

Algorithme JavaScript

ALGORITHME JAVASCRIPT

DURE TOTALE DU MODULE : 21H00

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

Table des matières

1 Présentation du langage :	4
2 Intégration de JavaScript dans une page web :	4
3 Les variables :	5
Exercices variables :	6
4 Expression et opérateurs :	6
Exercices opérateurs :	8
5 Lecture et écriture :	8
Exercices lecture et écriture :	10
6 Les conditions :	11
Exercices conditions :	12
7 Les boucles :	13
Boucle For :	13
Boucle do while :	13
Boucle while :	14
Exercices boucles :	14
8 les Tableaux :	15
Créer un tableau vide :	15
Ajouter une valeur au tableau :	15
Ajouter une valeur à la fin du tableau :	15
Récupérer une valeur d'un tableau depuis son index (numéro de colonne) :	16
Parcourir un tableau :	16
Supprimer le dernier élément d'un tableau (dernière colonne) :	16
Ajouter un élément au début d'un tableau :	16
Supprimer un élément par son index :	16
Exercices tableaux:	17

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

1 Présentation du langage :

Un langage de programmation est une convention pour donner des ordres à un ordinateur.

JavaScript est un langage interprété et Front-end.

JavaScript va s'exécuter du côté client (dans le navigateur internet). Celui-ci va interagir avec l'interface (la vue, la page web qui va s'afficher dans le navigateur internet).

Le navigateur internet va interpréter le code contenu dans le fichier.js lié ou à l'intérieur des balises script de la page html.

Pour coder en JavaScript nous allons avoir besoin :

-un éditeur de code (Visual Studio code, Notepad++, Bracket, Sublime Text...),

-un navigateur internet (Chrome, Firefox, Edge, Opera, Safari...)

2 Intégration de JavaScript dans une page web :

Pour intégrer du code JavaScript dans une page html nous allons ajouter dans le header d'une page html le code ci-dessous :

Au sein du header de la page html nous allons appeler notre fichier.js dans l'attribut html.

```
<head>
  <script type="text/javascript" src="nom_fichier.js"></script>
</head>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

3 Les variables :

Une Variable va nous servir à :

- Stocker des valeurs,
- Résultat d'un sous-programme,
- De types différents (texte, numérique...).

Une variable est une sorte de boîte étiquetée avec un contenu.

Pour avoir accès à son contenu nous utiliserons son étiquette (son nom).

Déclaration d'une variable :

En début de programme, créer la variable avec un nom

- Peut comporter lettre et chiffre. Mais commence par une lettre,
- Pas de ponctuation, pas d'espace,
- Court.

JavaScript n'est pas un langage typé (qui va demander de définir le type d'une variable à l'instanciation). Il va déterminer automatiquement le type de variable (int, string, boolean...) à la création de celle-ci.

Exemple :

```
//déclaration et affectation d'une variable de type entier(nombre)qui a valeur 1
let a = 1;
//déclaration et affectation d'une variable de type string(chaine de
caractère) avec pour valeur toto
let b = "toto";
```

En javascript une variable va s'écrire sous la forme suivante :

```
let nom_de la variable = valeur ;
```

On peut trouver du code avec des variables écrites sous la forme :

```
var a = 1 ;
```

C'est l'ancienne façon pour créer une variable.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

Exercices variables :

Exercice 1.1 : Les Variables :

Ecrire un algorithme permettant d'échanger les valeurs de deux variables A et B.

Exercice 1.2 : Les Variables :

let A = 5 ;

let B = 2 ;

A = B ;

B = A ;

Quels seront les valeurs de A et B ?

Exercice 1.3 : Les Variables :

Ecrivez un algorithme transférant à B la valeur de A, à C la valeur de B et à A la valeur de C.

Exercice 1.4 : Les Variables :

Que produit l'algorithme suivant ?

let A = "423" ;

let B = "12" ;

let C ;

C = A + B ;

4 Expression et opérateurs :

Dans une affectation à droite du signe = nous trouvons une expression qui va être affectée à une variable.

Une expression = ensemble de valeurs, reliées par des opérateurs, et équivalent à une seule valeur

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

Exemple d'expression de type numérique :

7 ;
 5 + 4 ;
 123-45+844 ;
 Var1-12+5-Var2 ;

Attention Var1 et Var2 doivent être de type numérique

Les opérateurs : « + » et le « - »

Un opérateur = signe qui relie deux valeurs pour produire un seul résultat

Opérateurs numériques :

+ : addition
 - : soustraction
 * : multiplication
 / : Division

La puissance « ** », les parenthèses sont autorisées

Même règle qu'en mathématique : La multiplication et la division ont « naturellement » priorité sur l'addition et la soustraction

Opérateurs alphanumérique :

+ : Concaténation

Exemple :

Variables A, B, C en Caractère(string).

let A = "BON" ;

let B = "JOUR" ;

let C = A + B ;

C est égal à : BONJOUR.

Concaténation string et int :

let A = "BON" + "JOUR" + 31 ;

A est égal à : BONJOUR31.

Opérateurs logiques :

ET s'écrit avec les symboles : &&

OU s'écrit avec les symboles : ||

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

Exercices opérateurs :

Exercice 4.1 :

Que produit l'algorithme suivant ?

