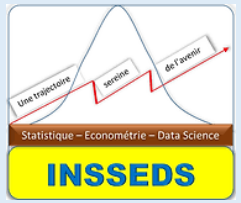
Formation Initiation à l'algorithme

- Introduction à l'algorithmique (histoire et définition)
- Les variables
    - Types de variable
    - Effectation
    - Exercices
- Les expressions et les operateurs
- La lecture et l'écriture
- Les Tests
- la Logique
- Les boucles
- Les tableaux
- Fonctions et les procédures
- Les Notions complémentaires
    - Programmation structurée
    - Programmation non structurée

- Extras : L'orienté Objet.

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica



AFRICA

Formation Initiation à l'algorithme

- Introduction à l'algorithmique (histoire et définition)

« Les ordinateurs sont comme les dieux de l'Ancien Testament : avec beaucoup de règles, et sans pitié. » - Joseph Campbell

« Il y a 10 sortes de gens au monde : ceux qui connaissent le binaire et les autres » - Anonyme

« Un langage de programmation est une convention pour donner des ordres à un ordinateur. Ce n'est pas censé être obscur, bizarre et plein de pièges subtils. Ca, ce sont les caractéristiques de la magie. » - Dave Small

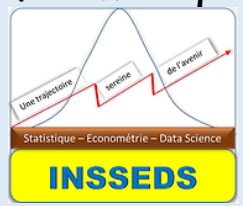
Un algorithme, c'est une suite d'instructions, qui une fois exécutée correctement, conduit à un résultat donné.

un algorithme doit donc contenir uniquement des instructions compréhensibles par celui qui devra l'exécuter.

l'algorithmique exprime les instructions résolvant un problème donné **indépendamment des particularités de tel ou tel langage.**

Apprendre l'algorithmique, c'est apprendre à manier la **structure logique** d'un programme informatique.

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica



Formation Initiation à l'algorithme

- Les variables

Pour employer une image, une variable est une **boîte**, que le programme (l'ordinateur) va repérer par une **étiquette**. Pour avoir accès au contenu de la boîte, il suffit de la désigner par son étiquette.

- Déclaration de la variable

  Pour pouvoir utiliser une variable, il faut **créer la boîte et de lui coller une étiquette**, c'est ce qu'on appelle une déclaration de variable.

  Lorsqu'on déclare une variable, il ne suffit pas de créer une boîte (réserver un emplacement mémoire) ; encore doit-on préciser ce que l'on voudra mettre dedans, car de cela dépendent la taille de la boîte (de l'emplacement mémoire) et le type de codage utilisé.

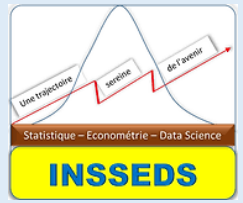
- Types de la variable

  Types numériques classiques – il existe plusieurs types de numériques en occurrence les entiers simple et long, les réels simple et double
  Types booléen– il existe plusieurs types de booléen qui prend 2 valeurs VRAI ou FAUX.
  Types Date
  Types Monétaire– strictement deux chiffres après la virgule.
  Types Alphanumérique– qui englobe tout ce qui est caractère, et des valeurs textuelles.

- Exemple:

  Variable nombre **en Numérique**
  ou encore
  Variables PrixHT, TauxTVA, PrixTTC **en Numérique**
  ou encore
  Variables Nom, Prenom **en Alphanumerique**

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica



Formation Initiation à l'algorithme

- L'affectation

  Exemples:

Exemple n°1
```
Début
    Riri ← "Loulou"
    Fifi ← "Riri"
Fin
```

Exemple n°2
```
Début
    nom ← " DOUAMPO"
    prenom ← " Jean-Michel"
Fin
```

Exemple n°3
```
Début
    a ← 10
    somme ← a + 10
Fin
```

- Expressions et operateurs:

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

Un opérateur est un signe qui relie deux valeurs, pour produire un résultat.

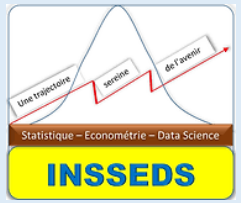
Operateurs Numériques
+ : addition
- : soustraction
* : multiplication
/ : division

Operateurs des Alphanumériques
& : concaténer

Operateurs Logiques
OU, ET, NON

Exemples:
```
Début
A ← "Gloubi"
B ← "Boulga"
C ← A & B
Fin
```

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica



Formation Initiation à l'algorithme

- Lecture et l'Ecriture

Dès que le programme rencontre une instruction Lire, l'exécution s'interrompt, attendant la frappe d'une valeur au clavier et ecrire permet d'afficher le contenu de lire.

