

VBScript

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`

Plan

- 1 Introduction
- 2 IDE
- 3 Commentaires
- 4 Variable
- 5 Opérations sur les variables
- 6 Types de données
- 7 Lecture et écriture

8 Structure conditionnelle

- If ... Then
- Else
- Elseif

9 Structure itérative

- Do ... While
- Do ... Until
- For

10 Tableau

- Tableau à une dimension
- Tableau à deux dimensions

11 Fonction et procédure

- Fonction
- Procédure
- Transmission par valeur et transmission par adresse
- Récursivité

VBScript

VBScript pour Microsoft Visual Basic Scripting Edition

- Langage de script pour l'environnement **Windows**
- Créé en 1996
- Interprété (comme tout langage de script)
- Procédural et orienté objet
- Faiblement typé
- Utilisant principalement l'extension `.vbs`

BASIC, Visual Basic, VBScript vs VB.NET**BASIC :**

- Créé par **John G. Kemeny** et **Thomas E. Kurtz** en 1964.
- Conçu pour l'enseignement et aux débutants en programmation.

Visual Basic (VB) :

- Créé par **Microsoft** en 1991.
- Fournissant un **IDE** pour faciliter la création d'applications **Windows** avec des interfaces graphiques (**GUI**).
- Première version (**VB 1**) publiée en 1991, dernière version (**VB 6**) publiée en 1998. **Microsoft** a cessé de fournir un support officiel pour **VB 6** en 2008.

VBScript :

- Créé par **Microsoft** en 1996.
- Langage de script léger pour l'automatisation des tâches sous **Windows**.
- Déprécié depuis 2024 (à supprimer dans la prochaine version).

VB.NET :

- Créé par **Microsoft** et lancé avec la plateforme **.NET** en 2002.
- Offrant une version moderne et orientée objet de Visual Basic pour le développement d'applications Desktop, web, et services web.

VBScript



Microsoft Learn

<https://learn.microsoft.com> › en-us · Traduire cette page

Resources for deprecated features in the Windows client

3 juin 2024 — **VBScript** will be available as a feature on demand before being retired in future Windows releases. Initially, the **VBScript** feature on demand ...

[Msdt](#) · [Package Schema](#) · [Windows client features lifecycle](#)

VBScript

Et VBA

- **V**isual **B**asic for **A**pplications
- Version dérivée de Visual Basic
- Conçu spécifiquement pour l'automatisation des tâches dans les applications **Microsoft Office** (telles que **Excel**, **Word**, **Access...**)
- Permettant aux utilisateurs de créer des scripts à l'intérieur de ces applications sans avoir besoin d'un **IDE** externe

VBScript

Quel IDE (Environnement de développement intégré) ?

- **Visual Studio Code** (À ne pas confondre avec Visual Studio)
- Microsoft Visual Studio
- ...

© Achref EL MOU

VBScript

Quel IDE (Environnement de développement intégré) ?

- **Visual Studio Code** (À ne pas confondre avec Visual Studio)
- Microsoft Visual Studio
- ...

Visual Studio Code (ou VSC) , pourquoi ?

- Gratuit.
- Multi-système d'exploitation.
- Extensible selon le langage de programmation.

Quelques raccourcis VSC

- Pour activer la sauvegarde automatique : aller dans `File > AutoSave`
- Pour indenter son code : `Alt` `Shift` `f`
- Pour commenter/décommenter : `Ctrl` `:`
- Pour sélectionner toutes les occurrences : `Ctrl` `f2`
- Pour sélectionner l'occurrence suivante : `Ctrl` `d`
- Pour placer le curseur dans plusieurs endroits différents : `Alt`

Quelques extensions **VSC** pour **VBScript**

- **VBS**
- **VBScript**
- **Code runner**

VBScript

Commentaire

texte ignoré par le compilateur/interpréteur

© Achref EL MOUELHI

VBScript

Commentaire

texte ignoré par le compilateur/interpréteur

Syntaxe

' le texte suivant sera ignoré

VBScript

Commentaire

texte ignoré par le compilateur/interpréteur

Syntaxe

' le texte suivant sera ignoré

Ou

Rem le texte suivant sera ignoré

VBScript

Remarque

VBScript n'a pas de symbole particulier pour les commentaires multi-lignes.

VBScript

Variable

- Concept utilisé en programmation pour stocker des valeurs.
- Référence vers un espace mémoire
- Pouvant avoir les caractéristiques suivantes :
 - nom
 - type
 - valeur
 - adresse
 - portée
- Pouvant changer de valeur (mais pas de type)

VBScript

Déclaration de variables

- Il est fortement recommandé de déclarer les variables au tout début du programme.
- Une variable non déclarée peut être utilisée si l'option `Explicit` n'est pas activée.
- Pour obliger la déclaration explicite des variables, on peut utiliser l'instruction `Option Explicit` au début du script.
- La déclaration explicite des variables permet également d'éviter les erreurs typographiques (liées aux fautes de frappe dans les noms de variables) et d'identifier plus facilement les variables non initialisées.

