

PROGRAMMATION WEB
JAVASCRIPT

PROGRAMME JAVASCRIPT

- Historique
- Inclusion de JavaScript dans une page web
- Le DOM
- Méthodes d'affichage
- Variables, tableaux, objets
- Objet Math
- Conditions
- Boucles
- Fonctions et expressions de fonction
- Closures
- Écouteurs



HISTORIQUE

JAVASCRIPT

- 1995
- Brendan Eich (@BrendanEich)
- Netscape
- Java mais plus simple



«
IT'S TIME TO GET BACK TO
INNOVATING IN THE BROWSER
SPACE

– *Brendan Eich*

»

EN BREF...

JAVASCRIPT

- Principalement utilisé sur les pages web interactives
- Peut-être utilisé pour créer des serveurs (nodeJS)
- Langage orienté objet
- Monté en puissance du langage ces dernières années grâce à de nouveau frameworks (ReactJS, AngularJS, etcJS), et à la démocratisation d'applications web jusqu'à la domotique

INCLUSION DE JAVASCRIPT

JAVASCRIPT

- À l'intérieur de la page HTML grâce à la balise `<script>`
- Dans un fichier JavaScript externe `<script src="main.js">`
- Le JavaScript peut être placé dans la balise `<head>` ou dans la balise `<body>`

EXEMPLE



www.mahua.com

DOM JAVASCRIPT

Le Modèle Objet de Document, ou DOM (Document Object Model) est un outil permettant l'accès aux documents HTML et XML.

Il permet deux choses au développeur :

1. Il fournit une représentation structurée du document
2. Il codifie la manière dont un script peut accéder à cette structure

CONCEPTION DE L'ARBORESCENCE

JAVASCRIPT

Un langage de marquage comme HTML ou tout autre langage basé sur XML peut être schématisé comme une arborescence hiérarchisée.

On utilise le terme noeud, il en existe 3 différents :

1. noeud-élément
2. noeud-attribut
3. noeud-texte

ACCÈS AUX NOEUDS

JAVASCRIPT

L'objet *document* possèdent plusieurs méthodes pour sélectionner directement un ou des éléments :

- *document.getElementById('id')*
Sélectionne l'élément avec l'attribut *id* correspondant
- *document.querySelector('.class')*
Sélectionne le premier élément avec la *class* correspondante
- *document.querySelectorAll('class')*
Sélectionne tous les éléments avec la *class* correspondante

EXAMPLE



MÉTHODES D'AFFICHAGE

JAVASCRIPT

JavaScript ne possède pas de fonction d'affichage, comme *print 'test'*; en PHP par exemple.

On peut cependant utiliser des méthodes alternatives :

- *document.write('test');*
- *document.getElementById('id').innerHTML = '<p>text</p>';*
- *document.getElementById('id').innerText = 'text';*
- *console.log('test');*

EXERCICES

JAVASCRIPT

1. Afficher le résultat d'une opération mathématique à l'aide d'une des méthodes vues précédemment.
2. Concaténer 2 chaînes de caractères et afficher le résultat.



VARIABLES

JAVASCRIPT

Les variables servent à stocker des données.

```
var player = ...;
```

Il existe 3 types de variables dites « primitives » :

1. *Strings*

Chaîne de caractères, toujours insérée entre des quotes (simple ou double)

2. *Number*

Nombres positifs, négatifs... bref, tous les nombres

3. *Booleans*

`true` ou `false` on peut aussi lui passer les valeurs `1` pour `true` ou `0` pour `false`

EXAMPLE



EXERCICES

JAVASCRIPT

1. Créer 3 variables, une de chaque type vu précédemment, et afficher dans la console les valeurs de ces 3 variables, comme ci-dessous :

