione di un utente
  - Descrivere l'utilizzo di campi input di tipo hidden nei form HTML per il mantenimento di informazioni sulla sessione di navigazione di un utente.
  - Descrivere l'utilizzo di cookie per il mantenimento di informazioni sulla sessione di navigazione di un utente.
  - Descrivere il ciclo di vita di una Servlet e di una HttpServlet
  - Descrivere il ciclo di vita di una pagina JSP, chiarendo tutti gli elementi applicativi necessari alla sua esecuzione
  - Affinche` una pagina JSP possa essere eseguita, e` necessario che il servlet container disponga di un compilatore java. Spiegarne le ragioni.
-

---

## Domande di verifica (3)

- Descrivere lo scopo del deployment descriptor indicandone i campi che si ritiene abbiano valore fondamentale nel funzionamento di una web application
  - Descrivere quali opzioni di configurazione di una applicazione sono necessarie, nel deployment descriptor e in altri descrittori xml affinché si possa eseguire correttamente una pagina JSP
  - Descrivere e motivare la struttura delle cartelle di una web application
  - Illustrare i metodi noti per la redirectione di richieste da una servlet ad altre risorse (servlet o pagine)
  - Illustrare i metodi noti per la redirectione di richieste da una pagina jsp ad altre risorse (servlet o pagine)
-

---

# Domande di verifica (4)

- *Spiegare come avviene l'esecuzione e l'interpretazione del codice misto presente in una pagina JSP, in particolare descrivere l'elaborazione dei seguenti elementi:*
    - *direttive,*
    - *scriptlet,*
    - *espressioni,*
    - *dichiarazioni,*
    - *commenti html,*
    - *commenti jsp,*
    - *commenti java,*
    - *Azioni standard JSP: include,*
    - *Azioni standard JSP: forward,*
    - *Azioni standard JSP: useBean*
      - *Azioni standard JSP: setProperty,*
      - *Azioni standard JSP: getProperty*
    - *Tag personalizzati che implementano l'interfaccia Tag*
    - *Tag personalizzati che implementano l'interfaccia BodyTag*
-

---

# Domande di verifica (5)

- *Differenza tra direttiva e azione standard di inclusione*
  - *Descrivere prima teoricamente e poi con un esempio l'utilità dei bean all'interno di pagine JSP*
  - *Ciclo di vita di un tag che implementa l'interfaccia Tag*
  - *Ciclo di vita di un tag che implementa l'interfaccia BodyTag*
  - *Descrivere la gestione a stack degli oggetti BodyContent nei tag personalizzati che implementano l'interfaccia BodyTag*
  - *Descrivere i metodi di autenticazione dichiarativa supportati dal container TOMCAT*
-

---

# Domande di verifica (6)

- *Descrivere un possibile approccio completamente programmato all'autenticazione, definire le operazioni fondamentali e suggerire buone pratiche di implementazione.*
  - *Descrivere i passaggi fondamentali per realizzare una connessione e la gestione di un database da una applicazione web basata su servlet e jsp*
  - *Descrivere problemi prestazionali nell'uso di connessioni a un database da parte di una applicazione web, e indicare possibili strategie risolutive*
-

---

# Domande di verifica (6)

- *Descrivere l'uso dei bean nelle pagine jsp. In particolare si spieghi come le informazioni di un bean istanziato in una pagina jsp attraverso un'azione standard, in tutti i possibili contesti di visibilità, possano essere reperite da una servlet che possa accedere all'oggetto rappresentativo del contesto di visibilità del bean stesso.*
  - *Si discutano i meccanismi di gestione di informazioni persistenti tramite JSP. In particolare si discutano i diversi contesti di visibilità con riferimento alle opportune classi, interfacce e metodi utilizzati. Fare un esempio per ciascun contesto di visibilità, discutendone i possibili utilizzi.*
-



---

# Esercizio 1

Si scriva un'applicazione basata su servlet che tenga traccia delle sessioni utente attraverso oggetti persistenti sul server. Si mostri un esempio di pagina di login e di servlet di autenticazione che permettano la creazione di una sessione di navigazione in caso di autenticazione corretta.

