



Notre expertise est votre avenir



Initiation à JavaScript

JVS-IN

Sommaire

I.	INTRODUCTION À JAVASCRIPT	P. 3
II.	LES BASES DE JAVASCRIPT	P. 13
III.	INTERACTIONS JAVASCRIPT – HTML	P. 55
IV.	MODIFICATION DE CSS	P. 69
V.	LES OBJETS	P. 75
VI.	JQUERY	P. 81
VII.	LE DOM	P. 94
VIII.	L'OBJET WINDOW	P. 105
IX.	LES FORMULAIRES	P. 109
X.	JSON	P. 115
XI.	AJAX	P. 121
XII.	BONNES PRATIQUES DE CODE ET SÉCURITÉ	P. 129
XIII.	APERÇU DU FRAMEWORK JAVASCRIPT	P. 136
XIV.	CHOIX EXERCICES	P. 148



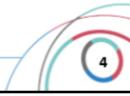
I. INTRODUCTION À JAVASCRIPT



Petite histoire

- En 1995 chez Netscape, Brendan Eich développe le LiveScript, un langage serveur.
- Netscape développe une version client : JavaScript.
- Nom donné en hommage au Java.

© m2ifformation JVS-IN





Organisation du code



```
graph LR; index["index.html"] --> style["style.css"]; index --> script["script.js"];
```

The diagram illustrates the organizational structure of a web page. At the center is the **HTML** logo with the text "index.html" below it. Two arrows point from external files to the central **HTML** icon: one from a **CSS** file icon labeled "style.css" and another from a **JS** file icon labeled "script.js".

© m2iformation JVS-IN

6

m²i
Formation

Qu'est-ce que JavaScript ?

- Langage de scripts orienté objet
- Peut modifier l'aspect d'une page Web à la volée
- Peut réagir aux actions de l'internaute

© m2ifformation JVS-IN



m²i
Formation

Hello World !

- Notre première page Web contenant du JavaScript
- Vous avez besoin de :
 - Un éditeur de texte
 - Un navigateur
- C'est tout !

© m2ifformation JVS-IN





Placer son code JavaScript

- Directement dans la page HTML entre <script></script>
- Dans un fichier externe avec le code <script src="script.js"></script>

Placer son code JavaScript

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Hello World!</title>
5  </head>
6  <body>
7      <script>
8          INSEREZ VOTRE JAVASCRIPT ICI
9      </script>
10 </body>
11 </html>
```

m²i
Formation

Première fonction

- Fonction alert(), affiche du texte

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Hello World!</title>
5  </head>
6  <body>
7      <script>
8          |     alert("Hello World");
9      </script>
10     </body>
11 </html>
```

© m2ifformation JVS-IN



11

Manip !

- Utiliser Hello World.html pour afficher Hello World !



II. LES BASES DE JAVASCRIPT

Les commentaires

- //Commentaire sur une ligne

- /*Commentaire
Sur
Plusieurs lignes */

Entrer des instructions

```
//Ces manières d'écrire fonctionnent :  
  
alert("Hello")  
alert("World")  
  
alert("Hello");alert("World")  
  
//Mais la plus classique est celle ci :  
  
alert("Hello");  
alert("World");
```

Les variables

- Un espace mémoire dans l'ordinateur
- Un nom -> une donnée
- On peut écrire, lire et modifier une variable

Les variables

```
//Déclaration de variables
var nomUtilisateur;
var motDePasse = "Azertyui";
var token1, token2;

//Affectation et modification d'une valeur de variable
nomUtilisateur = "Lou";
motDePasse = "1DKJalnaCop";

//Lecture de variable
alert(nomUtilisateur);
console.log(motDePasse);
```

m²i
Formation

Règles de nommage

- Caractères utilisables :
 - Lettres (majuscules ou minuscules)
 - Chiffres
 - Underscore « _ »
 - Dollar « \$ »
- Un nom de variable ne peut commencer par un chiffre
- Sensibles à la casse (« Nombre » et « nombre » sont des variables différentes)

© m2iformation JVS-IN



18



Bonnes pratiques de nommage

- Utiliser des noms clairs, quitte à être longs
- CamelCase
 - Commencer par une lettre minuscule
 - Chaque nouveau mot commence par une majuscule
 - Ex : nombreLettres, motDePasse
- Ou avec underscores :
 - nombre_lettres, mot_de_passe

m²i
Formation

Types de variables

- Nombres
 - 42 ; 7.8 ; 2.14e+14 ; 0x7a5
- Chaînes de Caractères
 - "JavaScript" ; "42" ; "" ; ""
- Booléens
 - True ; False
- Objets
- Fonctions

© m2ifformation JVS-IN



m²i
Formation

Types de variables

```
//Déclaration des variables
var texte = "42";
var nombre = 42;
var booleen = true;

//Vérification du type des variables (renvoie "string", "number" et "boolean")
alert(typeof texte);
alert(typeof nombre);
alert(typeof booleen);

//Conversion de variables
texte = Number(texte); //Conversion en nombre (peut retourner NaN "Not a Number")
nombre = String(nombre); //Conversion en chaîne de caractères
```

© m2ifformation JVS-IN



21



Opérateurs arithmétiques

- Addition : symbole +
- Soustraction : symbole -
- Produit : symbole *
- Division : symbole /
- Modulo (reste de la division) : symbole %

Opérateurs arithmétiques

```
//Définition de variables numériques
var x = 13, y = 5;

alert(x+y); //Retourne 18
alert(x-y); //Retourne 8
alert(x*y); //Retourne 65
alert(x/y); //Retourne 2.6
alert(x%y); //Retourne 3
```

m²i
Formation

Ajouter une valeur à une variable

- Pour ajouter 5 à une variable Nombre
 - Nombre = Nombre + 5;
- Abréviation
 - Nombre += 5;
 - -= ; *= ; /= ; %= existent aussi
- Pour ajouter 1 à une variable : le ++
 - Nombre++;

© m2iformation JVS-IN



24

Concaténation de Strings

- Concaténer : rassembler deux Strings en un nouveau
- Opérateur +

```
//Déclaration de variables
var texte1 = "Hello ", texte2 = "World";

//Nous voulons que texte3 contienne "Hello World !"
var texte3 =  texte1+texte2+" !";
```

m²i
Formation

« Petite » aide au développement

- F12 sur votre navigateur, onglet Console pour voir les potentielles erreurs retournées.
- Utiliser console.log() plutôt que alert()

© m2iformation JVS-IN



26



EXERCICES !