Ex:

```
Debut
        Variable NomFamille est Alphanumérique
        Ecrire "Entrez votre nom : "
        Lire NomFamille
Fin
```

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica

Formation Initiation à l'algorithme

- **Les Tests**
  - **Tests simples**

```
Si booléen Alors
        Instructions 1
Sinon Instructions 2
Finsi
```

```
Si Condition Alors
        Instructions 1
Finsi
Si condition2 Alors
        Instructions 2
Finsi
```

  - **Tests imbriqués:**

```
Variable Temp en Entier
Début
    Ecrire "Entrez la température de l'eau :"
    Lire Temp
    Si Temp =< 0 Alors
        Ecrire "C'est de la glace"
    FinSi
    Si Temp > 0 Et Temp < 100 Alors
        Ecrire "C'est du liquide"
    Finsi Si Temp > 100 Alors
        Ecrire "C'est de la vapeur"
    Finsi
Fin
```

?? Version en cas 2

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica

Formation Initiation à l'algorithme

- **Les Boucles – Structures iteratives**
  - **La boucle « Tant que »**

```
Variable Rep en Caractère
Début
    Ecrire "Voulez vous un café ? (O/N)"
    TantQue Rep <> "O" et Rep <> "N"
        Lire Rep
    FinTantQue
Fin
```

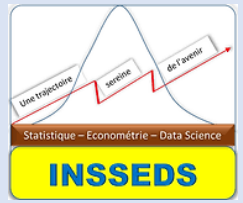
```
Variable Truc en Entier
Début
Truc ← 0
TantQue Truc < 15
Truc ← Truc + 1
Ecrire "Passage numéro : ", Truc
FinTantQue
Fin
```

  - **La boucle « Pour »**

```
Pour Compteur ← Initial à Final Pas ValeurDuPas
…
Instructions
…
Compteur suivant
```

```
Variable Truc en Entier
Début
Pour Truc ← 1 à 15
Ecrire "Passage numéro : ", Truc
Truc Suivant
Fin
```

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica



Formation Initiation à l'algorithme

- **Les Tableaux**

Un ensemble de valeurs portant le même nom de variable et repérées par un nombre, s'appelle un tableau, ou encore une variable indicée.
Le nombre qui, au sein d'un tableau, sert à repérer chaque valeur s'appelle l'indice.
Chaque fois que l'on doit désigner un élément du tableau, on fait figurer le nom du tableau, suivi de l'indice de l'élément, entre parenthèses

- **Définition de Tableau:**

Tableau Note(11) **en Entier**
Tableau Note(11,10) **en Entier //Tableau multidimensionnel avec 10*11 comme dimension.**

- **Exemple de Tableau:**

```
Tableau Note(11) en Numérique
Variables Moy, Som en Numérique
Début
        Pour i ← 0 à 11
        Ecrire "Entrez la note n°", i
        Lire Note(i)
        i Suivant
        Som ← 0
        Pour i ← 0 à 11
        Som ← Som + Note(i)
        i Suivant
        Moy ← Som / 12
FIN
```

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica

Formation Initiation à l'algorithme

- **Les Procédures et les fonctions:**

Une fonction réalise une simple opération dont le résultat peut être, par la suite, utilisé par une instruction. Une procédure est une instruction composée qui peut prendre des paramètres et dont le rôle est de modifier l'état courant. Les procédures ne retournent pas de résultat.

- **Exemple:**

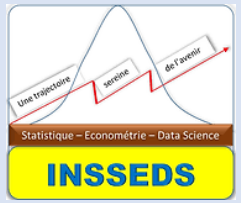
```
Fonction RepOuiNon(Msg en Caractère) en Caractère
    Ecrire Msg
    Truc ← ""
    TantQue Truc <> "Oui" et Truc <> "Non"
        Ecrire "Tapez Oui ou Non"
        Lire Truc
    FinTantQue
Renvoyer Truc
Fin Fonction
```

```
Procédure RepOuiNon(Msg en Caractère par valeur,
Truc en Caractère par référence)
    Ecrire Msg
    Truc ← ""
    TantQue Truc <> "Oui" et Truc <> "Non"
        Ecrire "Tapez Oui ou Non"
        Lire Truc
    FinTantQue
Fin Fonction

M ← "Etes-vous marié ?"
Appeler RepOuiNon(M, T)
Ecrire "Votre réponse est ", T
```

PS: il existe 2 types de passage de paramètres soit par valeur, soit par référence.
Par référence, on peut la récupérer en dehors de la fonction ou la procédure.

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica



Formation Initiation à l'algorithme

- **Notions complémentaires (Programmation structurée et non structuré):**
  - **Exemple structuré:**

Si condition **Alors**
    instructions 1
Sinon
    instructions 2
FinSi

Note: **La programmation structurée et la programmation non structurée sont deux approches différentes pour écrire du code informatique. Voici les principales différences entre les deux :**
- **La lisibilité**
- **Organisation**
- **Utilisation courante du code.**

- **Exemple structuré:**

1000 **Si** condition **Alors Aller** En 1200
1100 instruction 1
1110 etc.
1120 etc.
1190 **Aller en** 1400
1200 instruction 2
1210 etc.
1220 etc.
1400 suite de l'algorithme

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica

Formation Initiation à l'algorithme

- **Programmation orienté Objet:**
  - Langage procédural

Langage basé sur des appels de procédures. Une procédure contient une série d'étapes à réaliser et n'importe quelle procédure peut être appelée à n'importe quelle étape de l'exécution du programme. Cette approche permet de réutiliser le même code à différents emplacements du programme.