```
let A ="423";
let B = "12";
let C;
let C = A + B;
```

Exercice 4.2 :

Que produit l'algorithme suivant ?

```
let A = 423;
let B = "12";
let C;
let C = A + B;
```

5 Lecture et écriture :

Pour pouvoir récupérer des données depuis une page internet nous allons implémenter le code ci-dessous :

Page html :

```
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>javascript</title>
    <script type="text/javascript" src="script.js"></script>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h2>Lire une valeur dans un champs de nombre</h2>
    <form>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

```
<div>
  <label for="input1">Number :</label>
  <input type="number" id="input1" name="input1">
</div>
<input type="button" id="bouton" value="Recupérer" onclick="recup()">
</form>
</body>
</html>
```

Script javascript (à coller dans un fichier script.js à positionner au même niveau que le fichier.html) :

```
function recup() {
  //variable qui récupère le contenu de l'input1 (id="input1")
  let a = document.getElementById("input1").value;
  if(a == ""){
    console.log("des champs sont vides");
  }
  //condition si le champ est Complété
  else if (a !== NaN){
    a = parseInt(a);
  }
}
```

Pour pouvoir écrire (afficher des données) nous allons implémenter le code ci-dessous :

Affichage dans la console (script.js) :

```
console.log("valeur à afficher dans la console du navigateur internet");
```

Affichage dans la page web (page.html avec un script.js) :

```
document.write("valeur à afficher dans la page web");
```

Exemple :

Récupération de la valeur dans input1, on va et ajouter 2 à la valeur et afficher le résultat dans la console.

```
function recup() {
  //variable qui récupère le contenu de l'input1
  let a = document.getElementById("input1").value;
  if(a == ""){
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

```

    console.log("des champs sont vides");
}
//condition si le champs imput1 est rempli
else if (a !== NaN){
    a = parseInt(a);
    console.log(a+2);
}
}

```

Exercices lecture et écriture :

Exercice 5.1 : Lecture et Ecriture :

Quel résultat produit le programme suivant ?

```

let val = 231 ;
let double = val * 2 ;
console.log(val) ;
console.log(double) ;

```

Exercice 5.2 : Lecture et Ecriture :

Ecrire un programme qui demande un nombre à l'utilisateur (dans un champ de formulaire), puis qui calcule et affiche le carré de ce nombre dans la console du navigateur internet.

Exercice 5.3 : Lecture et Ecriture :

Ecrire un programme qui lit le prix HT d'un article, le nombre d'articles et le taux de TVA, et qui fournit le prix total TTC correspondant. Faire en sorte que des libellés apparaissent clairement.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

6 Les conditions :

Les tests sont ce que nous appelons les **structures alternatives**.

Ils consistent à donner des instructions différentes selon une situation.

```
If(VRAI){
    Instructions ;
}
```

```
If(VRAI){
    Instructions 1 ;
}
else{
    Instructions 2 ;
}
```

Dans le 1^{er} cas si la condition n'est pas respectée, le programme ne fait rien

Dans le 2^{ème} cas si la condition n'est pas respectée, le programme exécute l'instruction 2

Les conditions : = comparaison

Une condition est composée de 3 éléments :

- Valeur
- Opérateur de comparaison
- Une autre valeur

Les valeurs peuvent être de n'importe quel type.

Par contre deux valeurs comparées doivent avoir le même type

Opérateurs de comparaison :

- == égal à...
- != différent de...
- < Strictement plus petit que...
- > strictement plus grand que...
- <= plus petit ou égal à...
- >= plus grand ou égal à...

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

Exercices conditions :

Exercice 6.1 : Les tests :

Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on laisse de côté le cas où le nombre vaut zéro).

Exercice 6.2 : Les tests :

Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif ou positif (on laisse de côté le cas où le produit est nul). Attention toutefois : on ne doit pas calculer le produit des deux nombres.

Exercice 6.3 : Les tests :

Ecrire un algorithme qui demande trois noms à l'utilisateur et l'informe ensuite s'ils sont rangés ou non dans l'ordre alphabétique.

Exercice 6.4 : Les tests :

Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si le produit est négatif ou positif (on inclut cette fois le traitement du cas où le produit peut-être nul). Attention toutefois, on ne doit pas calculer le produit !

Exercice 6.5 : Les tests :

Ecrire un algorithme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie :

- "Poussin" de 6 à 7 ans
- "Pupille" de 8 à 9 ans
- "Minime" de 10 à 11 ans
- "Cadet" après 12 ans

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

7 Les boucles :

Les boucles permettent de répéter des actions simplement et rapidement.

Une boucle peut être vue comme une version informatique de « copier un certain nombre de lignes » ou de « faire X fois quelque chose ».

Il y a différents types de boucles mais elles se ressemblent toutes au sens où elles répètent une action un certain nombre de fois.

Les différents types de boucles permettent d'utiliser différentes façons de commencer et de terminer une boucle. Chaque type de boucle pourra être utilisé en fonction de la situation et du problème que l'on cherche à résoudre.

Boucle For :

Une boucle for répète des instructions jusqu'à ce qu'une condition donnée ne soit plus vérifiée.

Elle est composée d'un compteur (i) auquel on affecte une valeur de départ, une condition et une action qui va se passer à chaque tour de la boucle (incrément de +1).

Une boucle for s'utilise de la façon suivante :

