Développer des librairies en Javascript

Nom de la vidéo

# [Num et nom de la vidéo]

## [Num et nom du module vidéo]

|  |  |
| --- | --- |
| Quelle instruction permet de créer un objet ?   * var question = {enonce=«»;reponses=[]}; * var question = {enonce:«»;reponses:[]}; * var question = {enonce=«»,reponses=[]}; * var question = {enonce:«»,reponses:[]}; | [Module 03 – séquence 02] |
| Quelle instruction permet de déclarer une classe ?   * var Question = class(enonce, reponses){   this.enonce=enonce,  this.reponses=reponses  }   * var Question = function(enonce, reponses){   this.enonce=enonce,  this.reponses=reponses  }   * var Question = function(enonce, reponses){   this.enonce=enonce;  this.reponses=reponses;  }   * var Question = class(enonce, reponses){   this.enonce=enonce;  this.reponses=reponses;  } | [Module 03 – séquence 04] |
| Quelle affirmation est correcte ?   * L’héritage est impossible en Javascript * Il n’est possible d’hériter que des attributs * Il n’est possible d’hériter que des fonctions * Il est possible d’hériter à la fois des attributs et des fonctions | [Module 03 – séquence 06] |
| Quelle(s) affirmation(s) est(sont) correcte(s) ?   * Une fonction anonyme est une fonction sans nom * Une fonction anonyme doit être exécutée au moment de sa déclaration * Une fonction anonyme peut être stockée dans une variable * Une fonction anonyme ne peut pas être exécutée au moment de sa déclaration | [Module 03 – séquence 10] |
| Quelles mécanismes permettent de déclencher des traitements asynchrones ?   * Les closures * Les callbacks * Les promesses * Le stockage local * Les fonctions anonymes | [Module 03 – séquence 16] |