VBScript

Règles de nommage de variables

- Les noms de variables doivent commencer par une lettre. Ils ne peuvent pas commencer par un chiffre ni par un caractère spécial.
- Les noms de variables ne peuvent contenir que des lettres (A-Z, a-z), des chiffres (0-9) et des caractères de soulignement (_). Les espaces et les caractères spéciaux ne sont pas autorisés.
- Les noms de variables ne doivent pas être des mots réservés de **VBScript** (comme **Dim**, **If**, **End...**).
- Les noms de variables ne sont pas sensibles à la casse en **VBScript**. Cela signifie que `MyVariable`, `myvariable` et `MYVARIABLE` sont considérés comme la même variable.
- Il n'y a pas de limite stricte sur la longueur des noms de variables en **VBScript**, mais il est conseillé de rester raisonnable pour assurer la lisibilité du code.
- Les noms de variables doivent être significatifs : qui reflètent leur utilisation ou le type de données qu'elles contiennent.

VBScript

Noms corrects

- `formation`
- `poe_javascript`
- `FormationJava2020`

Noms incorrects

- `206Peugeot`
- `Peugeot 206`
- `for`

VBScript

Pour déclarer une variable

```
Dim texte
```

© Achref EL MOUL

VBScript

Pour déclarer une variable

```
Dim texte
```

Pour déclarer plusieurs variables de même type

```
Dim nom, prenom
```

VBScript

Pour affecter une valeur à une variable, on utilise l'opérateur <=

```
texte = "Hello World!"
```

© Achref EL MOUL

VBScript

Pour affecter une valeur à une variable, on utilise l'opérateur <=

```
texte = "Hello World!"
```

On peut aussi affecter la valeur d'une variable à une autre

```
nom = texte
```


VBScript

On peut déclarer et initialiser une variable sur une même ligne

```
Dim texte: texte = "Hello World!"
```

© Achref EL MOUELHI ©

VBScript

On peut déclarer et initialiser une variable sur une même ligne

```
Dim texte: texte = "Hello World!"
```

On peut initialiser deux variables sur une même ligne

```
nom = "Wick": prenom = "John"
```

VBScript

On peut déclarer et initialiser une variable sur une même ligne

```
Dim texte: texte = "Hello World!"
```

On peut initialiser deux variables sur une même ligne

```
nom = "Wick": prenom = "John"
```

On peut également affecter le résultat d'une expression à une variable (concaténation)

```
texte = nom & prenom
```

VBScript

Une variable peut être utilisée sans qu'elle soit déclarée

```
x = 2
```

© Achref EL MOUELHI ©

VBScript

Une variable peut être utilisée sans qu'elle soit déclarée

```
x = 2
```

Pour interdire l'utilisation des variables non-déclarées explicitement

```
Option Explicit
```

VBScript

Une variable peut être utilisée sans qu'elle soit déclarée

```
x = 2
```

Pour interdire l'utilisation des variables non-déclarées explicitement

```
Option Explicit
```

Ainsi, l'instruction précédente génère une erreur

```
x = 2
```

VBScript

Premier programme

```
A = 3
```

```
B = 7
```

```
A = B
```

```
B = A + 5
```

```
C = A + B
```

```
C = B - 1
```

© Achret L

VBScript

Premier programme

```
A = 3  
B = 7  
A = B  
B = A + 5  
C = A + B  
C = B - 1
```

Question

Quelles sont les valeurs finales des variables A, B et C ?

VBScript

Donnez les valeurs des variables A et B après exécution des instructions suivantes

```
A = 1  
B = 2  
A = B  
B = A
```

© Achref EL M...

VBScript

Donnez les valeurs des variables A et B après exécution des instructions suivantes

```
A = 1  
B = 2  
A = B  
B = A
```

Question

Les deux dernières instructions permettent-elles d'échanger les valeurs de A et B ?

VBScript

Exercice

Écrire un programme permettant d'échanger les valeurs de deux variables A et B.

VBScript

Les expressions

- Une expression, en algorithmique, retourne toujours un résultat.
- Une expression est évaluée de gauche à droite tout en respectant les priorités.
- Types d'expression :
 - Arithmétique en utilisant des opérateurs arithmétiques
 - Logique en utilisant des opérateurs logiques ou de comparaison

VBScript

Opérateurs arithmétiques

- $+$: addition
- $*$: multiplication
- $-$: soustraction
- $/$: division
- $\backslash$: division euclidienne
- `Mod` : reste de la division
- $^$: puissance

Exemples

```
a = 10
```

```
b = 5
```

```
c = a + b
```

```
' c vaut 15
```

```
c = a - b
```

```
' c vaut 5
```

```
c = a * b
```

```
' c vaut 50
```

```
c = a / b
```

```
' c vaut 2
```

```
c = a \ b
```

```
' c vaut 2
```

```
c = a Mod b
```

```
' c vaut 0
```

```
c = a ^ b
```

```
' c vaut 100000
```

VBScript

Un expression peut contenir plusieurs opérations arithmétiques évalués de gauche à droite

```
x = 2 + 5 + 3
```

© Achref EL MOUELHI ©

VBScript

Un expression peut contenir plusieurs opérations arithmétiques évalués de gauche à droite

```
x = 2 + 5 + 3
```