Name: `var1`, Score: `var2`, Armed: `var3`

2. Créer une 4e variable et additionner la avec `var2` avant d'afficher les valeurs des données dans la console.



TABLEAUX

JAVASCRIPT

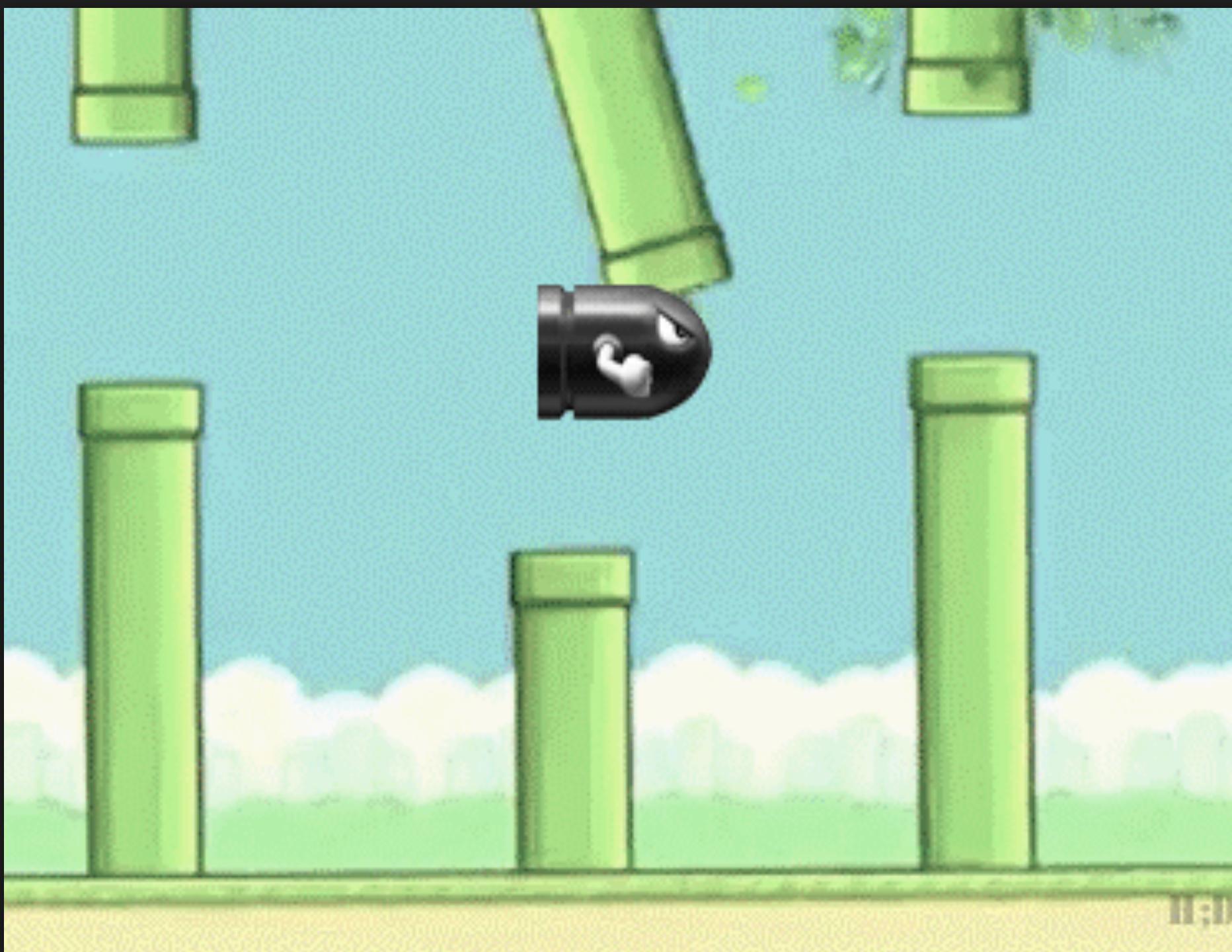
Les tableaux sont similaires aux variables, sauf qu'ils peuvent contenir plus d'une donnée.

```
var player = [ ... ];
```

Pour afficher une données précise d'un tableau nous avons besoin de connaître son **index** : un identifiant unique en relation avec la position de l'élément dans le tableau.

L'index commence à **0**, le premier élément aura pour index : **0**, le 2e : **1**, le 3e : **2**.