---

---

## Esercizio 2

Si scrivano due servlet (A e B) che utilizzino un contatore condiviso di accessi all'applicazione. Il contatore deve essere condiviso, e deve essere incrementato ad ogni accesso, indipendentemente da quale sia la servlet a cui si accede. Ciascuna servlet deve produrre come risposta una pagina html che visualizzi il valore del suddetto contatore. Descrivere tutti i file necessari al funzionamento dell'applicazione e descrivere la struttura della context root.

---

---

## Esercizio 3

Scrivere una servlet (servlet\_uno) che legga da un form nome e password di un utente. Una volta autenticato l'utente deve essere associato alla sessione un oggetto che lo rappresenti (che contenga almeno il nome), nel quale sia presente un parametro che dica se l'utente è amministratore o no. La servlet produca una pagina di risposta di avvenuta o errata autenticazione.

---

---

## Esercizio 4

*Scrivere una servlet che realizzi una bacheca di messaggi. Quando questa servlet viene interrogata senza parametri nella richiesta deve produrre un form attraverso il quale un utente possa inserire un messaggio testuale e in fondo alla pagina deve visualizzare tutti i messaggi ricevuti precedentemente, se ve ne sono. Questa stessa pagina costituirà anche la destinazione del form citato sopra e acquisirà dunque nuovi messaggi, producendo anche in questo caso il form di inserimento e visualizzando in fondo alla pagina tutti i messaggi ricevuti, compreso quello della richiesta corrente a cui la pagina sta rispondendo.*

---

---

## Esercizio 5

*Scrivere un'applicazione web che realizzi una pagina di sondaggio elettronico per la scelta di uno tra cinque candidati sindaci. L'applicazione deve prevedere l'impostazione di un cookie di registrazione del voto sulla macchina dei client che hanno già espresso la loro preferenza.*

*Se la macchina da cui si connette il client non contiene il cookie di registrazione del voto, l'utente è ammesso alla pagina di voto, altrimenti l'utente deve essere automaticamente ridiretto su una pagina di statistiche che visualizza le percentuali di voti acquisiti dai cinque candidati.*

*Qualora l'utente sia ammesso al voto, dopo l'espressione della propria preferenza, oltre all'impostazione del cookie di registrazione del voto, deve essere effettuata la ridirezione interna del client sulla pagina di statistiche menzionata in precedenza.*

---

---

## Esercizio 6

*Scrivere una pagina servlet che legga da un form alcuni dati anagrafici di un utente e li memorizzi in un apposito oggetto con visibilità di sessione.*

*La servlet dovrà poi aggiungere alla sessione un parametro booleano (confronto) corrispondente al fatto che l'utente sia nato prima o dopo il 2000 (vero se prima del 2000, falso altrimenti) e redirigere l'utente verso una servlet (servlet\_destinazione) con una redirezione esterna/interna.*

*La servlet di destinazione produrrà una pagina personalizzata a seconda del valore del parametro confronto.*

---

---

## Esercizio 7

*Si mostri un esempio di pagina di login e di servlet di autenticazione che permettano la creazione di una sessione di navigazione solo in caso di autenticazione corretta, inviando all'utente una serie di tre domande in sequenza, a cui l'utente dovrà rispondere singolarmente in tre richieste http consecutive, che permettano alla servlet di acquisire tre valori nome, età, nazionalità. Al termine dei tre inserimenti la servlet produrrà un messaggio riassuntivo dei dati acquisiti.*

---

---

## Esercizio 8

*Si scriva una servlet che, in risposta a un form in cui l'utente possa inserire il proprio nome, fornisca un messaggio di saluto.*

