XDNATIS

Algorithmique

Laurence
XDNATIS

XWIKI NATIS

Compétence demandée :
Maîtriser 5 concepts

XWIKI NATIS

1. Variables
2. Instructions de base
3. Blocs
4. Conditions
5. Boucles

XWIKI NATIS

La maîtrise de ces 5 concepts vous permettra d'écrire du code (traduction d'un algorithme dans un langage)

XWNATIS

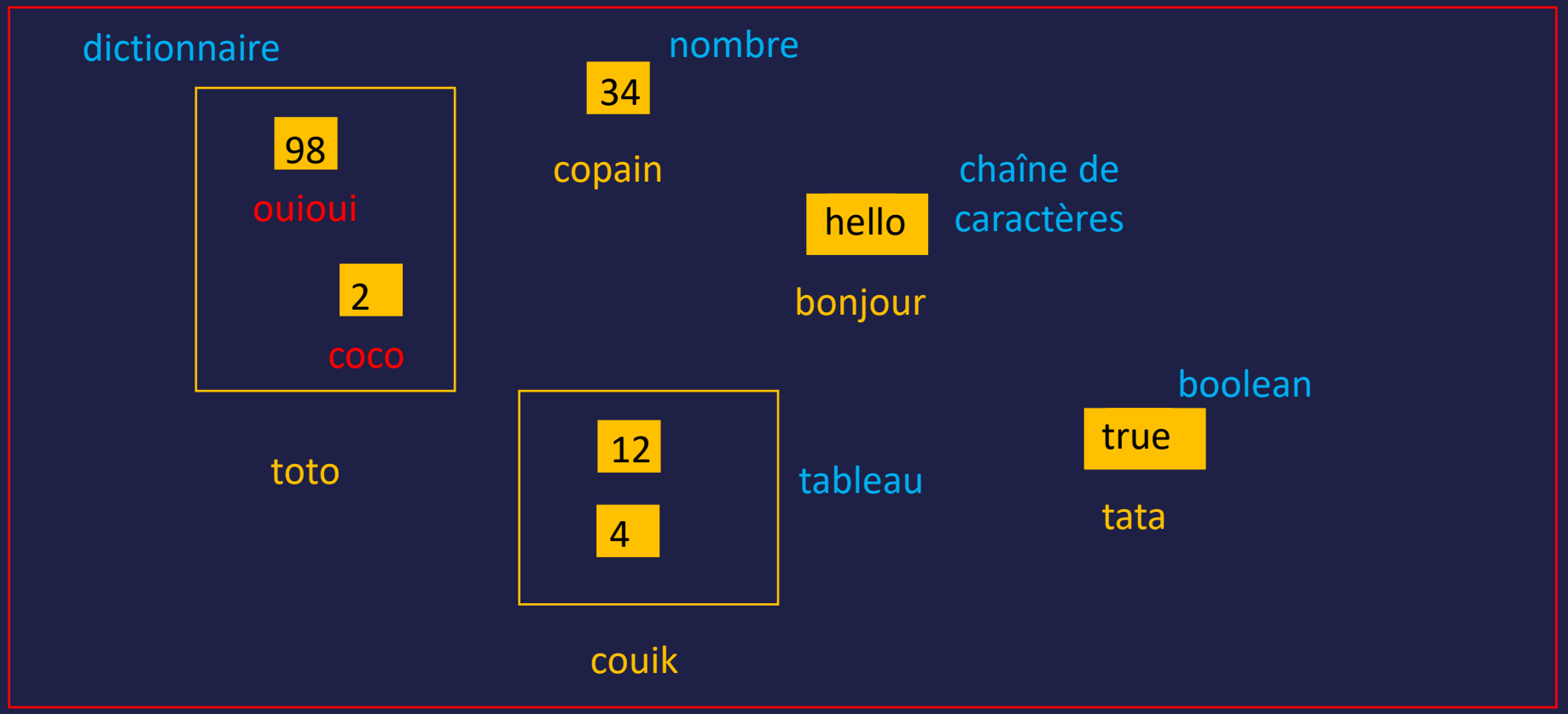
Processus habituel d'acquisition de compétences informatiques

1. Savoir écrire des algorithmes
2. Savoir traduire les algorithmes dans un langage
3. Savoir écrire et emboiter des fonctions
4. Savoir écrire des classes
5. Savoir concevoir une architecture

XWIKI NATIS

1. Variables

XNATIS



La RAM stocke des variables typées

XEBIA NATIS

Les variables sont TYPÉES !

