

PRÉSENTATION

Intervenant / Programme / Evaluation / Elèves

HELLO WORLD !

I'm Clément Denis

Geek & software engineer

@Ynov - Ingesup

@Capgemini

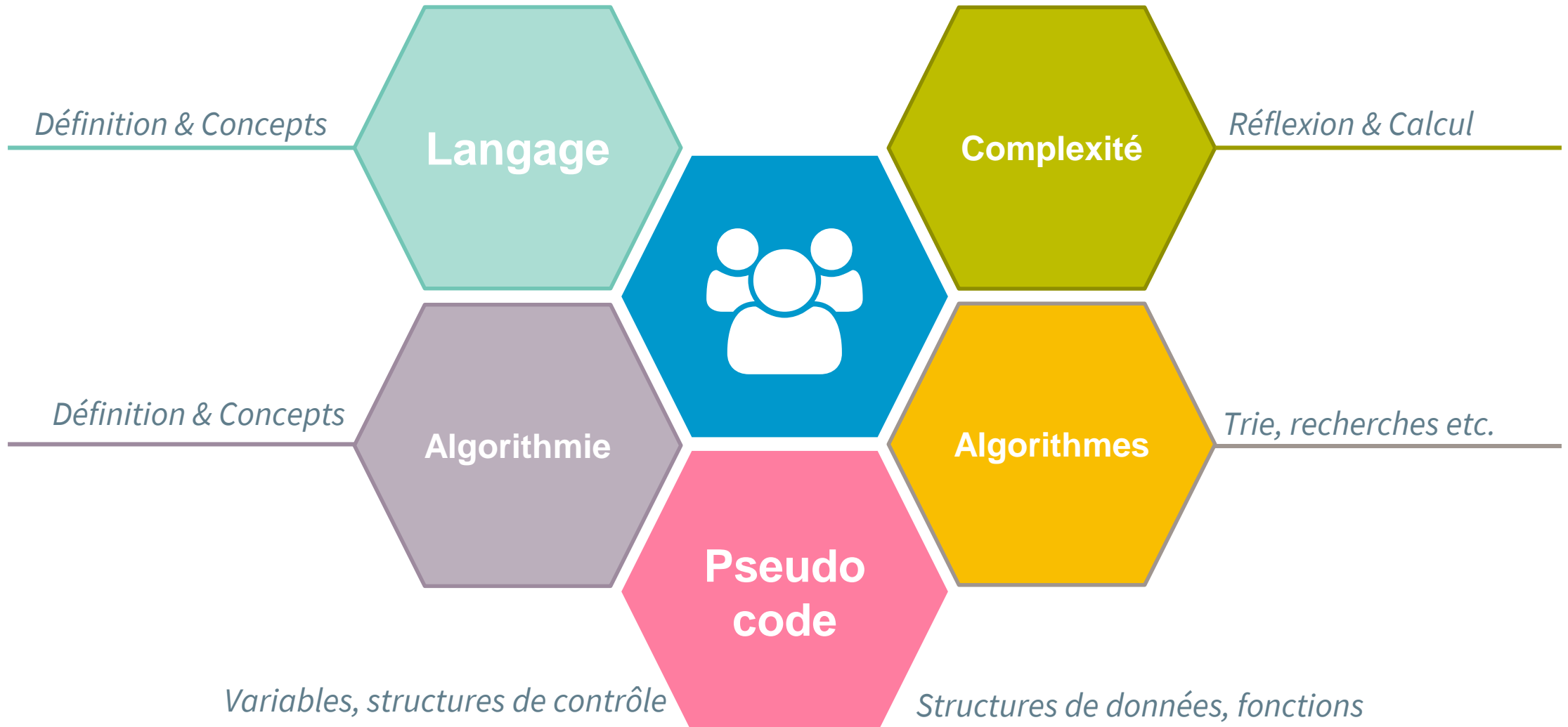
#intervenant

#formation

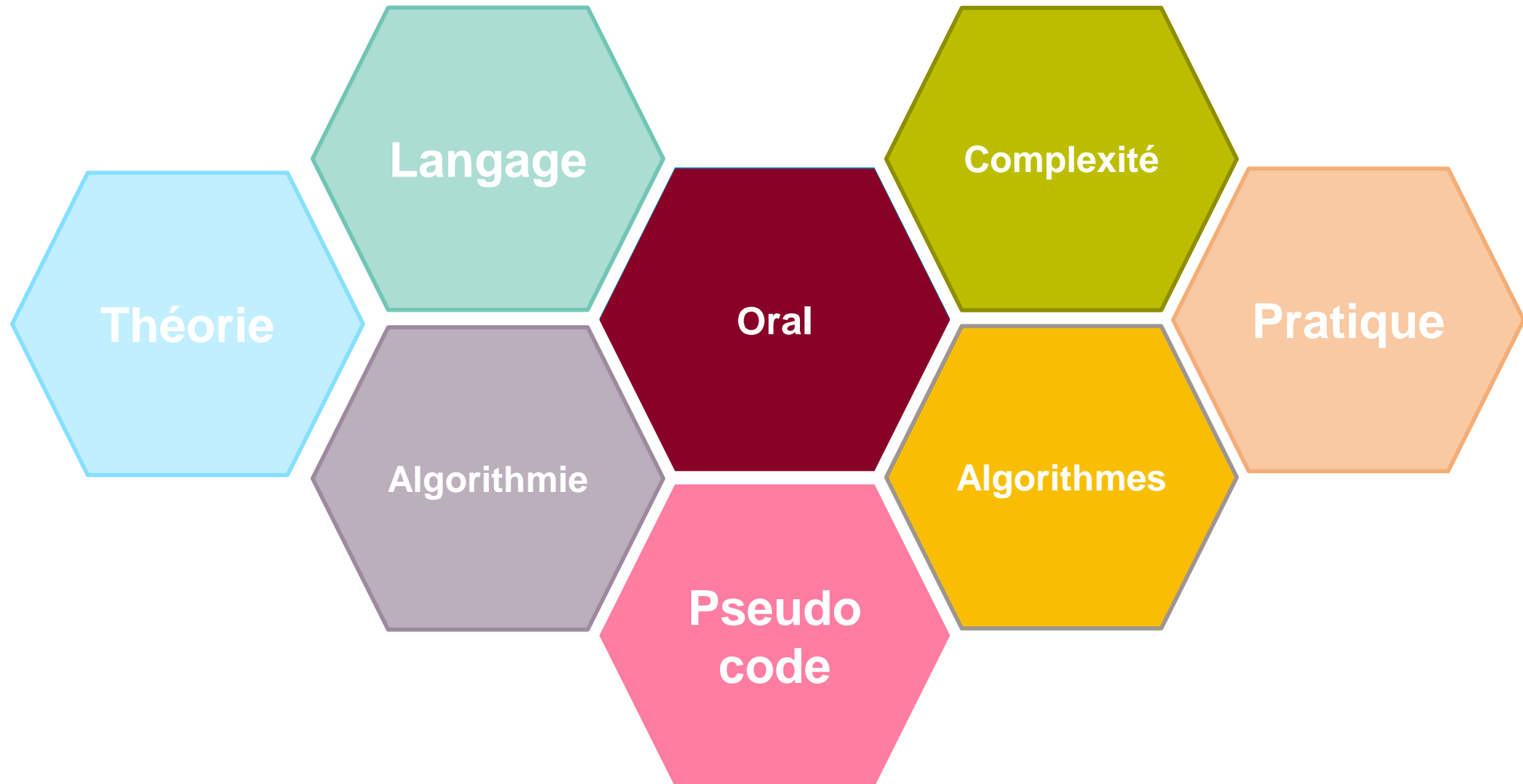
#challenge



ΔU PROGRAMME



EVALUATION



MODE DE TRAVAIL



CODE COULEURS

Cours magistral
Démonstration
Explication
L'intervenant à la parole

Question
Brainstorming
La classe à la parole, l'intervenant est animateur

Exercices
Travaux dirigés
Examen
L'intervenant à la parole

WHO ARE YOU ?

*Objectif professionnel ?
Antécédent ?
Passions ?
Algorithmie / Programmation ?*



ALGORITHMIQUE

ALGORITHMIQUE

DÉFINITION : ALGORITHMIQUE

« C'est la logique d'écrire des *algorithmes*. »



ALGORITHME

DÉFINITIONS : ALGORITHME

« Ensemble de règles opératoires dont l'application permet de résoudre un problème énoncé au moyen d'un nombre fini d'opérations » - Dictionnaire Larousse

« Un algorithme est une suite finie et non ambiguë d'opérations ou d'**instructions** permettant de résoudre un problème ou **d'obtenir un résultat** » - Wikipédia

« Un algorithme est une suite **d'instructions détaillées** qui, si elles sont **correctement exécutées**, conduit à un résultat donné. » - Mathématique

PIZZA FACILE, 45 MIN, 5 PERSONNES

Ingrédients

- 1 pâte à pizza prête à cuire
- 1 petite boîte de tomate
- 100 g de lardons nature
- 1 petite boîte de champignon de Paris en lamelles
- 2 poignées de gruyère râpée

Etape 1

Faire cuire dans une poêle les lardons et les champignons.