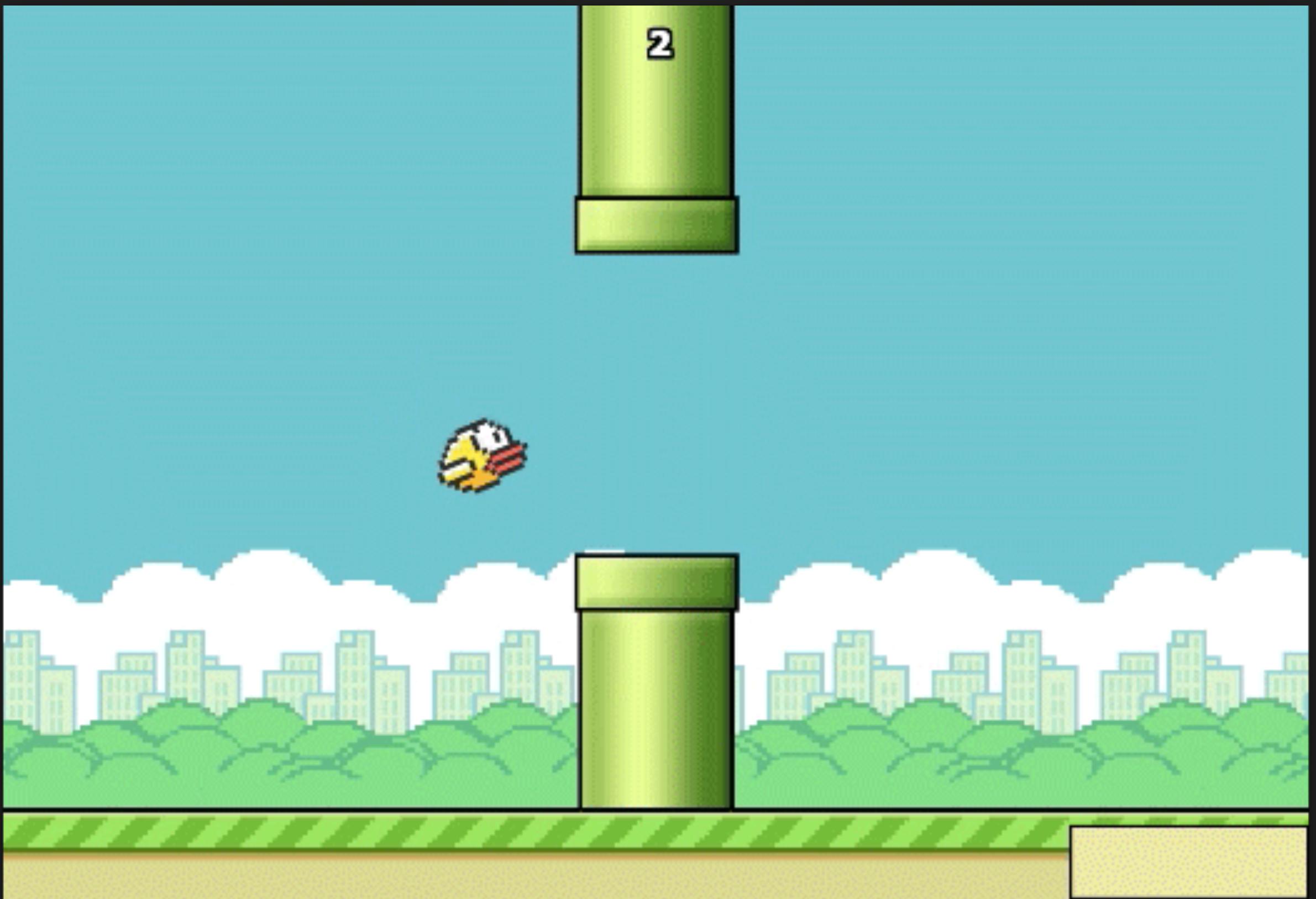
EXAMPLE



EXERCICES

JAVASCRIPT

1. Même exercice que tout à l'heure, en utilisant cette fois un tableau.
2. Utiliser la méthode *push()* propre aux tableaux pour ajouter une donnée au tableau.
3. Afficher cette nouvelle donnée dans la console à la suite des autres.



SPLICE BITCH!