```
for(let i=0; i<5; i++){
    console.log(i);
}
```

Boucle do while :

L'instruction do...while permet de répéter un ensemble d'instructions jusqu'à ce qu'une condition donnée ne soit plus vérifiée. (Signifie « faire... tant que »). Tant que la condition est vérifiée, l'instruction est à nouveau exécutée.

Quand la condition n'est plus vérifiée (vaut false), l'exécution de la boucle do...while est arrêtée. Une boucle do...while s'utilise de la façon suivante :

```
let i = 0;
do
{
    i += 1;
    console.log(i);
}
while (i < 5);
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

Boucle while :

Une instruction while permet d'exécuter une instruction tant qu'une condition donnée est vérifiée.

Cette boucle s'utilise de la façon suivante :

```
let i = 0;
while (i < 5)
{
    i++;
    console.log(i);
}
```

Exercices boucles :

Exercice 7.1 : Les boucles :

Récupérer une valeur dans un champ de formulaire et ajouter à chaque tour d'un boucle +1, 10 fois, afficher le résultat dans la console avec console.log() :

EX : je saisie 1-> 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 (dans la console).

Exercice 7.2 : Les boucles :

Récupérer une valeur dans un champ de formulaire et afficher dans la console la table de multiplication de cette valeur :

EX : je saisie 2 ->

2 x 1 = 2

2 x 2 = 4

...

2 x 10 = 20

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

8 les Tableaux :

Les tableaux sont des variables particulières qui vont nous permettre de stocker plusieurs valeurs au sein d'une même variable.

Un tableau va stocker des valeurs dans des colonnes, la première colonne commence à 0 puis 1 ...

Créer un tableau vide :

Pour créer un tableau vide nous allons utiliser la syntaxe suivante :

```
let tab = [] ;
```

ou

```
let tab = new Array() ;
```

Ajouter une valeur au tableau :

Pour ajouter une valeur à un tableau nous allons soit ajouter un valeur à une colonne soit l'ajouter à la fin du tableau avec la syntaxe suivante :

Ajouter une valeur à une colonne :

```
let tab = [] ;
```

Ajouter 1 dans la 1° colonne du tableau tab :

```
tab[0] = 1 ;
```

Ajouter "toto" dans la 2° colonne du tableau tab :

```
tab[1] = "toto" ;
```

Ajouter une valeur à la fin du tableau :

Pour ajouter une valeur à la fin d'un tableau nous allons utiliser la syntaxe suivante :

```
let tab = [];
```

ajouter la valeur 2 à la fin d'un tableau

```
tab.push(2);
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

Récupérer une valeur d'un tableau depuis son index (numéro de colonne) :

```
let tab = [] ;
```

```
tab[0] = 1 ;
```

```
let element = tab[0] //on récupère 1 dans element.
```

Parcourir un tableau :

Pour parcourir le contenu d'un tableau nous allons utiliser une boucle avec la syntaxe suivante :

```
tab = [1, 25, 36] //tableau avec 3 colonnes
```

```
for(i = 0 ; i < tab.length ; i++){
```

```
    console.log(tab[i]) ;
```

```
}
```

Cela va afficher dans la console 1, 2 et 36

Supprimer le dernier élément d'un tableau (dernière colonne) :

```
let tab = [1,2] ; //tableau avec 1 colonne qui contient le nombre 1 et un autre qui contient 2
```

```
tab.pop(); //cela va supprimer la dernière colonne (tab[1] qui contient 2)
```

Ajouter un élément au début d'un tableau :

```
let tab = [1];
```

```
let element= tab.unshift('toto') //ajoute toto au début du tableau colonne 0
```

Supprimer un élément par son index :

```
let tab = [1,2];
```

```
tab.splice(pos, 1) ; // supprime l'élément à l'index 1 colonne 1
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

Algorithme JavaScript

Exercices tableaux:

Exercice 8.1 tableaux :

Exercice tableau : depuis les 3 champs de formulaire compléter le tableau tab (colonne 0,1 et 2).

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		