Etape 2

Dans un bol, verser la boîte de concentré de tomate, y ajouter un demi verre d'eau, ensuite mettre un carré de sucre (pour enlever l'acidité de la tomate) une pincée de sel, de poivre, et une pincée d'herbe de Provence.

Etape 3

Dérouler la pâte à pizza sur le lèche frite de votre four.

Etape 4

Avec une cuillère à soupe, étaler délicatement la sauce tomate, ensuite y ajouter les lardons et les champignons bien dorer. Parsemer de fromage râpée.

Etape 5

Mettre au four à 220°, thermostat 7-8, pendant 20 min (ou lorsque le dessus de la pizza est doré).

PIZZA AU PESTO D'ÉPINARDS ET AU SAUMON FUMÉ

Ingrédients

4 galettes au blé
1 petit pot de pesto d'épinards
1 tomates coupée en dés
100 g de saumon fumé
80 g de mozzarella di buffala
poivre
thym

Préparation

Préchauffer le four thermostat 6 à 7. Superposer 2 galettes Sur la plaque du four recouverte de papier sulfurisé. Badigeonner chaque galette généreusement de pesto d'épinards. Parsemer de saumon fumé en lamelles, de tomate en dés, de mozzarella en petits morceaux, de poivre et de thym. Enfourner 10 minutes. Servir chaud avec une petite salade.

FULL ENGLISH PIZZA

Ingredients

500g pack bread mix
a little sunflower oil
6 tbsp passata
4 pork sausages, skinned and quartered
140g mushrooms, sliced
8 steaky bacon rashers, halved
4 medium eggs