© m2iformation JVS-IN

27

Les conditions

- Permettent d'adapter le code
- Fonctionnent avec des tests sur les variables

La condition « If Else »

- S'écrit `if(condition) { //code si condition vraie}`
- `Else if(condition) { //code si condition vraie}`
 - Autant de « else if » que nécessaire
- `Else { // code si toutes les conditions ont échoué}`
 - //Facultatif

La condition « If Else »

```
//Nous travaillons ici sur la variable temperature  
var temperature = xxxx;  
  
//En fonction de la température nous affichons un message  
if(temperature > 20)  
{  
    | alert("Il fait chaud !");  
}  
else if(temperature > 5)  
{  
    | alert("Il ne fait pas chaud !");  
}  
else  
{  
    | alert("Il fait froid !");  
}
```

m²i
Formation

Les opérateurs de comparaison

Symbol	Signification
A == B	A égal à B
A != B	A différent de B
A > B	A supérieur à B
A >= B	A supérieur ou égal à B
A < B	A inférieur à B
A <= B	A inférieur ou égal à B
A === B	A égal et de même type que B
A !== B	A différent ou type différent de B
!(expression)	Inverse l'expression

© m2iformation JVS-IN



31

m²i
Formation

Les opérateurs de comparaison

- `42 == "42"` retourne True
- `42 === "42"` retourne False
- `5 > 5` retourne False
- `-12 <= 27` retourne True
- `"Hello" != "World"` retourne True

© m2ifformation JVS-IN



m²i
Formation

Conditions multiples

- On veut que deux conditions soient respectées
 - Un « ET », symbole : &&
- On veut qu'une condition parmi les deux conditions soit respectée
 - Un « OU », symbole : || (Alt Gr 6)

© m2ifformation JVS-IN





Conditions multiples

```
//Variables obtenues ailleurs
var temperature = ????; //Type nombre
var meteo = "????"; //type string

//Les deux cas doivent être vrais
if(temperature > 20 && meteo == "Soleil")
{
    alert("Il fait tres beau !");
}
//Un des deux cas doit être vérifié
else if(temperature > 20 || meteo == "Soleil")
{
    alert("Il fait beau ou chaud, ca va !");
}
//Sinon...
else
{
    alert("Temps pourri...");
}
```

Le Switch

- Second type de condition
- Teste une seule variable
- Sur une liste de cas précis

Le Switch

```
//Nous travaillons ici sur la variable etage
var etage = xxxx;

//En fonction de l'etage nous affichons un message
switch (etage)
{
    case 0:
        alert("Rez de chaussee !");
        break;

    case 1:
        alert("Premier etage !");
        break;

    case 2:
        alert("Deuxieme etage !");
        break;

    default :
        alert("Cet etage n'existe pas !");
}
```

Les boucles

- Permettent de répéter des instructions
- Un test détermine si la boucle doit continuer
- Le plus souvent fait évoluer une variable



La boucle « While »

- La boucle la plus simple
- S'écrit while(condition)
- Tant que sa condition est vraie, continue de tourner
- Existe aussi « Do While » qui exécute au moins une fois le code

La boucle « While »

```
//Ce script compte de 1 à 9
var number = 1;

while(number < 10)
{
    alert(number);
    number++;
}

//Ce script n'affichera rien
number = 50;

while(number < 10)
{
    alert(number);
    number++;
}
```

La boucle « Do While »

```
//Ce script compte de 1 à 9
var number = 1;

do {
    alert(number);
    number++;
}while (number < 10);

//Ce script affichera 50
number = 50;

do {
    alert(number);
    number++;
}while (number < 10);
```



La boucle « For »

- Exactement les mêmes fonctionnalités que la boucle « While »
- Mais contient toute la gestion de la variable dans une ligne
- S'écrit :
 - `for(initialisation du cpt; condition; évolution du cpt)`

La boucle « For »

```
//Ce script compte de 1 à 9
for(var number = 1;number < 10;number++)
{
    alert(number);
}

//Affichera 10
alert(number);
```

« Break; » et « Continue; »

- Opérateurs de boucles
- « Break; » fait sortir de la boucle
- « Continue; » fait passer à l'itération suivante



EXERCICES !

© m2iformation JVS-IN 44

Les fonctions

- Permettent d'exécuter une série d'instruction à plusieurs endroits du code
- Peuvent recevoir des variables en entrée : les arguments
- Peuvent renvoyer UNE valeur avec le mot-clé « return »
- Permettent d'organiser son code



Définir et utiliser des fonctions

```
//Fonction affichant "Bonjour !"
function bonjour()
{
    alert("Bonjour !");
}

//Affiche "Bonjour !"
bonjour();

//Fonction renvoyant la somme de deux variables
function somme(var1, var2)
{
    return var1 + var2;

    code qui ne sera jamais exécuté
}

var nombre1 = 14, nombre2 = 28;
var resultat = somme(nombre1, nombre2);

//Affiche 42
alert(resultat);
```



Portée des variables

- Suivant où est déclarée une variable elle n'est pas accessible aux mêmes endroits
- Variable définie hors des fonctions : globale, accessible partout
- Variable définie dans une fonction : locale, n'existe que dans la fonction
- Dans une fonction, var locale priorité sur var globale du même nom



Portée des variables

```
var globale = 42;
var valeur = "dehors";

function afficheVariables()
{
    var locale = 84;
    var valeur = "dedans";
    console.log(globale);
    console.log(locale);
    console.log(valeur);
}

afficheVariables(); //Affiche 42, 84 et dedans

console.log(globale); //Affiche 42
console.log(valeur); //Affiche dehors
console.log(locale); //Affiche une erreur
```

```
42
84
dedans
42
dehors
▶ ReferenceError: locale is not defined
```



Fonctions anonymes

- Fonctions à usage unique
- Permettent l'utilisation de variables locales
- Nécessaires dans certaines interactions avec du HTML

```
(function() {  
    ....  
})()
```

m²i
Formation

Tableaux

- Un tableau est une variable (de type Objet) qui peut stocker plusieurs données indexées avec un indice

Indice	0	1	2	3
Valeur	Valeur 1	Valeur 2	Valeur 3	Valeur 4

- Définir un tableau :
 - var monTab = [];
 - var monTab = [element1, element2];

Lire un élément du tableau :

- Alert(monTab[0]); //Affiche Valeur1

© m2ifformation JVS-IN



Création et utilisation de tableaux

```
//On remplit un tableau avec des noms d'étudiants
var tableauEtudiants = ["Guillaume", "Brigitte", "Paulin", "Manon"];

//On affiche le nombre d'étudiants
alert(tableauEtudiants.length);

//On affiche les deux premiers noms
alert(tableauEtudiants[0]);
alert(tableauEtudiants[1]);

//Brigitte est malade et est remplacée par Johanna
tableauEtudiants[1] = "Johanna";

//Le tableau a été correctement mis à jour
alert(tableauEtudiants[1]);
```

Ajouter ou retirer des éléments

- Ajouter un élément en début de tableau : tab.unshift()
- Ajouter un élément en fin de tableau : tab.push()
- Retirer le premier élément d'un tableau : tab.shift()
- Retirer le dernier élément du tableau : tab.pop()



ForEach

- Boucle automatique
- Attention, « element » contient une copie de la case lue

```
var tab = [12,20,42];

tab.forEach(function(element, index) {
    console.log("A la case "+index+" est stocke l'element : "+element);
});
```

```
A la case 0 est stocke l'element : 12
A la case 1 est stocke l'element : 20
A la case 2 est stocke l'element : 42
```



EXERCICES !

