|  |  |
| --- | --- |
|  |  |
|  | C:\Users\hamadi\AppData\Local\Microsoft\Windows\INetCache\Content.Word\logo_inti_vector-04.jpg  Institut des Nouvelles Technologies  de l’Information |

QCM

JSP Servlet EJB

Session 2016

## Guide d'utilisation

* Notez qu'en fonction de la question, il est possible de cocher une ou plusieurs réponses. Ce sera à vous de déterminer le nombre de bonnes réponses possibles.

## Quelques conseils de mise en œuvre

* Ne restez pas bloqué sur une question pour laquelle vous n'avez pas de réponse. Passez à la question suivante. Vous reviendrez par la suite sur les questions sans réponse.
* Identifiez les questions auxquelles vous avez répondu sans certitude. Que ces réponses soient justes ou non, cela vous permettra de mémoriser les questions sur lesquelles vous devrez revenir dans la correction du TEST.
* Ne cherchez pas à aller le plus vite possible. Le temps imparti doit être utilisé au maximum pour vous permettre d'effectuer une synthèse des points abordés dans le module.

**QCM**

**1. Que peut contenir une JSP ?**

* Du code Java et du HTML
* Du code Java et du XML
* Du code Java et du Javascript
* Les trois

**2. La Jsp doit être compilée par le serveur.**

* Vrai, ne sera ensuite affiché que du HTML
* Vrai, mais le code Java reste visible
* Faux, elle est compilée par le client
* Faux, elle n’est pas compilée

**3. Lequel de ces codes ne permet pas d’afficher une variable laVar ?**

* <%= laVar %>
* ${ laVar }
* <% out.print(laVar) %>
* <% laVar %>

**4. Comment appelle-t-on le ${ } ?**

* Expression Tag
* Expression Language
* Markup Language
* Interpreted Language

**5. Qu’est-ce qu’un Taglib ?**

* Une librairie JSTL
* Des balises liées à des classes Java par un schéma XML
* Des balises appelant des JSP
* Des balises HTML spéciales pour Java

**6. En considérant les préfixes *c* pour le core et *fmt* pour le format, quelle est la syntaxe pour faire une boucle avec JSTL ?**

* <c:forEach items=”truc” var=”machin”></c:forEach> où *items* est la variable interne à la boucle et *var* la liste
* <fmt:forEach items=”truc” var=”machin”></fmt:forEach> où *items* est la variable interne à la boucle et *var* la liste
* <c:forEach items=”truc” var=”machin”></c:forEach> où *items* est la liste et *var* la variable interne à la boucle
* <fmt:forEach items=”truc” var=”machin”></fmt:forEach> où *items* est la liste et *var* la variable interne à la boucle

**7. De quelle classe doit hériter une *servlet* ?**

* HttpRequest
* Servlet
* HttpServlet
* RequestServlet

**8. Laquelle de ces méthodes n’est pas disponible dans une *servlet* ?**

* doPost()
* forward()
* doDelete()
* doGet()

**9. Quelle méthode sera appelée lors de l’accès d’une *servlet* par son URL sur un navigateur ?**

* doPost()
* forward()
* doGet()
* processRequest()

**10. Dans une *servlet*, quelle méthode permet de faire une redirection ?**

* response.redirect(“url/de/redirection”);
* request.getRequestDispatcher(“url/de/redirection”).forward(request, response);
* response.sendRedirect(“url/de/redirection”);
* request.forward(“url/de/redirection”);

**11. Que permet le patron MVC ?**

* L’interaction entre du code java et du code html
* La séparation des couches présentation, gestion des requêtes et modèle de données
* La séparation des couches entités, DAO et business
* L’appel de JSP au sein de *Servlet*

**12. Comment stocker une valeur qui pourra être accédée entre plusieurs JSP et *Servlet* après plusieurs requêtes ?**

* request.setAttribute(“clef”, valeur);
* request.getSession().setParameter(“clef”, valeur);
* request.setParameter(“clef”, valeur);
* request.getSession().setAttribute(“clef”, valeur);

**13. Quelle est l’erreur dans le code HTML suivant ?**

<p class=”laClasse”><span class=”laClasse”>test</span><img alt=”une image” title=”une image” src=”url/image.jpg” /> </p>

* La balise img n’est pas fermée
* En HTML il faut toujours utiliser des simple quotes pour les attributs
* Une classe ne peut être assignée qu’à une seule balise
* Il n’y a pas d’erreur

**14. Quelle est la version actuelle des Ejb ?**

* 1
* 2
* 3
* 4

**15. Un projet utilisant des Ejb a-t-il besoin de connaître leur implémentation ?**

* Oui, mais seulement dans le build path, pour le build
* Oui, mais seulement dans le deployment assembly
* Oui, dans tous les cas
* Non

**16. Quels sont les 3 types d’Ejb ?**

* Ejb Session, Ejb Message, Ejb Entity
* Ejb Stateful, Ejb Local, Ejb Remote
* Ejb Stateless, Ejb Instance, Ejb Singleton
* Ejb Implementation, Ejb Interface, Ejb Proxy

**17. Comment instancier un Ejb ?**

* Via la méthode lookup()
* Via l’annotation @EJB
* Via un new Ejb()
* L’instanciation des Ejb est gérée par le serveur

**18. Quelle annotation faut-il sur une interface pour qu’elle soit accessible dans un autre projet sur le même serveur ?**

* @Local
* @EJB
* @Stateless
* @Remote

**19. Quelle annotation permet de conserver l’état d’un Ejb entre deux appels d’un même utilisateur ?**

* @Stateless
* @Stateful
* @Session
* @Persist

**20. Un projet Ejb peut être déployé sur n’importe quel type de serveur.**

* Vrai
* Faux

FIN