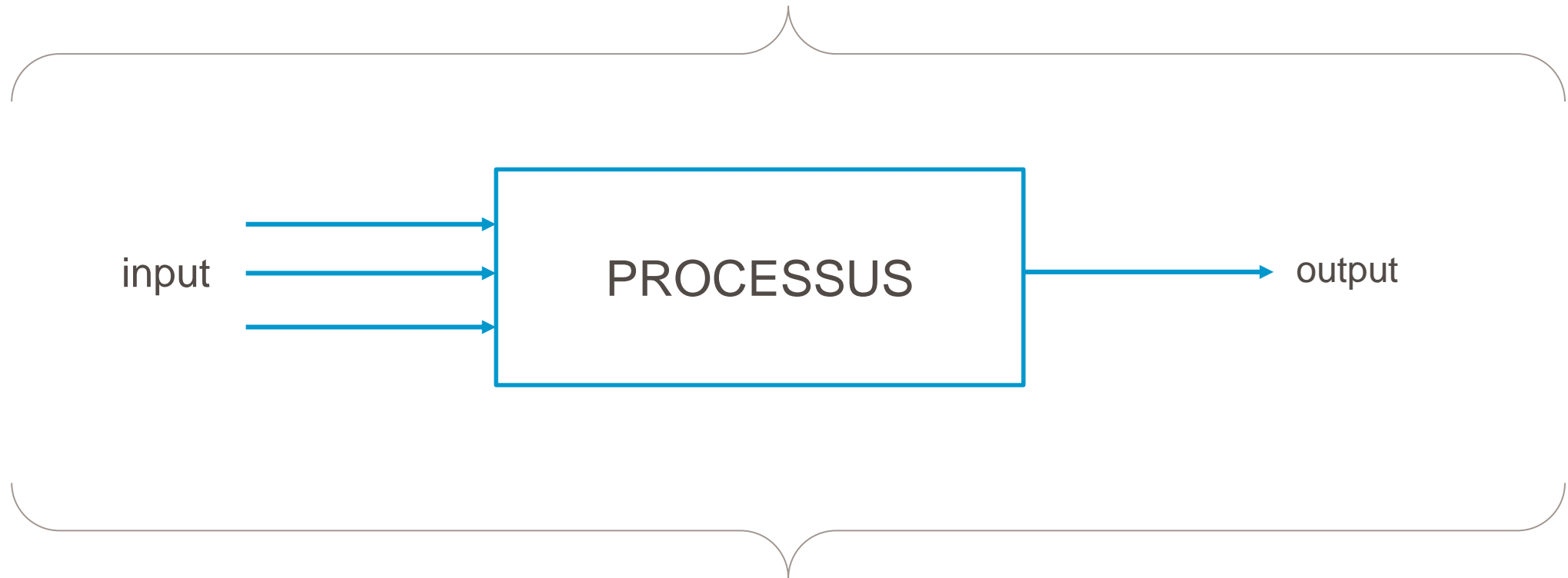
Method

1. Heat oven to 220C/200C fan/gas 7. Make up the bread mix following pack instructions. On a lightly floured surface, roll out to fit a lightly oiled 30 x 40cm baking tray, or two smaller trays. Spread the passata over the base, dot over the sausages and mushrooms, add the bacon, then cook for 20 mins at the top of the oven.
2. Remove the pizza from the oven and crack the eggs on. Return to the oven and cook for 5 mins more, or longer depending how well-cooked you like your eggs.

VOS REACTIONS ?

VISION BOITE NOIRE

PROBLEME



MÉTHODE DE RÉOLUTION D'UN PROBLÈME

1. **Comprendre** l'énoncé du problème
2. **Décomposer** le problème en **sous-problèmes** plus simple à résoudre
3. Associer à chaque sous problème, une **spécification** :
 - a) Les données nécessaires
 - b) Les données résultantes
 - c) La démarche à suivre pour arriver au résultat en partant d'un ensemble de données.
4. Elaboration d'un **algorithme**.

HAMBURGER 100% MAISON

1. **Comprendre** l'énoncé du problème
2. **Décomposer** le problème en **sous-problèmes** plus simple à résoudre
3. Associer à chaque sous problème, une **spécification** :
 - a) Les données nécessaires
 - b) Les données résultantes
 - c) La démarche à suivre pour arriver au résultat en partant d'un ensemble de données.
4. Elaboration d'un **algorithme**.



FORMALISER L'ÉCRITURE DE L'ALGORITHME

- ✓ Un langage commun
- ✓ Proche du langage naturel
- ✓ Indépendant de tout langage
- ✓ Facilement convertible en n'importe quelle autre langage
- ✓ Structuré



LANGUAGE

UN LANGUAGE

- ✓ **Règles de syntaxe**

Un grammaire formelle, ces règles régissent les différentes manières dont les éléments du langage peuvent être combiné

- ✓ **Vocabulaire**

L'ensemble des instructions construites d'après des symboles

- ✓ **Sémantique**

Le sens de chacune des phrases qui peuvent être construites dans le langage

- ✓ **Alphabet**

Lettre de A à Z par exemple

UN LANGUAGE DE PROGRAMMATION

- ✓ **Règles de syntaxe**

Un grammaire formelle, ces règles régissent les différentes manières dont les éléments du langage peuvent être combiné

- ✓ **Vocabulaire**

L'ensemble des instructions construites d'après des symboles

- ✓ **Sémantique**

Le sens de chacune des phrases qui peuvent être construites dans le langage

- ✓ **Alphabet**

Lettre de A à Z par exemple

- ✓ **Commentaires**

Texte permettant de donner des explications

- ✓ **Identifiants**

Mot clefs permettant d'organiser le script.

FORMALISER L'ÉCRITURE DE L'ALGORITHME

ADL

Algorithm Definition Language

- ✓ Un langage commun
- ✓ Proche du langage naturel
- ✓ Indépendant de tout langage
- ✓ Facilement convertible en n'importe quelle autre langage
- ✓ Structuré

EXERCICES DE GROUPE

Présentation et comparatif des langages de programmation 11/10/2018

- ✓ Groupe de 3 personnes
- ✓ Choisir 3 langages
- ✓ Brève description des langages
- ✓ Quels est son but ? Ou quel était sont but à sa création?
- ✓ Les éléments de recherches :
 - ✓ Niveau ?
 - ✓ Typage ?
 - ✓ Paradigme ?
 - ✓ Comment le langage s'exécute?
- ✓ Format PPT

ALGORITHM DEFINITION LANGUAGE (ADL)



Pseudo-code

Logigramme



ALGORITHM DEFINITION LANGUAGE (ADL)