© m2iformation JVS-IN 54



III. INTERACTIONS

JAVASCRIPT - HTML

m²i
Formation

Balise HTML

- Les balises HTML régissent les pages Web
- Ce sont elles qui contiennent l'information à afficher
 - Textes
 - Liens
 - Images
- Les modifier revient à modifier la page Web

© m2ifformation JVS-IN



Balise HTML

- Une balise est d'un certain type (tag) : <div>, <p>, <body>...
- Elle doit être fermée
 - avec une balise fermante (<div>Contenu</div>)
 - ou elle s'auto-ferme (
)
- Une balise peut avoir des attributs : ID, style...
- Exemple : <div id="element">Du contenu !</div>



HTML

```
<h1 id="Titrepage">Titre de la page</h1>
<div id="Corpsdepage">
    <div id="Premierepartie">
        <p id="Paragraphe1">Ceci est le <b>premier</b> paragraphe.</p>
        
    </div>
    <div id="Secondepartie">
```

m²i
Formation

Interactions entre JavaScript et HTML

- JavaScript peut :
 - Accéder aux balises HTML
 - Récupérer leur contenu et le modifier
 - Faire que des actions de l'utilisateur activent du code

© m2ifformation JVS-IN

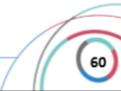


59

La fonction document.getElementById()

- Permet d'accéder à un élément HTML par son ID
- ```
var varBalise = document.getElementById("ID");
```
- Pour lire le HTML contenu dans l'élément
  - `console.log(varBalise.innerHTML)`
- Pour modifier le code HTML de l'élément
  - `varBalise.innerHTML="Nouveau code HTML";`

© m2iformation      JVS-IN



m<sup>2</sup>i  
Formation

## GetElementById et InnerHTML

```
<div id="Premierepartie">
 <p id="Paragraphe1">Ceci est le premier paragraphe.
 Il est illustré par une photo de chaton.</p>

</div>

<script>
 //Mise en variable JavaScript d'une balise HTML
 var varBalise = document.getElementById("Paragraphe1");
 //Modification du texte dans le paragraphe
 varBalise.innerHTML = "TEST";
</script>
```

Ceci est le **premier** paragraphe. Il est illustré par une photo de chaton.



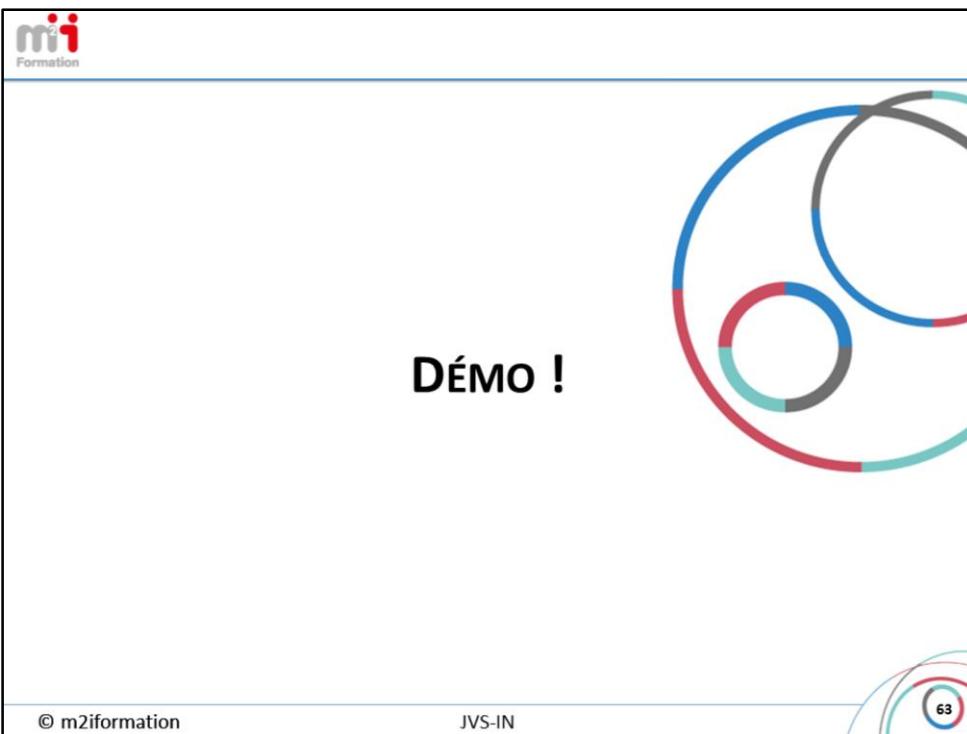
TEST



© m2iformation JVS-IN 61

## Les évènements

- Permettent de réagir en fonction des actions de l'internaute
- Peuvent déclencher du code JavaScript
- Et donc modifier du HTML



m<sup>2</sup>i  
Formation

## Evènements principaux

| Nom de l'évènement | Action le déclenchant                                                     |
|--------------------|---------------------------------------------------------------------------|
| click              | Cliquer (appuyer et relâcher le bouton gauche de la souris) sur l'élément |
| dblclick           | Double-cliquer sur l'élément                                              |
| mouseover          | Faire entrer le curseur sur l'élément                                     |
| mouseout           | Faire sortir le curseur de l'élément                                      |
| mousedown          | Appuyer sur le bouton gauche de la souris sur l'élément                   |
| mouseup            | Relâcher le bouton gauche de la souris sur l'élément                      |
| mousemove          | Déplacer le curseur sur l'élément                                         |
| change             | Modifier la valeur d'un champ de formulaire                               |
| select             | Sélectionner le contenu d'un champ de texte                               |

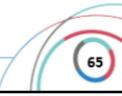
Liste complète : [https://www.w3schools.com/jsref/dom\\_obj\\_event.asp](https://www.w3schools.com/jsref/dom_obj_event.asp)

© m2ifformation JVS-IN 64



## Les évènements en DOM-0

- Manière historique d'écrire des évènements
- S'écrivent directement dans les balises HTML
- <baliseHTML évènement='fonctionAppeler();'>
- Note : en utilisant cette méthode il faut ajouter « on » au nom de l'évènement. Click devient onclick.
- Exemple : <p onclick="alert('Vous avez cliqué sur le paragraphe !');">Paragraphe</p>

© m2iformation JVS-IN 

## Les évènements en DOM-2

- Manière moderne d'écrire les évènements
- S'écrivent dans le script
- `variableBalise.addEventListener('évènement', function(){code}, false);`
- `function(){code}` est une fonction anonyme
  - « code » peut contenir n'importe quelles instructions JS

## Les évènements en DOM-2

```


</div>

<script>
//Mise en variable JavaScript d'une balise HTML
var chaton = document.getElementById("Image");
//Ajout d'une réaction au moment du clic
chaton.addEventListener("click", function(){
 alert("MIAOU !");
}, false);
</script>
```



