

M1IF 13 – Programmation Web avancée et mobile

Contrôle continu terminal

Durée : 1 heure 30 – Documents, ordinateurs, calculatrices et téléphones portables interdits

Identification

Anonymisez votre copie, et remplissez les parties « Diplôme », « Épreuve » et « Date ».

Ensuite, vous écrirez directement vos réponses sur cet énoncé. Les réponses écrites sur la copie d'examen ne seront pas lues.

Numéro de copie :

Cours (16 points)

1. Pourquoi `Node.js` ne comporte-t-il pas de `Browser Object Model (BOM)`, et qu'est-ce que cela implique pour le code tournant dans cette plateforme ? (1 pt.)
Le BOM est fourni par le navigateur ; le code ne peut donc pas utiliser les API du browser tels que le DOM, l'objet `window`, etc.
2. Citez 4 types (génériques) de fichiers qui peuvent être packagés dans un `bundle webpack`. (2 pts.)
`js`, `json`, `css`, `image`, `police`...
3. Indiquez 2 raisons pour lesquelles le fait de packager des fichiers dans un `bundle webpack` permet de gagner en performance au moment du chargement de l'application. (1 pt.)
Les fichiers sont minifiés et il y a une seule transaction HTTP pour les obtenir.
4. Que trouve-t-on dans un fichier `.vue` représentant un composant `VueJS` ? (1 pt.)
le template, le code JS et le CSS de ce composant.
5. Indiquez brièvement comment s'utilise le Router de `VueJS`. (2 pts.)
Il associe une route côté client (hash de l'URL) à un composant. Il se configure à l'aide d'un objet JSON spécifiant ces associations et d'éléments HTML `router-link` indiquant où placer les composants templétés.
6. Donnez un exemple d'utilisation des `Custom Events` dans `VueJS` dans une application. (1 pt.)
Échanger des messages entre composants éloignés dans l'arborescence.
7. Quelle est la différence principale entre `Actions` et `Mutations` dans un `store Vuex` ? (1 pt.)
Les mutations sont synchrones et les actions peuvent être asynchrones.
8. Citez 4 moyens de persister de l'information côté client et indiquez une caractéristique de chacun d'eux (2 pts.)
Cookie : petites quantités de données, lié à la session
SessionStorage : grandes quantités de données, lié à la session
LocalStorage : grandes quantités de données, indépendant de la session
Service worker : cache de données et de code applicatif
9. Citez 2 noms de `Device APIs HTML5` pour lesquelles le navigateur demande à l'utilisateur l'autorisation de les exécuter. (1 pt.)
Battery, Media capture, Geolocation.
10. Indiquez (par du texte ou du code) comment détecter le `multitouch` dans un navigateur (1 pt.)
S'abonner à l'événement `touchstart` et utiliser l'attribut `length` de la `TouchList` touches.
11. Citez un cas d'utilisation précis pour lequel il est préférable d'utiliser des `WebSocket` plutôt que des requêtes-réponses HTTP. (1 pt.)
Envoi de données « streamées » depuis un capteur de position, push de nouveaux messages d'un serveur de chat à ses clients.

12. Expliquez, en `WebAssembly`, l'enchaînement des 2 promesses à utiliser suite à la commande `fetch`. (2 pts.)

`let wasmProm = fetch (ressource serveur) ;`

`instantiateStreaming (wasmProm) // Crée l'instance à partir des résultats de la promesse précédente`

`.then (results) => { traitements }`