Attention à la priorité, x contient la valeur 17

```
x = 2 + 5 * 3
```


VBScript

Un expression peut contenir plusieurs opérations arithmétiques évalués de gauche à droite

```
x = 2 + 5 + 3
```

Attention à la priorité, x contient la valeur 17

```
x = 2 + 5 * 3
```

On peut aussi imposer un ordre de priorité différent en utilisant les parenthèses (x contient la valeur 17)

```
x = (2 + 5) * 3
```

VBScript

Donnez les valeurs finales des variables A, B et C après exécution des instructions suivantes

```
A = 3  
B = 7  
C = B Mod A  
B = A + B * 2 - C  
A = B - A
```

VBScript

Exercice

Écrire un programme permettant d'échanger les valeurs de deux variables A et B sans utiliser de troisième variable.

VBScript

Fonctions mathématiques pouvant être utilisées en VBScript

- `Sqr (x)` : retourne la racine carrée de `x`
- `Rnd ()` : retourne un nombre aléatoire compris entre 0 et 1
- `Round (x, y)` : arrondit un nombre à un nombre spécifié de chiffres décimaux
- `Abs (x)` : retourne la valeur absolue de `x`
- `Int (x)` : retourne la partie entière de `x`
- ...

VBScript

Exemples

Abs (-5)

' Retourne 5

Sqr (25)

' Retourne 5

Int (3.1415)

' Retourne 3

Round (3.1415, 2)

' Retourne 3.14

Rnd ()

' Retourne 0,7055475

VBScript

Chaînes de caractères

- Premier caractère d'indice 1
- Entourée obligatoirement par "
- Pouvant être concaténée avec & ou +

VBScript

Quelques opérations sur les chaînes de caractères

- `Len(string)` : retourne le nombre de caractères de `string`.
- `Mid(string, start, length)` : extrait un nombre spécifié de caractères (`length`) à partir d'une position spécifiée (`start`).
- `Instr(start, string1, string2, compare)` : retourne la position de la première occurrence de `string2` dans `string1`. (`compare` accepte la valeur 0 pour sensible à la casse (par défaut) ou 1 insensible à la casse.
- `LCase(string)` : convertit une chaîne de caractères en minuscules.
- `UCase(string)` : Convertit une chaîne de caractères en majuscules.
- `Trim(string)` : Supprime les espaces au début et à la fin d'une chaîne de caractères.
- ...

VBScript

Exemples

```
msg = "Hello World!"
```

```
Len(msg)
```

```
' Retourne 12
```

```
Mid(msg, 7)
```

```
' Retourne World!
```

```
UCase(msg)
```

```
' Retourne HELLO WORLD!
```

```
LCase(msg)
```

```
' Retourne hello world!
```

```
Instr(1, msg, "o")
```

```
' Retourne 5
```


VBScript

Booléens

True et False sont respectivement représentées par les entiers -1 et 0 .

VBScript

Types de données

- **Variant** : Type de données par défaut en **VBScript**, permettant de contenir différents types de données, y compris des nombres, des chaînes de caractères, des dates...
- **Integer** : Entier de 16 bits (de -32 768 à 32 767).
- **Long** : Entier de 32 bits (de -2 147 483 648 à 2 147 483 647).
- **Single** : Nombre à virgule flottante en simple précision (32 bits).
- **Double** : Nombre à virgule flottante en double précision (64 bits).
- **Date/Time** : Représente les dates et les heures.
- **String** : Chaîne de caractères de longueur variable, pouvant contenir jusqu'à environ 2 milliards de caractères.
- **Boolean** : Valeur booléenne (True ou False).
- **Empty** : Indique qu'une variable n'a pas encore été initialisée.
- **Null** : Indique l'absence de données valides.
- ...

VBScript

Pour déterminer le type d'une variable

```
Dim x  
x = 5  
TypeName (x)  
' retourne Integer
```

© Achref EL M...

VBScript

Pour déterminer le type d'une variable

```
Dim x  
x = 5  
TypeName (x)  
' retourne Integer
```

La fonction VarType un code correspondant au type

```
VarType (x)  
' retourne 2
```

Les codes types

- 0 : Empty (variable non initialisée)
- 1 : Null (aucune donnée valide)
- 2 : Integer
- 3 : Long
- 4 : Single
- 5 : Double
- 6 : Currency
- 7 : Date
- 8 : String
- 9 : Object
- 11 : Boolean
- 17 : Byte
- 8192 : Array

VBScript

Quelques fonctions de conversion

- `CInt (expression)` : Convertit l'expression en entier.
- `CDbl (expression)` : Convertit l'expression en nombre à virgule flottante.
- `CStr (expression)` : Convertit l'expression en chaîne de caractères.
- `CBool (expression)` : Convertit l'expression en booléen.
- `CLng (expression)` : Convertit l'expression en entier long.
- `CDate (expression)` : Convertit l'expression en date.
- ...

VBScript

Quelques fonctions de vérification de types

- `IsNumeric(expression)` : Retourne True si l'expression est un nombre, sinon False.
- `IsDate(expression)` : Retourne True si l'expression est une date valide, sinon False.
- `IsArray(expression)` : Retourne True si l'expression est un tableau, sinon False.
- `IsEmpty(expression)` : Retourne True si l'expression n'a pas été initialisée, sinon False.
- `IsNull(expression)` : Retourne True si l'expression est Null, sinon False.
- ...