**EXERCICES !**

© m2iformation      JVS-IN

68



## IV. MODIFICATION DE CSS



## Modifier le CSS d'un élément

- En HTML pur on va principalement agir sur le contenu des éléments
- Avec le CSS on va pouvoir réellement agir sur l'aspect visuel de notre page
- On peut par exemple agir sur :
  - La couleur des textes (color)
  - La couleur de zones (backgroundColor)
  - La position d'éléments (top, left... en position absolue)
  - La taille des éléments (height et width)

## Modifier le CSS d'un élément

- Impossible de modifier le fichier style.css avec JavaScript
- MAIS sélectionner un élément avec getElementById permet de :
  - Modifier sa classe avec .className =
  - Tous les éléments de « style » avec .style.parameterCSS =



## Modifier le CSS d'un élément

- Selectionner un élément avec getElementById :

```
varBalise = document.getElementById('element');
```

- Pour modifier un parametre CSS : varBalise.style.parametreCSS =

'nouvelle valeur';

- parametreCSS : backgroundColor, lenght, width...

- Exemple : varBalise.style.backgroundColor = 'red';

## Exemple

```
<p id="Paragraphe1">Ceci est le premier paragraphe.
Il est illustré par une photo de chaton.</p>

<script>
//Mise en variable JavaScript d'une balise HTML
var varBalise = document.getElementById("Paragraphe1");
//Modification de la couleur de fond du paragraphe
varBalise.style.backgroundColor = "red";
</script>
```

**Titre de la page**

Ceci est le premier paragraphe. Il est illustré par une pho



**Titre de la page**

Ceci est le premier paragraphe. Il est illustré par une pho



© m2ifformation JVS-IN

73



**EXERCICES !**

© m2iformation JVS-IN

74



## V. LES OBJETS



## Les classes

- En JavaScript, on peut créer des classes
- Une classe est une sorte de variable personnalisé
- On peut associer à cette classe des variables et des fonctions
  - On utilisera les termes de propriétés et de méthodes

© m2iformation JVS-IN 

## Les objets

- Un objet est une instance de classe
- Par ex : Voiture est une classe, clio2 sera un objet
- Le mot-clé « this » permet d'accéder aux éléments de l'objet

## Les objets

```
//Définition de l'objet par son constructeur
function Voiture(marque, modèle, cylindrée, année, prix)
{
 //Propriétés de l'objet Voiture
 this.marque = marque;
 this.modèle = modèle;
 this.cylindrée = cylindrée;
 this.année = année;
 this.prix = prix;

 //description() est une méthode de l'objet Voiture
 this.description = function()
 {
 alert("Il s\'agit d'une "+this.marque+" "+this.modèle+
 " avec un moteur de "+this.cylindrée+" chevaux. \n"
 "Elle a été mise en circulation en "+this.année+
 " et nous vous la proposons pour la modique somme de "+this.prix+" euros TTC !");
 }
}

//Création d'une nouvelle Voiture
var clio = new Voiture("Renault","Clio 2",75,2004,3000);

//Appel à la méthode de l'objet Voiture
clio.description();
```

### Constructeur

On y déclare les propriétés  
avec this.nom

The screenshot shows a web page with a header containing the m<sup>2</sup>i Formation logo. The main content area has a blue header bar with the text "Les objets". Below this, a modal dialog box titled "Alerte JavaScript" contains text about a Renault Clio 2 car and an "OK" button. At the bottom of the page, there are copyright information ("© m2iformation") and a footer ("JVS-IN"). A decorative graphic of overlapping colored arcs is visible on the right side.

Alerte JavaScript

Il s'agit d'une Renault Clio 2 avec un moteur de 75 chevaux.  
Elle a été mise en circulation en 2004 et nous vous la proposons pour la modique somme de 3000euros TTC !

OK

© m2iformation JVS-IN

**m<sup>2</sup>i**  
Formation

**EXERCICES !**



© m2iformation JVS-IN

80



## VI. JQUERY



jQuery

*write less, do more.*

- jQuery est une bibliothèque JavaScript permettant de développer en étant moins verbeux.
- Elle ne change rien aux fonctionnalités de JavaScript mais propose des fonctions supplémentaires très pratiques.
- Bonus : elle aide à la gestion des incompatibilités entre navigateurs (innerText() + textContent() -> text() ).

© m2iformation JVS-IN



## jQuery - Installation

- Pour « installer » jQuery :
  - La télécharger sous forme de fichier et insérer <script src='jquery.js'></script> juste avant l'appel à votre script JavaScript
  - Ou faire un appel en ligne avec
    - <script src=' http://code.jquery.com/jquery.min.js '></script> (serveurs Jquery)
    - <http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js> (Google)
    - <http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.2.min.js> (Microsoft)

© m2iformation      JVS-IN

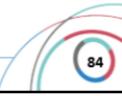


**m<sup>2</sup>i**  
Formation

## Utiliser jQuery

- Pour éviter les problèmes, exécuter le code jQuery après que le DOM ait été intégralement chargé
- `jQuery(document).ready(function() { votre code });`
- Abréviaire : `$(function() { votre code });`

© m2iformation JVS-IN



84



## Base de jQuery

- Une seule fonction : jQuery() abrégée \$()
  - Prend un ou plusieurs arguments
  - Retourne un objet jQuery qui ressemble à un tableau
    - (a un .length et permet d'accéder aux éléments avec .eq(x) )
- Exemple :
  - var balisesP = \$('p'); //sélectionne toutes les balises <p>
  - balisesP.eq(1); // sélectionne la seconde balise <p> de la page

m<sup>2</sup>i  
Formation

## Sélection avec jQuery

- Fonctionne avec des sélecteurs CSS :
  - 'x' pour toutes les balises <x>
  - '#x' pour l'élément ayant l'ID id='x'
  - '.x' pour les éléments de class class='x'
  - '[src]' pour les éléments ayant un attribut src='...'
  - Sélecteur descendant. 'div p' : tous les <p> dans un élément <div>

© m2ifformation JVS-IN 86

## Filtrer les résultats

- Mettre un argument 'bal' pour n'avoir que les éléments <bal>
- .eq(x) pour sélectionner le x+1<sup>e</sup> élément
  - Si x est un nombre négatif, sélectionne à partir de la fin
- .slice(x, y) pour sélectionner les éléments entre x+1 et y

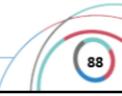
m<sup>2</sup>i  
Formation

## Dérouler la liste des éléments

- Pour dérouler tous les éléments sélectionnés par une requête :

```
$(requete).each(function(){
 alert($(this).html());
});
```
- Où \$(this) représente l'objet actuel

© m2iformation JVS-IN

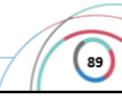




## Actions jQuery

- Une action en jQuery aura en général ce format : \$(selecteur).action(effet)
- Il est possible de chaîner les actions :  
\$(selecteur).action1(effet1).action2(effet2);
- Par ex : \$('#chap').html('Texte').css('background','red');