EXERCICES

JAVASCRIPT

1. En utilisant la propriété *length*, que nous n'avons pas encore vu (hihi), supprimer le dernier élément du tableau.



OBJETS, MÉTHODES(), PROPRIÉTÉS



OBJETS JAVASCRIPT

Arrêtons de stocker les variables dans le **contexte global** !

```
var player = { ... };
```

Les objets sont similaires aux tableaux sauf que les données qu'on y affecte ne sont pas ordonnées : pas d'**index**.

EXAMPLE



EXERCICES

JAVASCRIPT

1. Même exercice que tout à l'heure, en utilisant cette fois un objet.
2. Stocker un tableau dans votre objet contenant 3 données, afficher la 2e donnée de ce tableau dans la console.



LEEROY JENKINS!!!



L'OBJET MATH

JAVASCRIPT

L'objet Math est un **objet natif** dont les méthodes et propriétés permettent l'utilisation de constantes et fonctions mathématiques.

Cet objet n'est pas une PUTAIN fonction !

- *Math.random();* // Retourne un nombre entre 0 (inclus) et 1 (exclus)
- *Math.ceil(Number);* // Retourne l'entier supérieur de Number
- *Math.floor(Number);* // Retourne l'entier inférieur de Number
- *Math.round(Number);* // Retourne l'entier le plus proche
- *Math.pouet(Number);* // À votre avis ??

POUET! SI JE VEUX



CONDITIONS

JAVASCRIPT

Une condition est une expression qui est évaluée à *true* ou *false*.