VBScript

Lecture et écriture (entrée/sortie)

- Les instructions de lecture/écriture permettent à la machine de communiquer avec l'utilisateur.
- L'entrée standard est le clavier et la sortie standard est l'écran.
- L'opération de lecture est bloquante : L'exécution s'arrête lorsqu'on rencontre une instruction de lecture et ne se poursuit qu'après saisie d'une valeur.
- Avant de lire une variable, il est conseillé d'écrire un message à l'écran afin de prévenir l'utilisateur de ce qu'il doit saisir.

VBScript

Pour afficher un message dans un `popup` de Microsoft

```
MsgBox "Hello World!"
```

© Achref EL MOUELH

VBScript

Pour afficher un message dans un `popup` de Microsoft

```
MsgBox "Hello World!"
```

Pour afficher le contenu d'une variable dans un `popup` de Microsoft

```
msg = "Hello World!"  
MsgBox msg
```

VBScript

Pour afficher un message dans le terminal

```
WScript.Echo "Hello World!"
```

© Achref EL MOUËL

VBScript

Pour afficher un message dans le terminal

```
WScript.Echo "Hello World!"
```

Pour afficher le contenu d'une variable dans le terminal

```
msg = "Hello World!"  
WScript.Echo msg
```

VBScript

Pour lire une valeur saisie par l'utilisateur

```
nom = InputBox("Entrez votre nom : ", "Demande de saisie")  
MsgBox "Votre nom est " & nom
```

© Achref EL MOUELHI

VBScript

Pour lire une valeur saisie par l'utilisateur

```
nom = InputBox("Entrez votre nom : ", "Demande de saisie")  
MsgBox "Votre nom est " & nom
```

Explication

- InputBox affiche un popup avec
 - le message Entrez votre nom :
 - le titre Demande de saisie.
- La valeur saisie par l'utilisateur est stockée dans la variable nom.

VBScript

Exercice

Écrire un programme qui demande un nombre à l'utilisateur, puis calcule et affiche son double.

VBScript

Exercice

Écrire un programme qui demande à l'utilisateur de saisir son nom puis son prénom et qui récupère et affiche ensuite les initiales stockées dans une chaîne de caractères.

Structure conditionnelle

- Les structures conditionnelles servent à n'exécuter une instruction ou une séquence d'instructions que si une condition est vraie.
- Une **condition** doit toujours retourner une valeur booléenne (`vrai` ou `faux`).

VBScript

Exemple

```
Dim x: x = 2
If x > 0 Then
    MsgBox "positif"
End if
```

© Achret L.

VBScript

Exemple

```
Dim x: x = 2
If x > 0 Then
    MsgBox "positif"
End if
```

Pour les conditions, on utilise les opérateurs de comparaison.

VBScript

Opérateurs de comparaison

- =
- <>
- >
- <
- >=
- <=

VBScript

Exercice

Écrire un programme qui demande à l'utilisateur de saisir un nombre entier, puis affiche s'il s'agit d'un nombre pair.

VBScript

Exécuter un premier bloc si une condition est vraie, un deuxième sinon

```
Dim x: x = -2
If x > 0 Then
    MsgBox "positif"
Else
    MsgBox "négatif"
End if
```

VBScript

Exercice

Modifier le programme précédent pour afficher la parité du nombre saisi.

VBSript

On peut enchaîner les conditions avec `sinon si` (et avoir un troisième bloc voire ... un nième)

```
Dim x: x = -2
If x > 0 Then
    MsgBox "positif"
ElseIf x = 0 Then
    MsgBox "nul"
Else
    MsgBox "négatif"
End if
```


VBScript

Exercice

Écrire un programme qui permet de résoudre une équation de second degré ($ax^2 + bx + c = 0$). L'utilisateur saisit des valeurs pour a , b et c et on lui affiche la ou les solutions s'il en existe.

VBScript

Pour enchaîner les conditions, on utilise les opérateurs logiques

- And
- Or
- Not
- Xor (or exclusif)

VBScript

Exemple

```
WScript.echo (a > b) And (b > 0)
```

```
' affiche -1
```

```
WScript.echo (a > b) Or (b < 0)
```

```
' affiche -1
```

```
WScript.echo (a > b) Xor (b > 0)
```

```
' affiche 0
```

```
WScript.echo Not (a > b)
```

```
' affiche 0
```

VBScript

Exercice

Écrire un programme qui demande à l'utilisateur de saisir deux nombres et qui affiche le signe du résultat de la multiplication sans la calculer.

VBScript

Exercice

Écrire un programme qui demande à l'utilisateur de saisir l'indice d'un mois (un nombre compris entre 1 et 12) et qui affiche le nombre de jours du mois associé.

VBScript

Exercice

Écrire un programme qui demande à l'utilisateur de saisir trois notes et qui calcule puis affiche la moyenne.