© m2iformation JVS-IN



m<sup>2</sup>i  
Formation

## jQuery

- Modifier du HTML :
  - Sans jQuery :
    - document.getElementById("idBalise").innerHTML="New HTML";
  - Avec jQuery :
    - \$("#idBalise").html("new HTML");
- Lire du HTML :
  - \$("#idBalise").html();
- Modifier du CSS :
  - \$("#idBalise").css('backgroundColor','red');
- Lire du CSS :
  - \$("#idBalise").css('backgroundColor');

© m2iformation JVS-IN



## jQuery

- Modifier un attribut HTML (id, src, href...)
  - `$("#idBalise").attr('attribut','nouvelleValeur');`
- Ajout d'un évènement :
  - `$("#idBalise").evenement(function(){ code });`
- Exemple :
  - `$('#bouton').click(function(){alert('Clic !');});`

## jQuery en général

- Enormément de fonctions JavaScript ont été réécrites en jQuery.
- N'hésitez pas à chercher « principales fonctions jQuery » ou « FonctionJavaScriptNatif en jQuery »



## Exercices !

- Reprenez la page d'exercices « Agir sur HTML avec JavaScript » et réalisez-la en jQuery .



## VII. LE DOM

**m<sup>2</sup>i**  
Formation

## Le DOM

- Interface de programmation permettant de parcourir des documents HTML et XML
- Définition par le W3C
- Un document HTML est représenté sous la forme d'un arbre

© m2iformation JVS-IN



## Eléments d'un arbre

- Les entités sont des **nœuds**
- Les nœuds englobés dans d'autres nœuds sont leurs **enfants**
- Les nœuds englobant d'autres nœuds sont leurs **parents**

The diagram illustrates the Document Object Model (DOM) structure. On the left, the HTML code is shown:

```
<html>
<head>
| <title>Page de DOM</title>
|</head>

<body>

| <h1>Titre de la page</h1>

| <div>
| | <p> Voici un paragraphe </p>
| |
| | <div>
| | | <p>Ici un autre paragraphe</p>
| | |
| | | <p>Et le dernier paragraphe</p>
| |
| | </div>
| |
| </div>

| </body>
</html>
```

On the right, the resulting visual representation is displayed in a box:

- Titre de la page**
- Voici un paragraphe
- Ici un **autre** paragraphe
- Et le dernier paragraphe

At the bottom left: © m2ifformation

At the bottom center: JVS-IN

At the bottom right: 97

m<sup>2</sup>i Formation

```
<html>
<head>
| <title>Page de DOM</title>
</head>

<body>
 <h1>Titre de la page</h1>

 <div>
 <p> Voici un paragraphe </p>
 <div>
 <p>Ici un autre paragraphe</p>
 <p>Et le dernier paragraphe</p>
 </div>
 </div>

</body>
</html>
```

```
graph TD
 html["<html> (ou 'document')"] --> head["<head>"]
 html --> body["<body>"]
 head --> title["<title>"]
 body --> h1["<h1>"]
 body --> div1["<div>"]
 h1 --> p1["<p>"]
 div1 --> p2["<p>"]
 div1 --> b[""]
 p2 --> p3["<p>"]
 b --> b2[""]
```

© m2ifformation JVS-IN 98

**m<sup>2</sup>i**  
Formation

## Sélection des éléments par le DOM

- Variable contient un élément HTML sélectionné avec jQuery :
  - Variable.parent(); retourne le parent de l'élément sélectionné
  - Variable.parents(); retourne les ancêtres
  - Variable.children(); retourne les enfants directs
  - Variable.find('bal'); retourne les descendants de type <bal>
  - Variable.prev(); retourne le frère précédent
  - Variable.next(); retourne le frère suivant
  - Variable.prevAll(); retourne tous les frères précédents
  - Variable.nextAll(); retourne tous les frères suivants

© m2ifformation JVS-IN





## Exemple

```
<script src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.7.2.min.js"></script>
<script>
 //On associe à une variable une balise
 var body = $("body");
 //On affiche le html de la balise <h1> parmi les enfants
 console.log(body.find("h1").html());
</script>
```

Titre de la page

**m<sup>2</sup>i**  
Formation

## Information sur les résultats

- Pour connaître le type de balise de l'élément étudié :  
`$(selec).prop('nodeName');`
- Pour connaître l'ID de l'élément étudié :  
`$(selec).prop('id');`

© m2iformation JVS-IN



**m<sup>2</sup>i**  
Formation

## Créer de nouveaux éléments à la volée

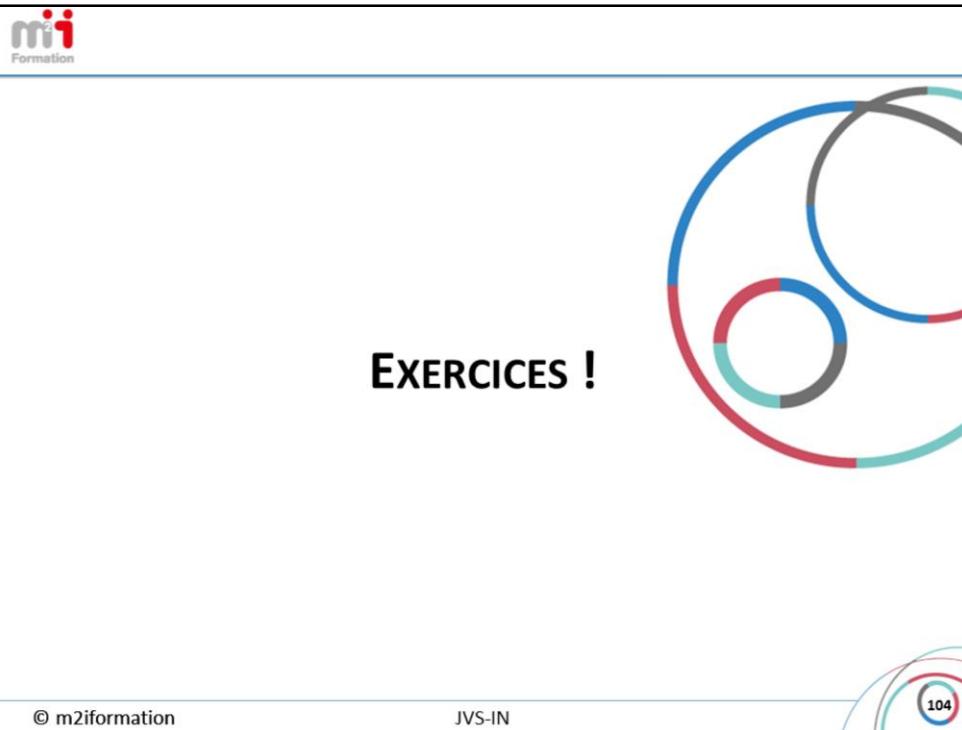
- Ajouter du contenu dans élément existant:
  - Au début : \$(selec).prepend(contenu);
  - A la fin : \$(selec).append(contenu);
- Ajouter du contenu en dehors d'un élément existant:
  - Avant : \$(selec).before(contenu);
  - Après : \$(selec).after(contenu);

