Nom complet :

**Skill Test** JavaScript / 2.2

# ◆ Connaissances

JavaScript est un langage… Cochez la ou les bonnes réponses.

|  |
| --- |
| Réponse :  [X] scripté [ ] compilé [X] procédural  [ ] orienté objet [ ] orienté classe [ ] orienté prototype  [ ] typé [X] multi-thread [ ] obsolète |

JavaScript est-il un langage fonctionnel ? Donnez un exemple.

Réponse :

JavaScript est-il un langage utilisable côté serveur ? Si oui, donnez un exemple.

Réponse : oui, il peut être utiliser avec NodeJS

JavaScript est-il un langage spécifiquement lié au DOM ?

Réponse : non

Citez certaines des nouvelles fonctionnalités présentes dans l'ECMAScript 6 (ES6) :

Réponse : mots clés «let» et «const» pour définir les variables, string multilignes, promesses, classes, modules

Connaissez-vous **TypeScript** ? Expliquez son apport par rapport à du JavaScript natif.

Réponse : TypeScript est un langage open source plus pratique pour les grosses applications, car il permet de typer les variables afin d’éviter des confusions et des erreurs de données.

Citez un ou plusieurs Frameworks JavaScript actuellement utilisés dans le monde professionnel :

Réponse : Angular, Vue.js, NodeJS, (React est une «library»)

Citez un Design pattern utilisé en JavaScript et l'expliquer brièvement :

Réponse : Factory Method Pattern

# ◆ Analyse & Algorithmie

Relevez les erreurs en les soulignant :

const obj = { a = 1, b = 2, c = 3 ; } if (obj[a] == 1) { var value = true ; } else { value = false ; }

return value ? false ;

Cet exemple est syntaxiquement correct. Qu'affiche la console en l'exécutant ?

|  |
| --- |
| var a = { x : 1, y : 2, z : 3  } ;  for (var i = 0 ; i < a.length ; i++) ; console.log(i) ; |

Réponse : undefined

Cet exemple fonctionne correctement. Simplifiez la fonction 'toBoolean' suivante :

|  |
| --- |
| function toBoolean(value) { if (value) { var result = true ; } else { var result = false ; } return result ;  }  toBool(1) ; // retourne true toBool(0) ; // retourne false toBool("") ; // retourne false |

Réponse :

function toBoolean(value) {

if (value) {

var result = true ;

}

return result ;

}

Relevez les erreurs et mauvaises pratiques : Pour chaque élément relevé, notez une brève explication **ou** une solution.

var types = new Array() ;

types['grass'] = 'Bulbasaur' ;

types['fire'] = 'Charmander' ;

types['water'] = "Squirtle" ;

|  |
| --- |
| var entities = Object() ; // utiliser « let »  for (type of ['grass', 'fire', 'water']) // utiliser { }  entities[type] = {} ;  function add(type, obj = null) {  if (obj == {}) {  entities[type].push(obj) ;  } else {  return ;  }  } |
| let i = 0,  pkmns = Pokémon.getList(),  ents = new Array() ;  while (i < pkmns.length) {  switch (pkmns[i].type) {  case grass :  var p = new GrassPokemon(pkmns[i]) ;  case fire :  var p = new FirePokemon(pkmns[i]) ;  case water :  var p = new WaterPokemon(pkmns[i]) ;  }  ents.add(p) ;  i++ ;  }  return ++i ; // va avant le dernier } et pas de return |

# ◆ Prototyping

Quel est le rôle de la fonction A() dans cet exemple ?

function A(v) {

this.state = false ;

this.value = v ;

}

var a = new A(1) ;

Réponse :

Cet exemple fonctionne correctement mais comprend une mauvaise pratique. Quelle estelle ?

function A() {

this.value = 123 ;

this.getValue = function() {

return this.value ;

} ;

}

var a = new A() ;

console.log(a.getValue()) ; // 123

Réponse :

Quel est

**le rôle**

de l'objet

.prototype

en JavaScript ?

function A() {}

A.prototype = { /\* ... \*/ } ;

Réponse : permet d’hériter des propriétés d’autres objets

Complétez cet exemple pour qu'il fonctionne :

function A() { /\* ... \*/ }

// Compléter ici A.staticFn() ;

Réponse :

Qu'affiche la console en exécutant ces trois exemples ?

|  |
| --- |
| function A() {  this.value = 123 ;    setTimeout(function() { console.log(this.value) ;  }, 1000) ;  } new A() ; |

Réponse : 123

|  |
| --- |
| function A() { var value = 123 ;    setTimeout(function() { console.log(value) ;  }, 1000) ;    console.log(value++) ;  } new A() ; |

Réponse : 123 124

|  |
| --- |
| function A() {  this.value = 123 ;    setTimeout(function() { console.log(this.value) ;  }.bind(this), 1000) ;  } new A() ; |

Réponse :

|  |
| --- |
| function A() {  this.value = 123 ;    setTimeout(() => { console.log(this.value) ;  }, 1000) ;  } new A() ; |

Réponse : 123

# ◆ Bonus

Quelles librairies utilisez-vous en JavaScript ?

Réponse : jQuery, React

Expliquez la différence entre ces deux façons d'instancier un objet :

/\* 1 \*/ new A() ;

/\* 2 \*/ Object.create(A.prototype) ;

Réponse :

Creat retourne un nouvelle objet alors que new retourne le constructor de l’objet ou l’objet

Qu'affiche la console en exécutant cet exemple ?

var a, b, c ; a = !function() { b = this ; c = arguments.length ; }.call(1, 2, 3,

4) ; console.log(a, b, c) ;

Réponse :