XWIKI NATIS

Type = Structure de données

XWIKI NATIS

Les types permettent à l'ordinateur d'identifier les actions possibles

Les types prennent un espace différent en RAM

XWIKI NATIS

2. Instructions de base

XDNATIS

Affectation

XEBIA NATIS

```
taille <- 34
```

```
toto <- [23, 34, 32, 3]
```

```
resultat <- 'Petit'
```

```
yop <- Vrai
```

XDNATIS

Structure de données	Actions possibles
Nombre	Addition Soustraction Division Multiplication
Chaîne de caractères	Concaténation
Boolean	Et Ou Non
Tableau	Adressage (position) Ajout Suppression
Dictionnaire	Adressage (clé) Ajout Suppression

XEBIA NATIS

Structure de données	Actions possibles	Javascript
Nombre	Addition Soustraction Division Multiplication	+ - / *
Chaîne de caractères	Concaténation	+
Boolean	Et Ou Non	&& || !
Tableau	Adressage (position) Ajout Suppression	[position] push(element) splice(position)
Dictionnaire	Adressage (clé) Ajout Suppression	[clé] [clé] = element delete [clé]

XEBIA NATIS

Exercices d'algorithmique

1 & 2

XNATIS

Exercice 1 :

Echanger les valeurs de 2 variables R et Q

R <- 5

Q <- 34

Algorithme

A <- R (Création de la nouvelle variable à qui on affecte 5 (valeur initiale de R)

R <- Q (On affecte à R la valeur initiale de Q qui est 34)

Q <- A (On affecte à Q la valeur initiale de R qui avait été transmise à la nouvelle variable A)

XNATIS

Exercice 2 :

Echanger les valeurs de 3 variables R, Q et V. Dans V, mettre Q
Dans Q, mettre R et dans R, mettre V.

R <- 5
Q <- 34
V <- 45

Algorithme

A <- V (Création de la nouvelle variable A à qui on affecte 45 (valeur initiale de V)
V <- Q (On affecte à V la valeur initiale de Q qui est 34)
Q <- R (On affecte à Q la valeur initiale de R qui est 5)
R <- A (On affecte à R la valeur initiale de V qui était 45 et qui avait été affecté à A)

XWIKI NATIS

3. Blocs

XEBIA NATIS

Un bloc permet de rassembler des instructions

XEBIA NATIS

Les variables définies dans un bloc meurent à la fin du bloc

Portée (scope)

XWIKI NATIS

Un bloc est un ensemble d'instructions qui peuvent être conditionnées ou répétées

XWIKI NATIS

4. Conditions

XDNATIS

Une condition permet de conditionner l'exécution d'un bloc

XWIKI NATIS

```
taille <- 34

resultat <- 'Petit'
@Si taille >= 50
    @DebutBloc
    resultat <- 'Grand’
    @FinBloc
```

```
taille <- 34

resultat <- 'Petit'
@Si   @Non (taille < 50)
    @DebutBloc
    resultat <- 'Grand’
    @FinBloc
```

XDNATIS

Une condition se base sur 1 ou plusieurs prédicats

XNATIS

La valeur logique d'un prédicat est toujours « Vrai » ou « Faux »

XNATIS

```
taille <- 34
forme <- 'Rectangle'

resultat <- 'Petit'
@Si taille >= 50 @Et forme = 'Rectangle’
    @DebutBloc
    resultat <- 'Grand’
    @FinBloc
```

XWIKI NATIS

```
taille <- 34
forme <- 'Rectangle'

resultat <- 'Petit'
@Si taille >= 50 @Ou forme = 'Rectangle’
    @DebutBloc
    resultat <- 'Grand’
    @FinBloc
```