© m2iformation JVS-IN



## Enlever du contenu

- `$(selec).empty();` vide un élément
- `$(selec).remove();` supprimer un élément et son contenu





## VIII. L'OBJET WINDOW

## L'objet window

- Objet correspond à la fenêtre courante
- Parent des autres objets de la page
  - Par ex : document
- alert(), prompt(), confirm() sont en fait des méthodes de l'objet window

© m2ifformation JVS-IN



106



## Sous objets de window

- window.location contient l'URL de la page actuelle
  - Utiliser window.location="URL"; charge la page URL
- window.document.links est un tableau contenant tous les liens de la page
- window.document.anchors est un tableau contenant toutes les ancre de la page

© m2ifformation JVS-IN



**m<sup>2</sup>i**  
Formation

## Méthodes de window

- Ouvrir une nouvelle fenêtre :
  - `window.open("URL","nom_de_la_fenetre","options_de_la_fenetre");`
- Fermer la fenêtre en cours :
  - `window.close();`

© m2ifformation JVS-IN





## IX. LES FORMULAIRES

**m<sup>2</sup>i**  
Formation

## Formulaires et JavaScript

- JavaScript permet de contrôler le remplissage des formulaires. Par exemple :
  - S'assurer de la cohérence des données (si vous avez entré « 0 enfants » et rempli des prénoms d'enfants => erreur)
  - Vérifier le format des données ([mail@domaine.extension](mailto:mail@domaine.extension) pour un mail)

© m2iformation JVS-IN



**m<sup>2</sup>i**  
Formation

## Champs de texte

- Pour les champs de type <input type="text"> et <textarea>
  - On récupère le contenu avec la propriété "value"
  - L'évènement onBlur permet d'agir à la sortie d'un champ de formulaire
- Exemple : <input type="text" onBlur="alert('Vous avez écrit '+this.value);"  
/>

© m2iformation JVS-IN



**m<sup>2</sup>i**  
Formation

## Les Checkbox et boutons Radio

- Pour les Checkbox, il suffit de vérifier la propriété "checked" qui contient un booléen
- Pour les boutons Radio, il faut récupérer tous les éléments (avec `getElementsByName()` par ex) et utiliser une boucle pour vérifier quel bouton est sélectionné sur l'attribut "checked"

Sexe :  Homme  Femme

Hobbies :

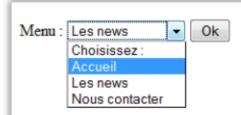
- Sport
- Lecture
- Informatique

© m2iformation JVS-IN



112

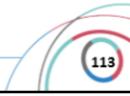
## Listes déroulantes



- Deux propriétés principales :
  - « selectedIndex » qui renvoi l'ID de la valeur sélectionnée
  - « options » qui liste dans un tableau tous les éléments <option> de la liste
- Pour récupérer l'intitulé de l'option sélectionnée il suffit donc d'écrire :

```
alert(list.options[list.selectedIndex].innerHTML);
```

où list est l'élément liste

© m2iformation JVS-IN  113

## Boutons

- À utiliser avec des évènements
- La propriété « value » retourne le nom du bouton
- `<input type="submit" />` génère un bouton d'envoi du formulaire
- `<input type="reset" />` génère un bouton de réinitialisation du formulaire



## X. JSON



## JSON

- JSON est un format de données utilisé pour transmettre des informations par requête
- JSON : JavaScript Object Notation
- Plus simple et compact que XML
- XML doit être parse avec un parseur XML alors que JSON peut être parse par une fonction JS native

© m2ifformation JVS-IN



116

## Les données en JSON

- Manière de présenter les données similaires au langage JavaScript natif :
  - Ensemble de paires 'nom' : 'valeur'
  - Les objets sont entre accolades
  - Les tableaux entre crochets
  - Les chaînes de caractères entre guillemets
  - Les nombres, booléens et null directement

m<sup>2</sup>i Formation

- XML :

```
<employees>
 <employee>
 <firstName>John</firstName> <lastName>Doe</lastName>
 </employee>
 <employee>
 <firstName>Anna</firstName> <lastName>Smith</lastName>
 </employee>
 <employee>
 <firstName>Peter</firstName> <lastName>Jones</lastName>
 </employee>
</employees>
```
- JSON :

```
{"employees": [
 {"firstName": "John", "lastName": "Doe"},
 {"firstName": "Anna", "lastName": "Smith"},
 {"firstName": "Peter", "lastName": "Jones"}]}
```

© m2ifformation JVS-IN 118

## Traitement JSON

- `JSON.stringify(var1);` pour adapter une variable en une chaîne de caractères au format JSON
- `var var1=JSON.parse(string);` pour adapter une chaîne de caractères au format JSON en une variable utilisable

m<sup>2</sup>i  
Formation

## Exemple JSON

```
function Moto(marque, modele, couleur1, couleur2)
{
 this.marque = marque;
 this.modele = modele;
 this.couleurs = [couleur1, couleur2];
}

var ninja = new Moto("Kawasaki","Ninja 300","bleue","noire");

var string = JSON.stringify(ninja);

//Affiche {"marque":"Kawasaki","modele":"Ninja 300","couleurs":["bleue","noire"]}
console.log(string);

var data = JSON.parse(string);

//Affiche Moto de type Kawasaki Ninja 300
//Couleurs : bleue et noire
console.log("Moto de type "+data.marque+" "+data.modele);
console.log("Couleurs : "+data.couleurs[0]+" et "+data.couleurs[1]);
```

© m2iformation JVS-IN 120



# XI. AJAX

**m<sup>2</sup>i**  
Formation

## Qu'est-ce qu'AJAX ?

- Asynchronous JavaScript and XML
- Permet de mettre à jour le contenu d'une page Web sans recharger la page
- Permet des requêtes serveurs alors que la page est déjà générée

© m2ifformation JVS-IN



**m<sup>2</sup>i  
Formation**

## Principe de fonctionnement

- JavaScript envoie une requête au serveur
  - Typiquement envoie des arguments à du code PHP
- Le serveur retourne des données
  - En texte, HTML, XML ou JSON
- JavaScript décode ces données et met à jour la page en fonction

© m2iformation JVS-IN



m<sup>2</sup>i Formation

## Requête Ajax

```
//Appelle le requeteur PHP et retourne sa réponse
function getData(param, argument1, argument2) {
 var XhrObj = new XMLHttpRequest();

 XhrObj.open("GET", "requeteur.php?arg1="+argument1+"&arg2="+argument2, false);

 XhrObj.setRequestHeader('Content-Type','application/x-www-form-urlencoded');
 XhrObj.send(param);

 if (XhrObj.readyState == 4 && XhrObj.status == 200)
 return XhrObj.responseText;
 else
 alert("erreur ajax :"+XhrObj.readyState+", "+XhrObj.status);
}
```