VBScript

Une solution possible

```
Dim note1, note2, note3, moyenne

note1 = InputBox("Note", "Première note")
note2 = InputBox("Note", "Deuxième note")
note3 = InputBox("Note", "Troisième note")

moyenne = (CSng(note1) + CSng(note2) + CSng(note3)) / 3

MsgBox moyenne
```

VBScript

Question

Et si on voulait calculer la moyenne de plusieurs notes (des dizaines voire des centaines), faudrait-il répéter écrire et lire autant de fois que nécessaire ?

© Achref EL

VBScript

Question

Et si on voulait calculer la moyenne de plusieurs notes (des dizaines voire des centaines), faudrait-il répéter écrire et lire autant de fois que nécessaire ?

Réponse

Non, on peut utiliser les **boucles**.

VBScript

Structure itérative (boucle)

- Les structures itératives servent à répéter l'exécution d'un bloc d'instructions un certain nombre de fois.
- Trois types de boucles proposées :
 - **Do ... While** : on répète des instructions tant qu'une certaine condition est vraie.
 - **Do ... Until** : on répète des instructions jusqu'à ce qu'une certaine condition soit vraie.
 - **For** : on répète des instructions ou avec compteur en faisant évoluer un compteur entre une valeur initiale et une valeur finale.

VBScript

Boucle Do ... While : à chaque itération on teste si la condition est vraie avant d'accéder aux traitements

```
Do While condition  
    ' Traitements  
Loop
```

© Achref EL M...

VBScript

Boucle Do ... While : à chaque itération on teste si la condition est vraie avant d'accéder aux traitements

```
Do While condition  
    ' Traitements  
Loop
```

Remarque

Attention aux boucles infinies, vérifier que la condition d'arrêt sera bien atteinte après un certain nombre d'itérations.

VBScript

Exemple

```
Dim i
i = 1
Do While i <= 5
    WScript.Echo i
    i = i + 1
Loop
```

© Achref EL M...

VBScript

Exemple

```
Dim i
i = 1
Do While i <= 5
    WScript.Echo i
    i = i + 1
Loop
```

Le résultat est

```
1
2
3
4
5
```

VBScript

Do ... While permet aussi d'exécuter le bloc de code au moins une fois, puis continue tant que la condition est vraie.

```
Dim i
i = 1
Do
    WScript.Echo i
    i = i + 1
Loop While i <= 5
```

© Achref EL

VBScript

Do ... While permet aussi d'exécuter le bloc de code au moins une fois, puis continue tant que la condition est vraie.

```
Dim i
i = 1
Do
    WScript.Echo i
    i = i + 1
Loop While i <= 5
```

Le résultat est

```
1
2
3
4
5
```


VBScript

Exercice

En utilisant une boucle `Do ... While`, réécrire l'algorithme précédent pour permettre à l'utilisateur de saisir trois notes et lui calculer puis afficher la moyenne.

VBScript

La Boucle Do ... Until exécute le bloc au moins une fois ensuite elle vérifie la condition

```
Do  
    ' Traitements  
Loop While condition
```

© Achref EL ME

VBScript

La Boucle Do ... Until exécute le bloc au moins une fois ensuite elle vérifie la condition

```
Do  
    ' Traitements  
Loop While condition
```

Remarque

Attention aux boucles infinies, vérifier que la condition d'arrêt sera bien atteinte après un certain nombre d'itérations.

VBScript

Exemple

```
Dim i
i = 1
Do
    WScript.Echo i
    i = i + 1
Loop While i <= 5
```

© Achref EL M...

VBScript

Exemple

```
Dim i
i = 1
Do
    WScript.Echo i
    i = i + 1
Loop While i <= 5
```

Le résultat est

```
1
2
3
4
5
```

VBScript

Do ... Until permet d'exécuter le bloc de code au moins une fois, puis continue jusqu'à ce que la condition soit vraie

```
Dim i
i = 1
Do
    WScript.Echo i
    i = i + 1
Loop Until i > 5
```

© Achref EL

VBScript

Do ... Until permet d'exécuter le bloc de code au moins une fois, puis continue jusqu'à ce que la condition soit vraie

```
Dim i
i = 1
Do
    WScript.Echo i
    i = i + 1
Loop Until i > 5
```

Le résultat est

```
1
2
3
4
5
```

VBScript

Boucle pour

```
For i = 1 To 5  
    ' Traitements  
Next
```

© Achref EL MOU

VBScript

Boucle pour

```
For i = 1 To 5  
    ' Traitements  
Next
```

Remarque

Attention aux boucles infinies si vous modifiez la valeur du compteur à l'intérieur de la boucle.

VBScript

Exemple (par défaut le pas = 1)

```
Dim i
i = 1

For i = 1 To 5
    WScript.Echo i
Next
```

© Achref EL

VBScript

Exemple (par défaut le pas = 1)

```
Dim i  
i = 1  
  
For i = 1 To 5  
    WScript.Echo i  
Next
```

Le résultat est

```
1  
2  
3  
4  
5
```

VBScript

Exemple avec un pas différent de 1

```
Dim i
i = 1

For i = 1 To 10 Step 2
    WScript.Echo i
Next
```

© Achref EL

VBScript

Exemple avec un pas différent de 1

```
Dim i  
i = 1  
  
For i = 1 To 10 Step 2  
    WScript.Echo i  
Next
```

Le résultat est

```
1  
3  
5  
7  
9
```

VBScript

Exercice

Écrire un programme qui permet à l'utilisateur de saisir deux entiers x et y et qui calcule, en utilisant une boucle `For`, puis affiche x^y .

