

Nom:

Prénom:

**1. Culture web**

**/ 5**

- Que signifie l'acronyme XML ?
- Idem pour ... SQL ?
- Idem pour ... MCD ?
- Idem pour ... DOM ?
- Idem pour ... CRUD ?
- Idem pour ... NPM ?
- Idem pour ... MVC ?
- Idem pour ... JSON ?
- Idem pour ... AMP ?
- Idem pour ... PHP ?

**2. Lire le programme et répondre aux questions**

**/ 3**

```
var birds, cat, getKey, choices, goal, nb = Number( "1,3" ), x = " au soleil", y;  
getKey = function(n) { return n - 1 } ;  
birds = { color: "bleu", getNb: (n) => n } ;  
cat = { name: "kitty", age: 13, hobbies: [ "souris", "birds", "sieste", "pelote de laine" ] } ;  
y = getKey(cat.hobbies.length) ;  
choices = [ cat.hobbies[ y ], cat.hobbies[ 2 ], cat.hobbies[ 1 ], cat.hobbies[ 0 ], "" ] ;  
  
if ( cat.age < 8 ) {  
    goal = choices[ 2 ] + x;  
  
} else if ( birds.getNb(nb) ) {  
    goal = choices[ 3 ] + x;  
  
} else goal = choices[ 1 ] + x;
```

- Le chat va t'il faire sa sieste au soleil ?      oui - non
- cat.hobbies.length === choices.length ?
- Que vaut choices[ 0 ] ?

### 3. Ghost in the Shell (UNIX)

/ 2

- Commande pour cloner un repo distant git ?
- Idem pour créer un fichier foo.js ?
- Idem pour supprimer un fichier readme.txt ?
- Idem pour supprimer un dossier my-files contenant des fichiers ?
- Idem pour démarrer un programme avec nodemon (sans script) ?
- Idem pour sauvegarder sur la branch git master distante ?
- Idem pour afficher l'historique des commits ?
- Idem pour entrer dans /js contenu dans ./assets ?

### 4. Quelle est la valeur des expressions suivantes (scope global) ?

/ 2

- true || false || true
- window === "this"
- !0 || !false
- Boolean(1 || NaN)

### 5. this and that

/ 3 (barème: 0 fautes → 3, 1 faute → 2, 2 fautes → 1, plus → 0)

```
const app = ( function app() {
    "use strict";
    /* @todo : indiquer la valeur de chaque log */

    console.log(this);           //
    var that = this;

    const Bear = function Bear( n ) {
        this.name = n;
        console.log(this);       //
    };

    const foo = function foo() {
        console.log(this === that); //
    };

    const bar = function bar() {
        console.log(this);         //
        const ourson = new Bear("Winnie");
    };

    const baz = function baz() {

        const btn = document.getElementById("add_panier"); // btn !== null

        console.log(this);           //
        btn.onclick = bar;
    };

    foo();
    window.onload = baz;

})();
```

## 6. NaN ? Donnez le résultat pour:

/ 1

```
var x, y, z = 0,  
cat = { nom: "Mina", age: "8", miaou: function() { return 42 } };
```

- `isNaN( z )`
- `isNaN( cat[ "ages" ] )`
- `!isNaN( Number( false + false ) )`
- `isNaN( cat.miaou )`

## 7. Rédiger une fonction

/ 2

- Déclarez une fonction `foo` ( 0.5 )
- Prend un array `arr` en paramètre ( 0.5 )
- Lève une erreur si `arr` n'est pas un array ( 0.5 )
- Retourne la longueur de `arr` ( 0.5 )

## 8. Tourne en boucle

/ 2

Parcourir `t` avec une boucle `forEach` et afficher chacune des valeurs

( 1 )

Si la valeur courante est un array, afficher chacune de ses valeurs avec `for` ou `forEach`

( 1 )

```
var t = [ [ 1, 2, 3, 4, 5 ], "a", "ab", "abc", "abcd", [ "x", "y", "z" ], "abcde", [ ] ];
```