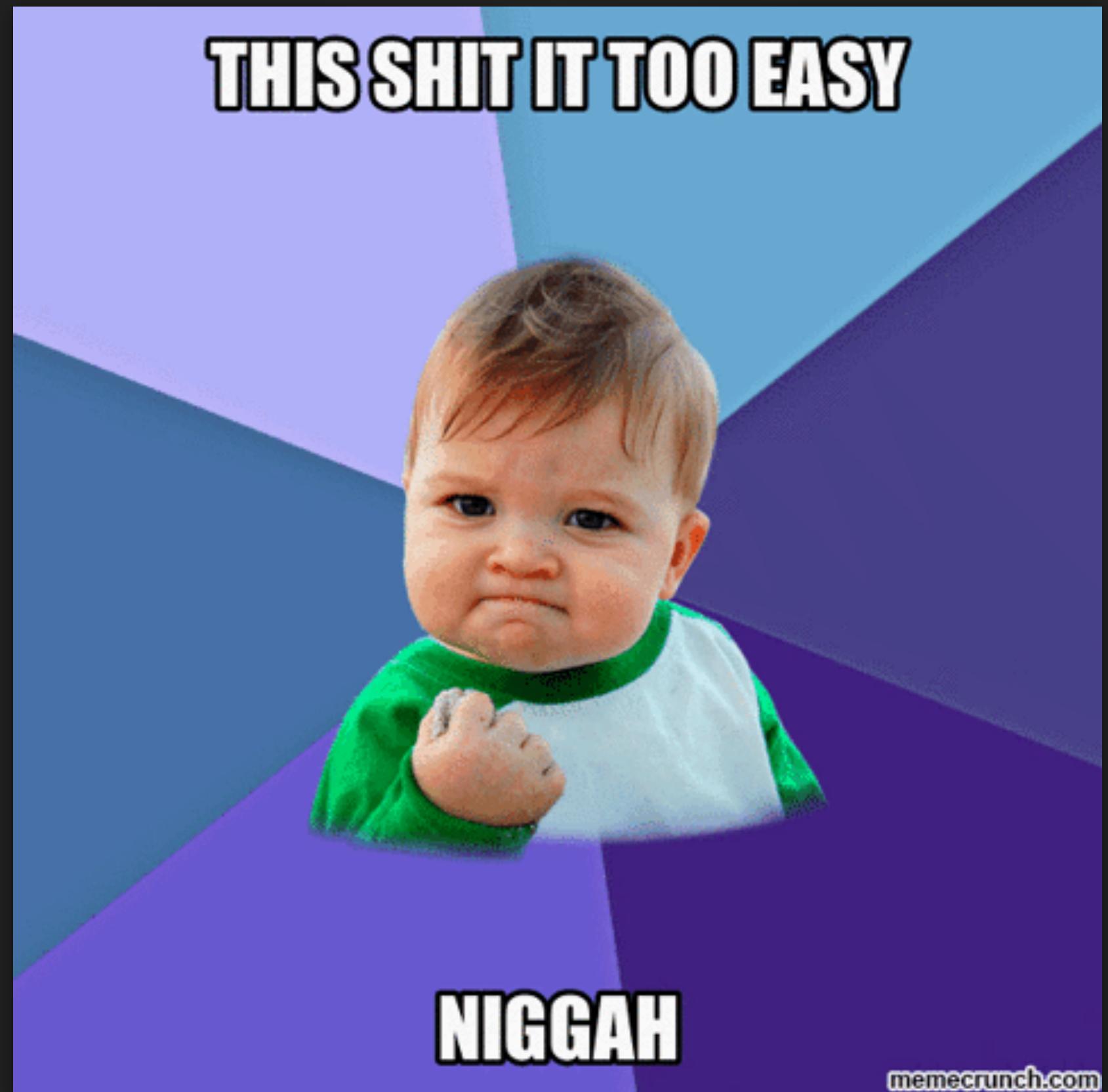
- *if (condition) { ... } else { ... }*

Cette instruction exécute une instruction si une condition donnée est vérifiée.
Si la condition n'est pas vérifiée une autre instruction pourra être exécutée.
On peut aussi utiliser un **opérateur conditionnel** (ternaire).

- *switch(expression) { ... }*

Cette instruction permet d'évaluer une expression et de faire correspondre le résultat de cette expression avec différents cas et d'exécuter les instructions associées aux cas qui ont chacun un identifiant.

EXAMPLE



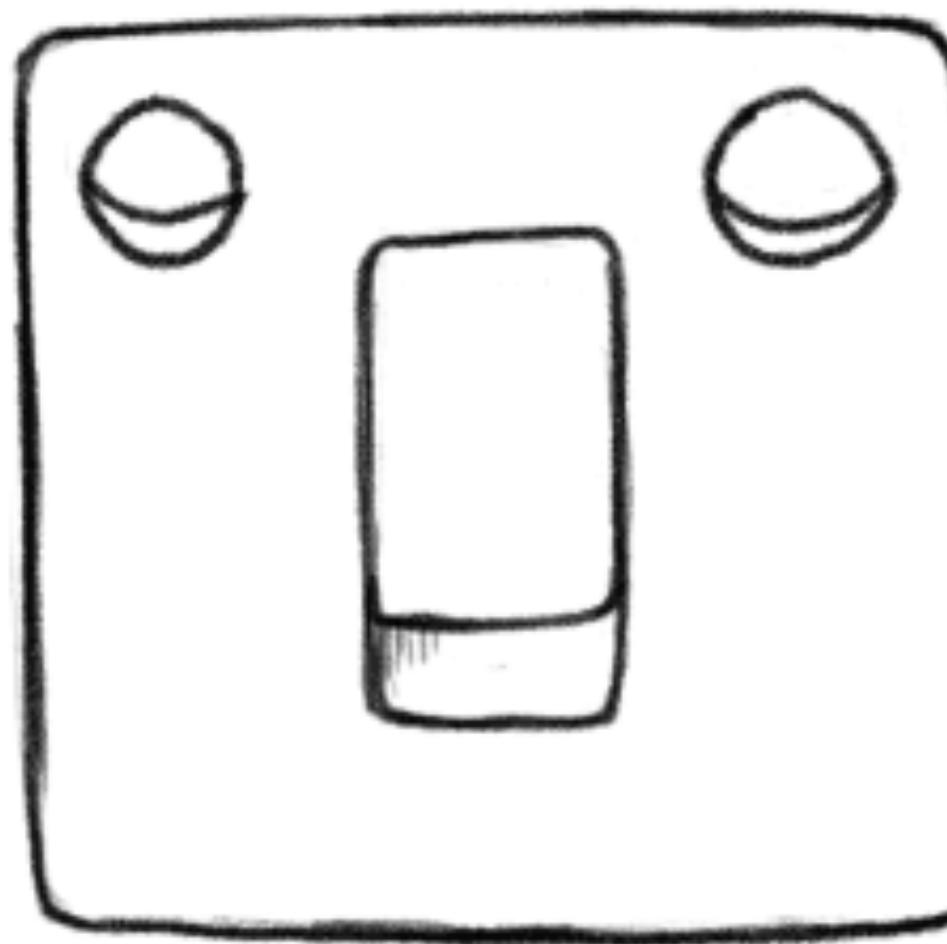
EXERCICES

JAVASCRIPT

1. Créer un **switch** le plus optimisé possible qui renvoi le score de base chaque lettre au Scrabble.

Scoring du Scrabble :