XWIKI NATIS

Opérateurs binaires sur les prédicats

ET = « et en même temps … »

OU = « ou soit … »

XWIKI NATIS

Opérateurs binaires

a ET b

c OU d

XWIKI NATIS

Opérateurs unaires sur les prédicats

NON = « ne pas … » ou
« contraire »

XNATIS

Opérateurs unaires sur les prédicats

NON (a)

XNATIS

Table de vérité

Les tables de vérité présentent tous les résultats possibles d'une opération logique

a	b	a ET b
Faux	Faux	Faux
Faux	Vrai	Faux
Vrai	Faux	Faux
Vrai	Vrai	Vrai

a	b	a OU b
Faux	Faux	Faux
Faux	Vrai	Vrai
Vrai	Faux	Vrai
Vrai	Vrai	Vrai

XEFI NATIS

Loi de De Morgan

La loi De Morgan permet de « casser »
un NON englobant un ET ou un OU

XEBIA NATIS

Loi de
De Morgan

NON (a ET b) = NON (a) OU NON (b)
NON (a OU b) = NON (a) ET NON (b)

XEBIA NATIS

Exercices d'algorithmique
3, 4, & 5

XNATIS

Exercice 3 :

Vérifier si un utilisateur est majeur grâce à son âge. Mettre une variable « resultat » à Vrai si cela est le cas, sinon mettez la à Faux.

Algorithme

Instruction : Age <- 25

```
@Si age >= 18
        @DebutBloc
                resultat <- Vrai
        @FinBloc
@Sinon
        @DebutBloc
                resultat <- Faux
        @FinBloc
```

Algorithme

Instruction : resultat <- Faux

```
@Si age >= 18
        @DebutBloc
                resultat <- Vrai
        @FinBloc
```

XNATIS

Exercice 4 :

Vérifier si un utilisateur est mineur grâce à son âge. Mettre une variable « resultat » à Vrai si cela est le cas, sinon mettez la à Faux.

Instruction : Age <- 12

Algorithme

```
@Si age < 18                    (Ici l'âge est strictement inférieur à 18)
        @DebutBloc
                resultat <- Vrai
        @FinBloc
@Sinon
        @DebutBloc
                resultat <- Faux
        @FinBloc
```

XNATIS

Exercice 5 :

Echanger/Inverser les 2 premiers éléments d'un tableau.

Instruction :

tab <- [23, 4, 2, 543, 34, 12, 4]

Algorithme

A <- tab[0]
tab[0] <- tab[1]
tab[1] <- A

On obtient : tab <- [4, 23, 2, 543, 34, 12, 4]

XWIKI NATIS

5. Boucles

XWIKI NATIS

Les boucles

Les boucles permettent de répéter un bloc d'instructions

XWIKI NATIS

Il y a 3 types de boucles pour répéter un bloc

1. @PourChaque
2. @Pour @De @A
3. @TantQue ou Boucle + @Stop

XEBIA NATIS

1. Il faut s'arrêter à la fin du tableau

```
tab <- [23, 43, 32, 4, 3]

@PourChaque element @Dans tab
    @DebutBloc
    Afficher element
    @FinBloc
```

XEBIA NATIS

2. Il faut s'arrêter avec un nombre maximal

```
tab <- [23, 43, 32, 4, 3]

@Pour i @De 0 @A 4
    @DebutBloc
    Afficher tab[i]
    @FinBloc
```

XEBIANATIS

3. Il faut s'arrêter avec une condition

```
tab <- [23, 43, 32, 4, 3]

position <- 0
@TantQue tab[position] < 30
    @DebutBloc
    position <- position + 1
    @FinBloc
Afficher position
```

```
tab <- [23, 43, 32, 4, 3]

position <- 0
@PourChaque element @Dans tab
    @DebutBloc
    @Si element >= 30
        @DebutBloc
        Afficher position
        @Stop
        @FinBloc
    position <- position + 1
    @FinBloc
```

XWIKI NATIS

Exercices d'algorithmique
6, 7, 8, 9, 10, 11 & 12

XNATIS

Exercice 6 :

Compter le nombre d'éléments dans un tableau. Mettre le compte dans une variable "**resultat**".