Exercice

Écrire un programme qui demande à l'utilisateur de saisir une suite de valeurs entières différentes de 0. À la fin, l'algorithme affiche le nombre de saisies positives.

VBScript

Problème

Chaque fois que l'utilisateur saisissait une valeur la précédente est perdue (remplacée par la nouvelle).

© Achref EL M...

VBScript

Problème

Chaque fois que l'utilisateur saisissait une valeur la précédente est perdue (remplacée par la nouvelle).

Solution

Utiliser les tableaux.

VBScript

Deux types de tableau en **VBScript**

- Un tableau statique est un tableau dont la taille est spécifiée lors de la déclaration et ne peut pas être modifiée par la suite.
- Un tableau dynamique est déclaré sans taille initiale et peut être redimensionné à l'aide de `ReDim`.

VBScript

Pour créer un tableau statique de 5 éléments

```
Dim arr(4)
arr(0) = "A"
arr(1) = "B"
arr(2) = "C"
arr(3) = "D"
arr(4) = "E"
```

© Achre

VBScript

Pour créer un tableau statique de 5 éléments

```
Dim arr(4)
arr(0) = "A"
arr(1) = "B"
arr(2) = "C"
arr(3) = "D"
arr(4) = "E"
```

Ou

```
Dim arr
arr = Array("A", "B", "C", "D", "E")
```

VBScript

Pour créer un tableau dynamique de 5 éléments

```
Dim arr()  
ReDim arr(4)  
arr(0) = "A"  
arr(1) = "B"  
arr(2) = "C"  
arr(3) = "D"  
arr(4) = "E"
```

VBScript

Pour déterminer la taille d'un tableau

```
taille = UBound(arr) - LBound(arr) + 1  
WScript.Echo taille
```

© Achref EL MOUELHI

VBScript

Pour déterminer la taille d'un tableau

```
taille = UBound(arr) - LBound(arr) + 1  
WScript.Echo taille
```

Explication

- LBound et UBound renvoient respectivement le plus petit et le plus grand indice d'un tableau.
- LBound retourne toujours 0 pour les tableaux standard, car les tableaux en **VBScript** sont toujours indexés à partir de 0.

VBScript

Pour parcourir un tableau

```
For Each item In arr  
    WScript.Echo item  
Next
```

© Achref EL MOUËL

VBScript

Pour parcourir un tableau

```
For Each item In arr  
    WScript.Echo item  
Next
```

Résultat

A
B
C
D
E

VBScript

Ou

```
Dim i
For i = 0 To UBound(arr)
    MsgBox arr(i)
Next
```

© Achref EL MOU

VBScript

Ou

```
Dim i
For i = 0 To UBound(arr)
    MsgBox arr(i)
Next
```

Résultat

A
B
C
D
E

Pour ajouter des nouveaux éléments dans un tableau dynamique, il faut le re-dimensionner

```
Dim arr()  
ReDim arr(4)  
arr(0) = "A"  
arr(1) = "B"  
arr(2) = "C"  
arr(3) = "D"  
arr(4) = "E"  
  
' Preserve permet de préserver les éléments déjà présents dans le tableau  
ReDim Preserve arr(6)  
arr(5) = "F"  
arr(6) = "G"  
  
taille = UBound(arr) - LBound(arr) + 1  
WScript.Echo taille  
  
For Each item In arr  
    WScript.Echo item  
Next
```

Pour ajouter des nouveaux éléments dans un tableau dynamique, il faut le re-dimensionner

```
Dim arr()  
ReDim arr(4)  
arr(0) = "A"  
arr(1) = "B"  
arr(2) = "C"  
arr(3) = "D"  
arr(4) = "E"  
  
' Preserve permet de préserver les éléments déjà présents dans le tableau  
ReDim Preserve arr(6)  
arr(5) = "F"  
arr(6) = "G"  
  
taille = UBound(arr) - LBound(arr) + 1  
WScript.Echo taille  
  
For Each item In arr  
    WScript.Echo item  
Next
```

Résultat

```
7  
A  
B  
C  
D  
E  
F  
G
```

VBScript

Exercice

Écrire un programme qui

- 1 permet de remplir un tableau de 5 éléments avec des valeurs saisies par l'utilisateur.
- 2 affiche tous les éléments.
- 3 calcule et affiche la somme de tous les éléments.

VBScript

Exercice

Écrire un programme qui

- 1 permet de remplir un tableau de 5 éléments avec des valeurs saisies par l'utilisateur.
- 2 permet à l'utilisateur de saisir une valeur à chercher dans le tableau.
- 3 affiche si la valeur saisie existe dans le tableau.

VBScript

Exercice

Écrire un programme qui

- 1 permet de remplir un tableau de 5 éléments avec des valeurs saisies par l'utilisateur.
- 2 cherche et affiche la plus grande valeur du tableau (max).