- g, d : 2 pts
- m, b, c, p : 3 pts
- y, f, v, w, h : 4 pts
- k : 5 pts
- j, x : 8 pts
- q, z : 10 pts
- autres : 1 pts



BOUCLES

JAVASCRIPT

- *for (expressions) { instruction }*

Crée une boucle qui se base sur trois expressions facultatives. Ces expressions sont entre parenthèses, séparées par des points virgules et suivies par l'instruction à exécuter dans la boucle.

- *do { instruction } while (condition)*

Crée une boucle qui exécute une instruction jusqu'à ce qu'une condition de test ne soit plus vérifiée. La condition est testée après que l'instruction soit exécutée, l'instruction définie dans la boucle est donc exécutée au moins une fois.

EXERCICES

JAVASCRIPT

1. Créer un tableau contenant 2 tableaux qui contiennent l'inventaire des joueurs (pioche, bois, rations, ...).
2. Afficher l'inventaire de chaque joueur dans la console sous forme de liste de courses.



FONCTIONS

JAVASCRIPT

Les fonctions sont des conteneurs d'instructions réutilisables !

Elles nous permettent de décider QUAND l'instruction doit être exécuté.

- *function (arguments) { instruction };*

Fonction anonyme

- *function maJolieFonction (arguments) { instruction };*

Fonction nommée

CONTEXTE, IMBRICATION, ET RÉCURSIVITÉ



EXERCICES

JAVASCRIPT

1. Créer un objet player contenant les propriétés suivantes :
name Matthieu
life 100
2. Créer une fonction récursive qui va baisser la vie de notre joueur de 1 et l'afficher dans la console à chaque boucle.
3. Afficher un message de statut de la vie du joueur quand elle atteint 75, 50, 25 et afficher un message quand le joueur est mort en reprenant son **name**.



EXPRESSIONS DE FONCTION

JAVASCRIPT

Syntaxiquement, la déclaration de fonction que nous venons de voir est une instruction.

On peut également créer une fonction grâce à une expression de fonction.

```
var maJolieFonction = function ( arguments ) { instruction };
```

Expression de fonction BIM !

MAIS CÉ KWA ?!



CLOSURE

JAVASCRIPT

Une fermeture, ou closure en anglais, est une fonction qui fait utiliser des variables indépendantes : utilisées localement mais définies dans la portée englobante.

Autrement dit, ces fonctions se « souviennent » de l'environnement dans lequel elles ont été créées (on dit aussi que la fonction capture son « environnement »).

Cache pistache, ça va la tête ?