Pseudo-code

Logigramme



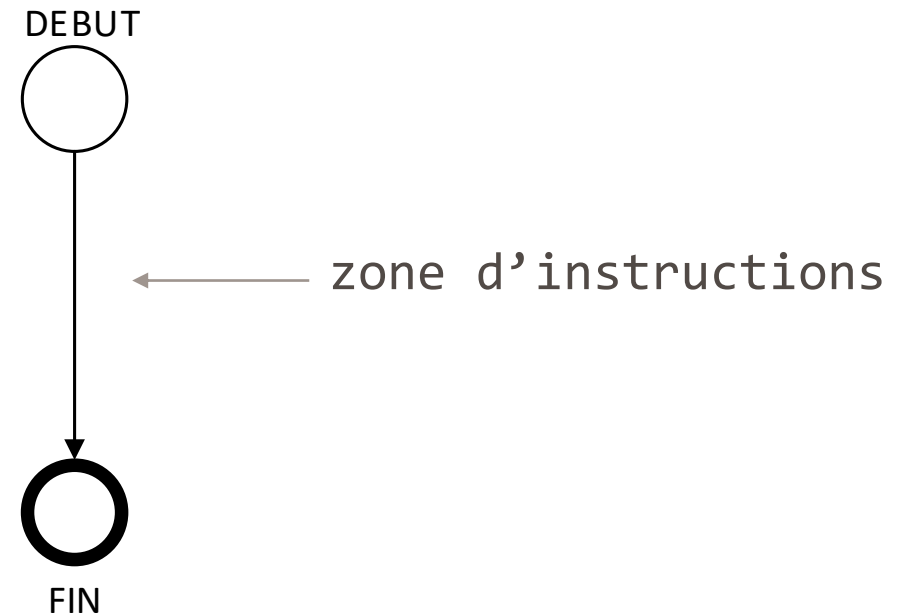
STRUCTURE DE L'ALGORITHME

Pseudo-code

```
ALGORITHME identifiant  
    <zone de déclaration>  
DEBUT  
    <zone d'instructions>  
FIN
```

/!\ indentation

Logigramme



/!\ pas de zone de déclaration

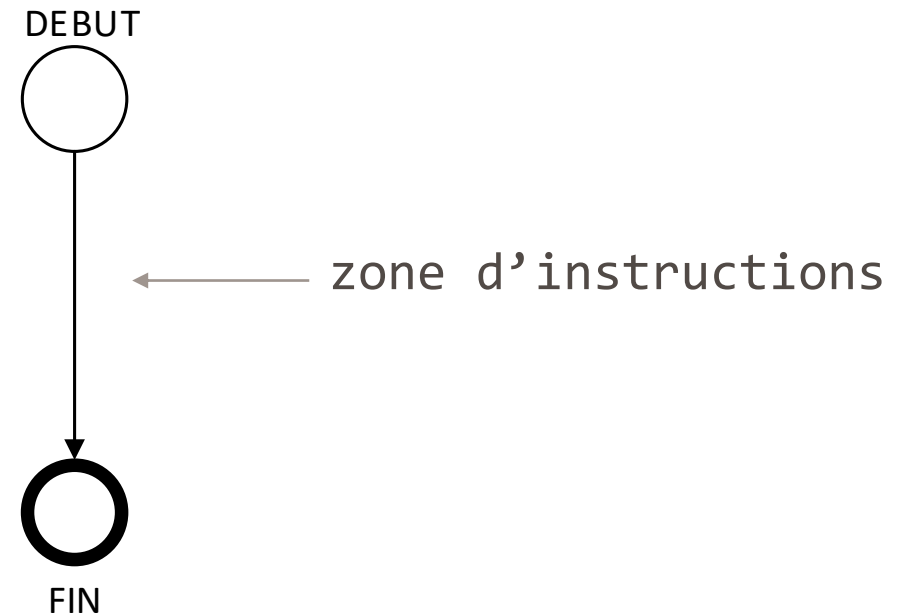
STRUCTURE DE L'ALGORITHME

Pseudo-code

```
ALGORITHME identifiant  
    <zone de déclaration>  
DEBUT  
    <zone d'instructions>  
FIN
```

/!\ indentation

Logigramme



/!\ pas de zone de déclaration

LES COMMENTAIRES

✓ **Commentaire**

Portion de code ignoré lors de l'exécution
A destination des développeurs
Permet d'expliquer une portion de code