  - Langage Orienté Objet

La programmation orientée objet répartit l'effort de résolution des problèmes sur un ensemble d'objets communiquant entre eux. Un objet représente un concept ou une entité physique. Chaque objet est composé de 2 parties : une partie statique (les attributs) qui décrit l'objet et une partie dynamique qui détermine les comportements de celui-ci. Dans la partie dynamique, on retrouve du langage procédural.

  - Définition

La programmation orientée objet (POO) est un paradigme de programmation qui repose sur la notion d'objets, qui sont des instances de classes. Les objets peuvent encapsuler des données et des méthodes qui agissent sur ces données. La POO vise à organiser le code de manière modulaire, en regroupant des fonctionnalités connexes dans des classes, ce qui favorise la réutilisation du code et la maintenance.

  - concepts

En POO, les principaux concepts incluent les classes, les objets, l'encapsulation (cachement des détails internes de l'objet), l'héritage (la capacité d'une classe à hériter des caractéristiques d'une autre), et le polymorphisme (la capacité à traiter des objets de différentes classes de manière uniforme).

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica

Formation Initiation à l'algorithme

- **Programmation orienté Objet:**
  - Exemple de Classe

classe Enseignant
propriétés
nom
matière
constructeur Enseignant(nom, matière)
méthodes
noter(devoir)
sePrésenter()

  - Exemple d'utilisation de la Classe Enseignant : Creation de 2 objets Enseignant

```
guillaume = new Enseignant("Guillaume", "Psychologie");
liliane = new Enseignant("Liliane", "Poésie");

guillaume.matière; // "Psychologie"
guillaume.sePrésenter(); // "Je m'appelle Guillaume et je serai votre enseignant·e en psychologie."

liliane.matière; // "Poésie"
liliane.sePrésenter(); // "Je m'appelle Liliane et je serai votre enseignant·e en poésie."
```

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica

Formation Initiation à l'algorithme

- **Programmation orienté Objet:**
  - Les concepts de OO
    - Instanciation (voir exemple précèdent)
    - L'héritage

Imaginons qu'on veuille également représenter les étudiants dans notre système. À la différence des enseignants, un élève ne peut pas noter de devoirs, n'enseigne pas une matière donnée et appartient à une promotion d'une année donnée. Toutefois, les élèves ont également un nom et peuvent aussi se présenter. On pourrait alors écrire la définition de la classe d'un élève ainsi :

classe Personne
propriétés
nom
constructeur Personne(nom)
Méthodes
sePrésenter()

classe Enseignant : **étend** Personne
propriétés
matière
constructeur Enseignant(nom, matière)
méthodes
noter(devoir)
sePrésenter()

classe Élève : **étend** Personne
Propriétés
année
constructeur Élève(nom, année)
méthodes
sePrésenter()

  - Exemple:

```
guillaume = new Enseignant("Guillaume", "Psychologie");
guillaume.sePrésenter(); // "Je m'appelle Guillaume et je serai votre enseignant·e en psychologie."

suzanne = new Élève("Suzanne", 1);
suzanne.sePrésenter(); // "Je m'appelle Suzanne et je suis en première année."

thomas = new Person("Thomas");
thomas.sePrésenter(); // "Je m'appelle Thomas."
```

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica

Formation Initiation à l'algorithme

- **Programmation orienté Objet:**
  - Les concepts de OO
    - L'encapsulation

Les objets fournissent une interface au reste du code qui voudrait les utiliser et ils maintiennent leur propre état interne. L'état interne d'un objet est **privé**, et peut uniquement être manipulé par les méthodes de l'objet (mais pas par celles des autres objets). Séparer l'état privé interne d'un objet et son interface publique est ce qu'on appelle **l'encapsulation**.
Par exemple, si les élèves ne sont autorisés à étudier le tir à l'arc qu'à partir de la deuxième année, on pourrait implémenter cette règle en exposant la propriété année pour que le code externe puisse la consulter et décider si l'élève peut s'inscrire au cours :

classe Élève : **étend** Personne
Propriétés
privée année
constructeur Élève(nom, année)
méthodes
sePrésenter()
peutEtudierTirArc() { renvoyer ceci.année > 1 }

unÉlève = nouvel Élève('Weber', 1)
unÉlève.année // erreur : 'année' est une propriété privée de Élève

Facilitateur: **Jean-Michel DOUAMPO**
Consultant IT et Gérant de DIDAfrica