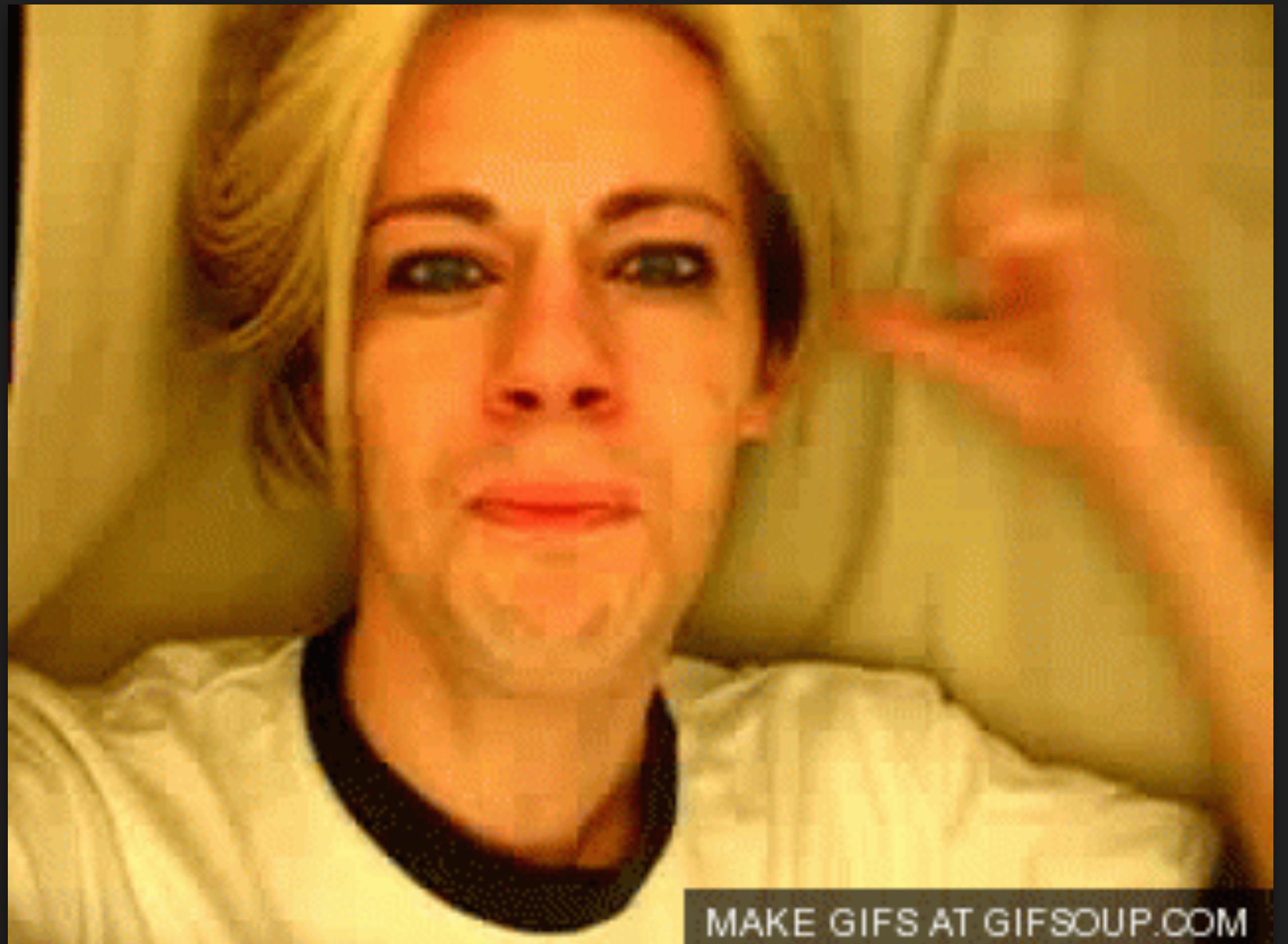
EXPLICATIONS



EXERCICES

JAVASCRIPT

1. Créer une expression de fonction nommée *stateLife*.
2. Dans cette fonction, une variable *life* à 100, et une fonction imbriquée *changeLife* avec un argument *val*.
3. L'instruction de cette fonction est : j'ajoute *val* à *life*.
4. Retourner un objet comprenant 3 fonctions : *addLife*, *removeLife* et *showLife* ; *addLife* et *removeLife* appellent *changeLife* pour ajouter ou ôter 1 points de vie ; *showLife* retourne notre *vie*.
5. Tester votre code. Bisous



MAKE GIFS AT GFSOUP.COM

ÉCOUTEURS

JAVASCRIPT

Permet l'assignation de gestionnaires d'évènements sur un élément cible.

La **cible** d'un évènement peut être un **nœud** dans un document

```
element.addEventListener(type, fonction);
```

EXEMPLE



CRÉER UNE ROULETTE RUSSE

JAVASCRIPT

1. Créer en HTML, CSS (avec ou sans frameworks) une interface très simple :
 - Titre du jeu
 - Bouton « Jouer »
 - Zone de résultats
2. Paramètres du jeu :
 - le pistolet peut contenir 7 balles
 - on met 2 balles dans le pistolet
3. À chaque clique sur « Jouer » (tir), on affiche dans la Zone de résultats le total de tir sans mourrir ou bien un message qui lui indique qu'il est mort.