VBScript

Tableau à deux dimensions

- Équivalent d'une matrice en mathématiques.
- Tableaux avec des valeurs repérées par deux indices.
 - Premier indice : pour les lignes
 - Deuxième indice : pour les colonnes.

VBScript

Pour déclarer un tableau à deux dimensions (3x3)

```
Dim matrix(2, 2)
matrix(0, 0) = 1
matrix(0, 1) = 2
matrix(0, 2) = 3
matrix(1, 0) = 4
matrix(1, 1) = 5
matrix(1, 2) = 6
matrix(2, 0) = 7
matrix(2, 1) = 8
matrix(2, 2) = 9
```

VBScript

Pour déclarer un tableau à deux dimensions (3x3)

```
Dim matrix(2, 2)
matrix(0, 0) = 1
matrix(0, 1) = 2
matrix(0, 2) = 3
matrix(1, 0) = 4
matrix(1, 1) = 5
matrix(1, 2) = 6
matrix(2, 0) = 7
matrix(2, 1) = 8
matrix(2, 2) = 9
```

Pour accéder à un élément selon ses indices

```
MsgBox matrix(1, 1)
' Affiche 5
```

VBScript

Pour déterminer le nombre de lignes et le nombre de colonnes d'un tableau à deux dimensions (3x3)

```
Dim numRows, numCols

' Calculer le nombre de lignes
numRows = UBound(matrix, 1) - LBound(matrix, 1) + 1

' Calculer le nombre de colonnes
numCols = UBound(matrix, 2) - LBound(matrix, 2) + 1

' Afficher les résultats
WScript.Echo "Nombre de lignes : " & numRows
WScript.Echo "Nombre de colonnes : " & numCols
```

VBScript

Exercice

Écrire un programme qui

- 1 permet de remplir une matrice carrée (nombre de lignes = nombre de colonnes) de taille 3.
- 2 affiche tous les éléments.
- 3 calcule et affiche la somme de tous les éléments.

VBScript

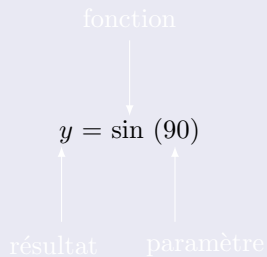
Exercice

Écrire un programme qui permet de

- 1 multiplier une matrice A par une constante α .
- 2 calculer la somme de deux matrices carrées A et B .
- 3 vérifier si une matrice A est diagonale.
- 4 calculer le produit de deux matrices carrées A et B .

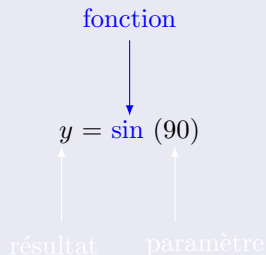
VBScript

En mathématiques



VBScript

En mathématiques

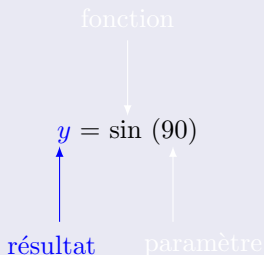


Remarque

En algorithmique aussi on peut utiliser et écrire des fonctions.

VBScript

En mathématiques



Remarque

En algorithmique aussi on peut utiliser et écrire des fonctions.

VBScript

En mathématiques

fonction

$y = \sin(90)$

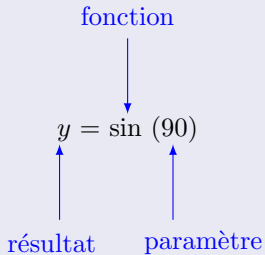
résultat paramètre

Remarque

En algorithmique aussi on peut utiliser et écrire des fonctions.

VBScript

En mathématiques



Remarque

En algorithmique aussi on peut utiliser et écrire des fonctions.

VBScript

Les fonctions en **VBScript**, on en a déjà utilisées pas mal

- Len ()
- UBound ()
- ...

© Achret L

VBScript

Les fonctions en **VBScript**, on en a déjà utilisées pas mal

- Len ()
- UBound ()
- ...

Remarque

Il est possible de définir de nouvelles fonctions.

VBScript

Fonction

- Regroupement d'instructions réalisant une tâche
- Pouvant accepter un ou plusieurs paramètres et retournant une unique valeur
- Pouvant avoir ses variables locales
- Pouvant appeler une ou plusieurs autres fonctions (ou elle-même ⇒ récursivité)
- Aidant à décomposer un problème difficile en un ensemble de fonctions (sous algorithme) de complexité inférieure
- Permettant une structuration et une meilleure lisibilité
- Facilitant la maintenance du code

VBScript

Déclaration d'une fonction : syntaxe

```
Function NomDeLaFonction(param1, param2, ...)  
    ' Corps de la fonction  
    NomDeLaFonction = valeur_de_retour  
End Function
```

© Achref EL MOUELFI

VBScript

Déclaration d'une fonction : syntaxe

```
Function NomDeLaFonction(param1, param2, ...)  
    ' Corps de la fonction  
    NomDeLaFonction = valeur_de_retour  
End Function
```