ALGORITHME *identifiant*

DEBUT

```
// Commentaire  
/*  
    bloc  
*/
```

FIN

ZONE DE DÉCLARATION

✓ Variables

Repéré par son nom, pouvant contenir des données, qui pourront être modifiées lors de l'exécution du programme

✓ Constantes

Une constante est une variable dont la valeur est inchangeable lors de l'exécution d'un programme

- ❖ Déclaré avec :
Un identifiant.
Un type : *integer, float, boolean, char, string*

- ❖ Symbole d'affectation : ←

ALGORITHME *Exemple*

CONSTANTE

Type identifiant ← valeur

VARIABLE

Type identifiant

DEBUT

<zone d'instructions>

FIN

LES IDENTIFIANTS

✓ Règles de construction d'un identifiant

Lettres

Chiffres

Underscore : _

Ne doit jamais commencer par un chiffre.

✓ Conseil

L'identifiant doit représenter ce pourquoi on l'utilise : l'intention

toto	xy2	Clément	2nis
x	_TEST	3_TOTO	ELECTr0de
monNom	coucou	TeSt43_	Ingesup

LES TYPES

INTEGER

FLOAT

BOOLEAN

CHAR

STRING

35	3,14	'C'	'@'
"COUCOU"	CoCOU	VRAI	"FAUX"
'3'	K	TeSt43_	"23"

ZONE D'INSTRUCTIONS

- ✓ **Affectations**
- ✓ **Calculs**
- ✓ **Lire, Ecrire, Afficher**
- ✓ **Appeler des fonctions**
- ✓ **Utiliser des structures de contrôle**
 - **Structures conditionnelles**
 - **Structures de répétitions**
- ✓ **Les data-structures**

AFFECTATION

✓ Syntaxe :

identifiant \longleftarrow valeur

ALGORITHME *Exemple*

VARIABLE

INTEGER x, y, z

DEBUT

DEBUT

x \longleftarrow 2

y \longleftarrow 35

z \longleftarrow 8

z \longleftarrow x

y \longleftarrow z

FIN

CALCULS

✓ Les opérateurs :

Addition : +

Soustraction : -

Multiplication : x

Division : div ou /

Modulo : mod

NON : NOT ou NON

OU : OR ou OU

ET : AND ou ET

OU ex : XOR ou OUEX

ALGORITHME *Exemple*

VARIABLE

DEBUT

FIN

LIRE, ECRIRE

ALGORITHME *Exemple*

VARIABLE

INTEGER : age

DEBUT

ECRIRE("Entrez votre age svp.")

LIRE(age)

AFFICHER("Votre age est : ", age)

FIN

LES PROCEDURE

Un sous programme, sous algorithme, ayant la capacité d'effectuer une tâche indépendante

PROCEDURE

```
PROCEDURE MaProcedure (type param1, type param2, ...)
```

```
    <partie déclaration>
```

```
DEBUT
```

```
    <instructions>
```

```
FIN
```

LES FONCTIONS

Un sous programme, sous algorithme ayant la capacité d'effectuer une tâche indépendante

En math : $y = f(x)$

LES FONCTIONS

Un sous programme, ayant la capacité d'effectuer une tâche indépendante

En math : $y = f(x)$

LES FONCTIONS

Un sous programme, ayant la capacité d'effectuer une tâche indépendante

En math : $y = f(x)$

Une fonction peut avoir des paramètres

Une fonction peut retourner une valeur

FONCTION

FONCTION *CalculerAge*(*ENTIER jour, ENTIER mois, ENTIER ANNEE*): *ENTIER*
ENTIER Val

DEBUT

<zone d'instructions>

RETOURNE val

FIN

FONCTION

ALGORITHME *Exemple*
VARIABLE

INTEGER : age
STRING : str

DEBUT

ECRIRE("Entrez votre age svp.")
LIRE(age)

str ← CalculeAnneeDeNaissance(age)

ECRIRE(str)

FIN

EXERCICES

EXERCICE : ECHANGE DE VARIABLE

Ecrire un programme qui échange la valeur de deux variables.

EXERCICE : CALCUL DU CARRE

Ecrire un programme qui demande un nombre a l'utilisateur,
puis qui calcule et affiche le carré de ce nombre.

EXERCICE : LA CAISSE

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 libellées apparaissent clairement

EXERCICE : LE PRODUIT

Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif, positif ou nul.

V1 : Par le calcul

V2 : Sans calcul

LES STRUCTURES DE CONTRÔLE



Qu'est-ce qu'une structure de contrôle ?