© m2iformation JVS-IN



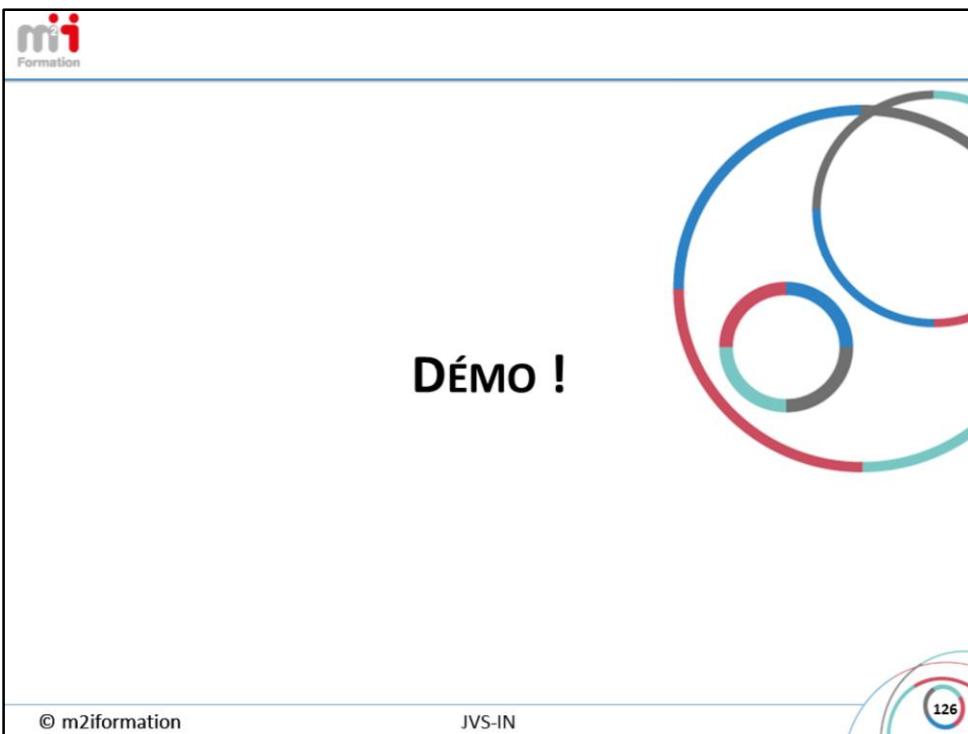
124

## Requête Ajax avec jQuery

```
$("#bouton2").click(function() {

 $.get('requeteurSimple.php?nombre=42', function(data) {
 $("#resultat").html("La reponse est : "+data);
 });

});
```



**Autres fonctions**

- `$.get()` : la plus simple, transmet par GET
- `$.post()` : similaire, transmet par POST
- `$('#bal').load()` : similaire, mais est associable directement à un élément
- `$.ajax()` : la plus complète, offre plus d'options de configuration et de gestion d'erreurs

© m2ifformation JVS-IN



m<sup>2</sup>i  
Formation

EXERCICES !



© m2iformation JVS-IN 128



## XII. BONNES PRATIQUES DE CODE ET SÉCURITÉ

## Écriture du code

- Faire fonctionner / nettoyer et optimiser / commenter
- Repenser le nommage
- Commenter chaque fonction : but, input, output

**m<sup>2</sup>i**  
Formation

## Où placer son code

- Une fonctionnalité : dans le HTML
- Un petit traitement : dans un fichier .js
- Une application JavaScript modeste : un fichier fonction, un fichier application
- RIA : un fichier par module

© m2ifformation JVS-IN



131

**m<sup>2</sup>i**  
Formation

## Lisibilité du code

- Tout code JavaScript est affiché en clair.
- Pour rendre plus difficile le rétro-engineering : Obfuscuer son code
  - UglifyJS
  - <https://javascriptobfuscator.herokuapp.com/>
- Mais aucune solution parfaite

© m2ifformation JVS-IN



**m<sup>2</sup>i  
Formation**

## Ne jamais faire confiance à ce que reçoit JavaScript

- JavaScript ne sert pas à sécuriser une page
- Il permet d'ajouter de l'interactivité ou des informations
- Ne jamais oublier que le client peut désactiver JavaScript

© m2ifformation JVS-IN

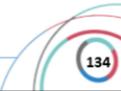


**m<sup>2</sup>i**  
Formation

## Attaques DDoS

- DDoS : Distributed Denial of Service
- De nombreux ordinateurs vont requéter le même serveur jusqu'à le saturer
- Moyen : insérer sur des sites avec des failles de sécurité un script qui sera exécuté par chaque visiteur
- Exemple : remplacer jQuery par une requête Ajax

© m2iformation JVS-IN



134



## Attaques XSS ou Spoofing

- XSS : Cross-Site Scripting
- Faire exécuter un script par un utilisateur à son insu pour récupérer des données
- Exemple : insérer un script renvoyant au serveur de l'attaquant le contenu de document.cookie



# XIII. APERÇU DU FRAMEWORK JAVASCRIPT

**m<sup>2</sup>i**  
Formation

## Bibliothèques

- jQuery et DOJO
- Redéfinissent les fonctionnalités de base
- Peuvent inclure des éléments graphiques
  - jQuery GUI pour jQuery
  - Dijit pour DOJO
- Ne modifient pas radicalement l'aspect d'un projet

© m2iformation JVS-IN



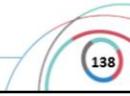
137



## JavaScript natif VS Frameworks

- Le JavaScript natif
  - Parfait pour ajouter quelques fonctionnalités
  - Vite limité pour les gros développements
- Frameworks
  - Faits pour cadrer de gros développements
  - Nécessitent un apprentissage spécifique
  - Tous les frameworks ne sont pas idéaux pour tous les projets
  - Implémentent souvent le patron d'architecture MVC