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Algorithme

```
resultat <- 0

@PourChaque element  @Dans tab
        @DebutBloc
                resultat <- resultat + 1
        @FinBloc
```

XNATIS

Exercice 7.1 :

Faire la somme des éléments d'un tableau.

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Algorithme

```
resultat <- 0

@PourChaque element @Dans tab
        @DebutBloc
                resultat <- resultat + element
        @FinBloc
```

XNATIS

Exercice 7.2 :

Faire la somme des éléments d'un tableau.

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Algorithme

```
compte <- 0
@PourChaque element @Dans tab
        @DebutBloc
                compte <- compte + 1
        @FinBloc

resultat <- 0
@Pour i @De 0 @A  compte
        @DebutBloc
                resultat <- resultat + tab[i]
        @FinBloc
```

XEBIANATIS

Exercice 8 :

Trouver l'élément maximum d'un tableau. Mettre le nombre maximum dans une variable "**resultat**".

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Algorithme

```
resultat <- 0
@PourChaque element @Dans tab
    @DebutBloc
        @Si element > resultat
            @DebutBloc
                resultat <- element
            @FinBloc
    @FinBloc
```

XNATIS

Exercice 9 :

Trouver l'élément minimum d'un tableau. Mettre le nombre maximum dans une variable "**resultat**".

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Algorithme

```
resultat <- tab[0]
@PourChaque element @Dans tab
        @DebutBloc
            @Si element < resultat
                    @DebutBloc
                            resultat <- element
                    @FinBloc
        @FinBloc
```

XNATIS

Exercice 10.1 :

Trouver le premier élément supérieur à 500. S'il n'y en a pas, le résultat doit être 0.

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Algorithme (Méthode 1 : For… of)

```
resultat <- 0
@PourChaque element @Dans tab (Pour chaque élément du tableau parcouru)
  @DebutBloc
    @Si element > 500
        @DebutBloc
            resultat <- element
        @Stop (On termine la boucle avec l'instruction « Break »)
        @FinBloc
  @FinBloc
```

XNATIS

Exercice 10.2 :

Trouver le premier élément supérieur à 500. S'il n'y en a pas, le résultat doit être 0.

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Algorithme (Méthode 2 : While)

```
i <- 0
indiceMax <- longueur -1
@TantQue tab[i] < 500 @Et i <= indiceMax (Tant que l'élément à la position actuelle est strictement inférieur à 500 et que i ne dépasse pas la position maximale autorisée c'est que l'on a pas encore trouvé ce que l'on cherche, on peut avancer)
    @DebutBloc
        i <- i + 1 (On continue d'avancer)
    @Si i > indiceMax (Si la position actuelle est strictement supérieure à la position maximale autorisée)
        @DebutBloc
            @Stop (On termine la boucle avec l'instruction « Break »)
        @FinBloc
    @FinBloc
```

XNATIS

Exercice 10.3 :

Trouver le premier élément supérieur à 500. S'il n'y en a pas, le résultat doit être 0.

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Algorithme (Méthode 3 : For)

```
resultat <- 0
@Si tab[i] >= 500 (Si l'on trouve ce que l'on cherche)
        @DebutBloc
        resultat <- tab[i] (On affecte la valeur trouvée à resultat)
        @FinBloc
```

XNATIS

Exercice 11 :

Copier le tableau "tab" dans un autre tableau intitulé "autre".

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Algorithme

Méthode 1

```
autre <- []
@PourChaque element @Dans tab
        @DebutBloc
                Ajouter element Dans autre
        @FinBloc
```

Méthode 2

```
autre <- tab
```

XNATIS

Exercice 12 :

Copier les premiers éléments d'un tableau dont la somme fait au moins 500 (dans un autre tableau intitulé « autre ».

Instruction : tab <- [23, 4, 2, 543, 34, 12, 4]

Algorithme

```
autre <- []
somme <- 0
@PourChaque element @Dans tab
        @DebutBloc
                somme <- somme + element
                Ajouter element Dans autre
                @Si somme >= 500
                    @DebutBloc
                     @Stop
                    @FinBloc
        @FinBloc
```