*La suddetta servlet dovrà creare una sessione per ciascun nuovo utente e visualizzare nel relativo messaggio di saluto un contatore del numero di sessioni istanziate fino al momento dell'interrogazione. Se la servlet viene interrogata più volte consecutive dallo stesso utente, il contatore di sessioni NON deve essere aggiornato, ma solo visualizzato insieme al messaggio di benvenuto.*

---



---

## Esercizio 9

*Si scriva una applicazione web che realizzi un sondaggio consentendo agli utenti di eseguire un test con dieci domande in sequenza con risposta binaria, si` o no. Si realizzi una pagina JSP che visualizzi le statistiche di risposta alle 10 domande (utilizzare un tag che iterativamente scandisca e visualizzi le percentuali di risposta).*

---

---

## Esercizio 10

*Scrivere una pagina jsp che realizzi una bacheca di messaggi. Quando questa pagina viene interrogata senza parametri nella richiesta deve produrre un form attraverso il quale un utente possa inserire un messaggio testuale e in fondo alla pagina deve visualizzare tutti i messaggi ricevuti precedentemente, se ve ne sono.*

*Questa stessa pagina costituirà anche la destinazione del form citato sopra e acquisirà dunque nuovi messaggi, producendo anche in questo caso il form di inserimento e visualizzando in fondo alla pagina tutti i messaggi ricevuti, compreso quello della richiesta corrente a cui la pagina sta rispondendo.*

---

---

# Esercizio 11

*Scrivere una pagina jsp (pagina.jsp) che legga da un form alcuni dati anagrafici di un utente e li memorizzi in un apposito bean con visibilità di sessione. Per fare questo si utilizzi un'azione standard di popolamento dei dati del bean attraverso i parametri del form. La pagina jsp dovrà poi aggiungere alla sessione, attraverso un'ulteriore azione standard, un parametro booleano (confronto) corrispondente al fatto che l'utente sia nato prima o dopo il 2001 (vero se prima del 2001, falso altrimenti) e redirigere l'utente verso una servlet (servlet\_destinazione) con una redirezione esterna. La servlet di destinazione produrrà una pagina personalizzata a seconda del valore del parametro confronto.*

---

---

## Esercizio 12

*Scrivere una servlet (servlet\_uno) che legga da un form nome e password di un utente. Una volta autenticato, l'utente deve essere associato alla sessione con un bean che lo rappresenti (che contenga almeno il nome), nel quale sia presente un parametro che dica se l'utente è amministratore o no. La servlet produca una pagina di risposta di avvenuta o errata autenticazione.*

*Si preveda poi nell'applicazione l'uso di pagine jsp contenenti un tag personalizzato con un parametro admin il cui valore venga letto dal bean di sessione. Tale tag personalizzato dovrà aggiungere alla pagina in cui si trova un pannello di amministrazione solo nel caso in cui l'utente si sia autenticato come amministratore. (Il tag non aggiunge nulla per utenti non amministratori o non autenticati). Si faccia un esempio di pagina jsp che usi il suddetto tag.*

---

---

# Esercizio 13

*1) Descrivere la tecnica di URL rewriting per il mantenimento di informazioni relative alla sessione di navigazione degli utenti quando non sia noto se i relativi browser consentano l'uso di cookie.*

*In particolare:*

*a) Spiegare quali URL richiedono riscrittura, quando, e come effettuarla.*

*b) Descrivere una possibile implementazione del metodo `encodeURL()` dell'interfaccia `HttpServletResponse`.*

*c) In quali casi è necessario riscrivere l'URL di un percorso di redirectione?*

---

---

# Esercizio 14

*Si descriva il ciclo di vita di un tag personalizzato che implementa l'interfaccia BodyTag.*

*Si faccia un esempio di utilizzo di questa interfaccia per elaborare iterativamente il corpo di un tag, mostrando il codice della classe handler del tag e una pagina jsp che ne esemplifichi l'utilizzo.*

---