Déclaration d'une fonction : exemple

```
Function Puissance(a, b)  
    Dim resultat  
    resultat = 1  
    For i = 1 To b  
        resultat = resultat * a  
    Next 'i  
    Puissance = resultat  
End Function
```


VBScript

Appel de la fonction puissance

```
Dim valeur  
valeur = Puissance(2, 3)  
WScript.Echo "Le résultat est " & valeur  
' Affiche 8
```

VBScript

Exercice

- Écrire une fonction `CalculMax2` qui prend deux paramètres de type nombre et retourne le plus grand.
- Écrire un programme qui demande à l'utilisateur de saisir deux nombres et appelle la fonction `CalculMax2` pour récupérer le plus grand et l'afficher.
- Écrire une fonction `CalculMax3` qui prend trois paramètres de type nombre et retourne le plus grand.
- Modifier l'algorithme précédent pour permettre à l'utilisateur de saisir trois nombres et utiliser `CalculMax3` pour récupérer et afficher le plus grand.

VBScript

Exercice

Écrire une fonction qui retourne la factorielle d'un nombre reçu en paramètre.

- $0! = 1$
- $1! = 1$
- $n! = n * (n - 1)!$

VBScript

Exercice

Écrire une fonction qui retourne `vrai` si le paramètre reçu est un nombre premier (divisible seulement par 1 et lui-même), `faux` sinon.

- 5 est un nombre premier parce qu'il est seulement divisible par 1 et 5.
- 6 n'est pas premier parce qu'il est divisible par 1, 2, 3 et 6.

VBScript

Procédure

- Appelée aussi sous-routine
- Fonction ne retournant pas de valeur
- Définie à l'aide du mot-clé `Sub`

VBScript

Déclaration d'une procédure : syntaxe

```
Sub NomDeLaProcédure (param1, param2, ...)  
    ' Corps de la procédure  
End Sub
```

© Achref EL MOU

VBScript

Déclaration d'une procédure : syntaxe

```
Sub NomDeLaProcEDURE (param1, param2, ...)  
    ' Corps de la procédure  
End Sub
```

Déclaration d'une procédure : exemple

```
Sub AfficherBienvenue (nom)  
    WScript.Echo "Bienvenue, " & nom & "!"  
End Sub
```

VBScript

Appel de la procédure AfficherBienvenue

```
AfficherBienvenue ("Doe")  
' Affiche Bienvenue, Doe!
```

© Achref EL MOU

VBScript

Appel de la procédure AfficherBienvenue

```
AfficherBienvenue ("Doe")  
' Affiche Bienvenue, Doe!
```

Ou

```
AfficherBienvenue "Doe"  
' Affiche Bienvenue, Doe!
```

VBScript

Exercice

- Écrire une procédure qui prend en paramètre deux nombres et qui permute leurs valeurs.
- Dans le programme principal, demander à l'utilisateur de saisir deux nombres, ensuite appeler la procédure pour permuter les deux valeurs et enfin afficher les.

© Achre

VBScript

Exercice

- Écrire une procédure qui prend en paramètre deux nombres et qui permute leurs valeurs.
- Dans le programme principal, demander à l'utilisateur de saisir deux nombres, ensuite appeler la procédure pour permuter les deux valeurs et enfin afficher les.

Remarques

Faites une exécution à la main et vérifiez que le programme n'affiche pas le résultat attendu.

VBScript

Deux modes de transmission

- **Transmission par valeur** (par défaut) : les valeurs des paramètres effectifs sont recopiées dans les paramètres formels correspondants au moment de l'appel de la procédure. Les paramètres effectifs ne subissent aucune modification.
- **Transmission par adresse** (référence) : les adresses des paramètres effectifs sont transmises à la procédure appelée. Les paramètres effectifs subissent les modifications lors de l'exécution de la procédure.

VBScript

Exemple de paramètres passées par référence

```
Sub Permuter(ByRef a, ByRef b)
    Dim temp
    temp = a
    a = b
    b = temp
End Sub
```

VBScript

Considérons l'algorithme suivant

```
Dim x, y
x = 10
y = 20

WScript.Echo "Avant permutation : x = " & x & ", y = " & y
Permuter x, y
WScript.Echo "Après permutation : x = " & x & ", y = " & y
```

VBScript

Récursivité

- Une fonction est dite **récursive** si elle s'appelle elle-même.
- À chaque appel, une nouvelle instance indépendante sera créée avec ses propres paramètres.

VBScript

Exemple de récursivité avec la fonction Factorielle

```
Function Factorielle(n)
    If n <= 1 Then
        Factorielle = 1
    Else
        Factorielle = n * Factorielle(n - 1)
    End If
End Function

' Exemple d'utilisation de la fonction Factorielle
Dim nombre, resultat
nombre = 5
resultat = Factorielle(nombre)
WScript.Echo "La factorielle de " & nombre & " est " & resultat
```


VBScript

Exercice

Écrire une fonction récursive qui calcule le nième terme de la suite de **Fibonacci** (n étant l'unique paramètre de la fonction).

- $U_0 = 0$
- $U_0 = 1$
- $U_n = U_{n-1} + U_{n-2}$