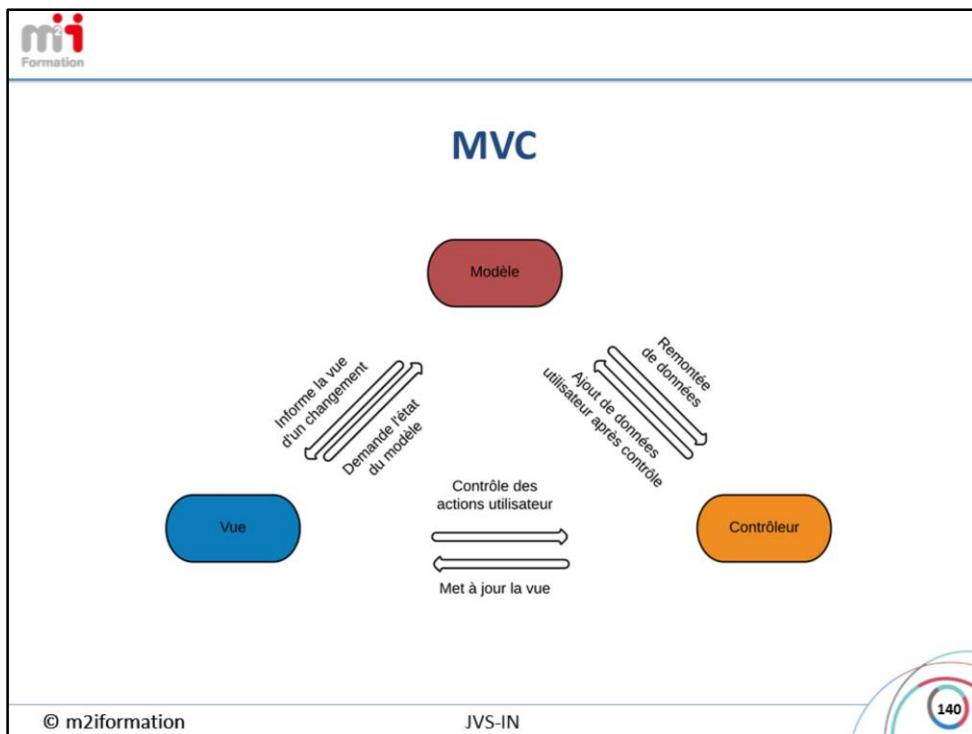
© m2iformation JVS-IN





## MVC

- Modèle - Vue - Contrôleur
- Divise les tâches → Cadre le projet
- Modèle : Gestion des données
- Vue : Interface utilisateur
- Contrôleur : Gestion des évènements







## Ember.js



- Le Framework utilisant le plus classiquement l'architecture MVC
- Recommandé pour ceux qui ont développé en MVC dans d'autres langages et pour les applications lourdes
- Framework ayant eu du mal à se stabiliser



The slide features a white background with a thin black border. In the top-left corner is the m<sup>2</sup>i Formation logo. The title "Backbone.js" is centered at the top in a dark blue sans-serif font. Below it is the Backbone.js logo, which consists of a blue stylized 'B' icon followed by the word "BACKBONE.JS" in a large, bold, dark blue sans-serif font. At the bottom left is the copyright notice "© m2iformation". At the bottom center is the code editor identifier "JVS-IN". In the bottom right corner, there is a small circular icon containing the number "143", with three curved lines extending from its top right towards the bottom right corner of the slide.



## Backbone.js



BACKBONE.JS

- Un framework léger
- N'impose pas une structure particulière mais propose des outils pour créer sa structure
- Peut servir de base pour créer son propre Framework



**Angular.js**

© m2iformation      JVS-IN

145



## Angular.js



- Un Framework boostant les capacités du HTML : recommandé aux purs développeurs Web
- Concept de base : permet d'associer des variables et des fonctions aux balises HTML
- Soutenu par Google, il est le Framework ayant la plus grosse communauté et donc le plus de contenus en ligne
- Adapté aux petits comme aux gros développements



## Quelques tutoriaux « pied à l'étrier »

- Ember.js
  - <http://vfsvp.fr/article/une-introduction-en-profondeur-a-emberjs/>
- Backbone.js
  - <http://jasongiedymin.github.io/hello-backbonejs/>
- Angular.js
  - [http://www.synbioz.com/blog/introduction\\_a\\_angularjs](http://www.synbioz.com/blog/introduction_a_angularjs)



## XIV. CHOIX EXERCICES

## Jeu de la vie

- Résumé : Créer une petite vie artificielle
- Intérêt pédagogique : Implanter de l'algorithmique en JavaScript avec un léger aspect graphique

## Mining

- Résumé : Ajouter une extension au jeu de la ferme
- Intérêt pédagogique : Ajouter de l'interaction avec des éléments graphiques



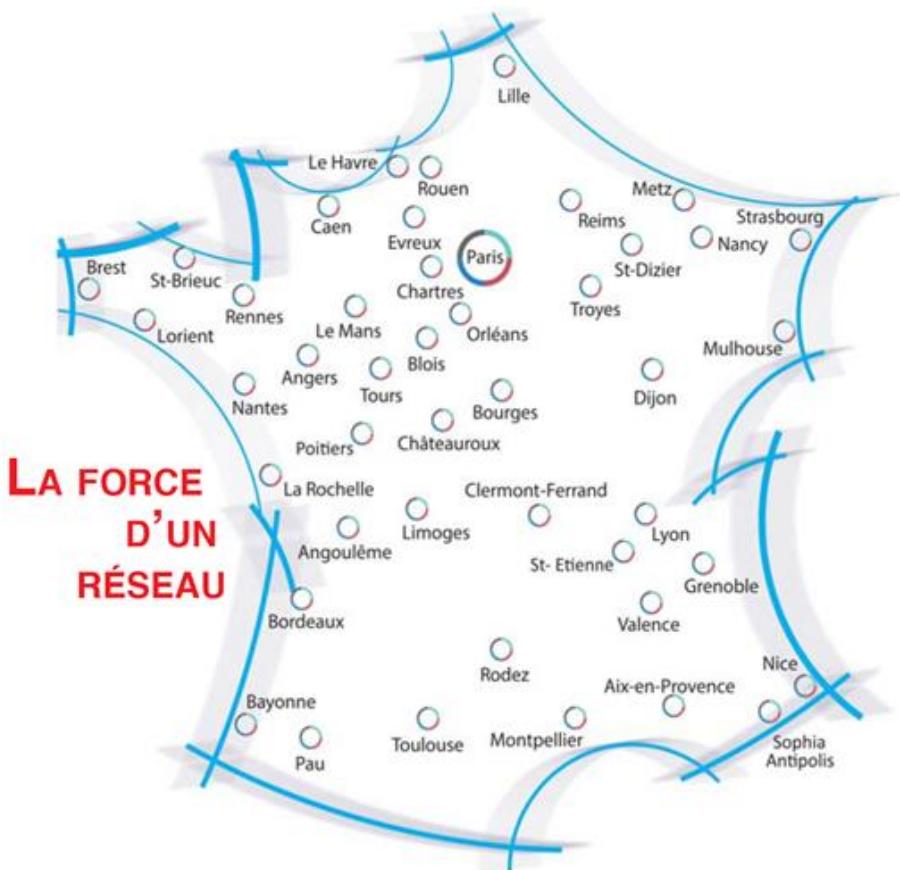
## Mini Docs

- Résumé : Créer un outil de rédaction de documents enrichis
- Intérêt pédagogique : Interactions entre éléments HTML et JavaScript.  
Utilisation d'Ajax et LocalStorage en option



## Vérification de Formulaires

- Résumé : Ajouter des contrôles JavaScript à un formulaire HTML
- Intérêt pédagogique : Interactions avec des formulaires...



► N°Azur 0 810 007 689

PRIX D'UN APPEL LOCAL DEPUIS UN POSTE FIXE

Découvrez également l'ensemble des stages à  
votre disposition sur notre site



Site web

<http://www.m2iformation.fr>