

Algorithmique de base et python

Cours 01

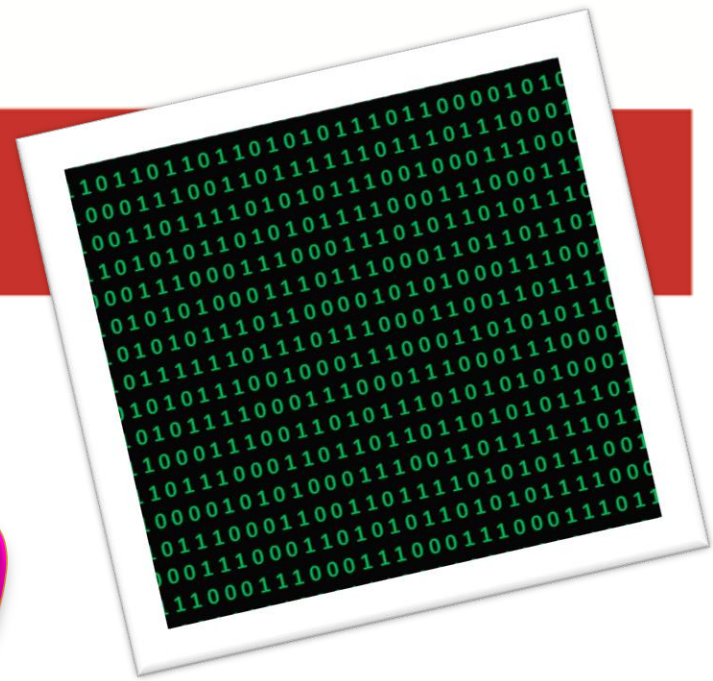
Master G2M

2024-2025

celine.jost@univ-paris8.fr



Les supports du cours sont déposés sur Moodle
après chaque cours.

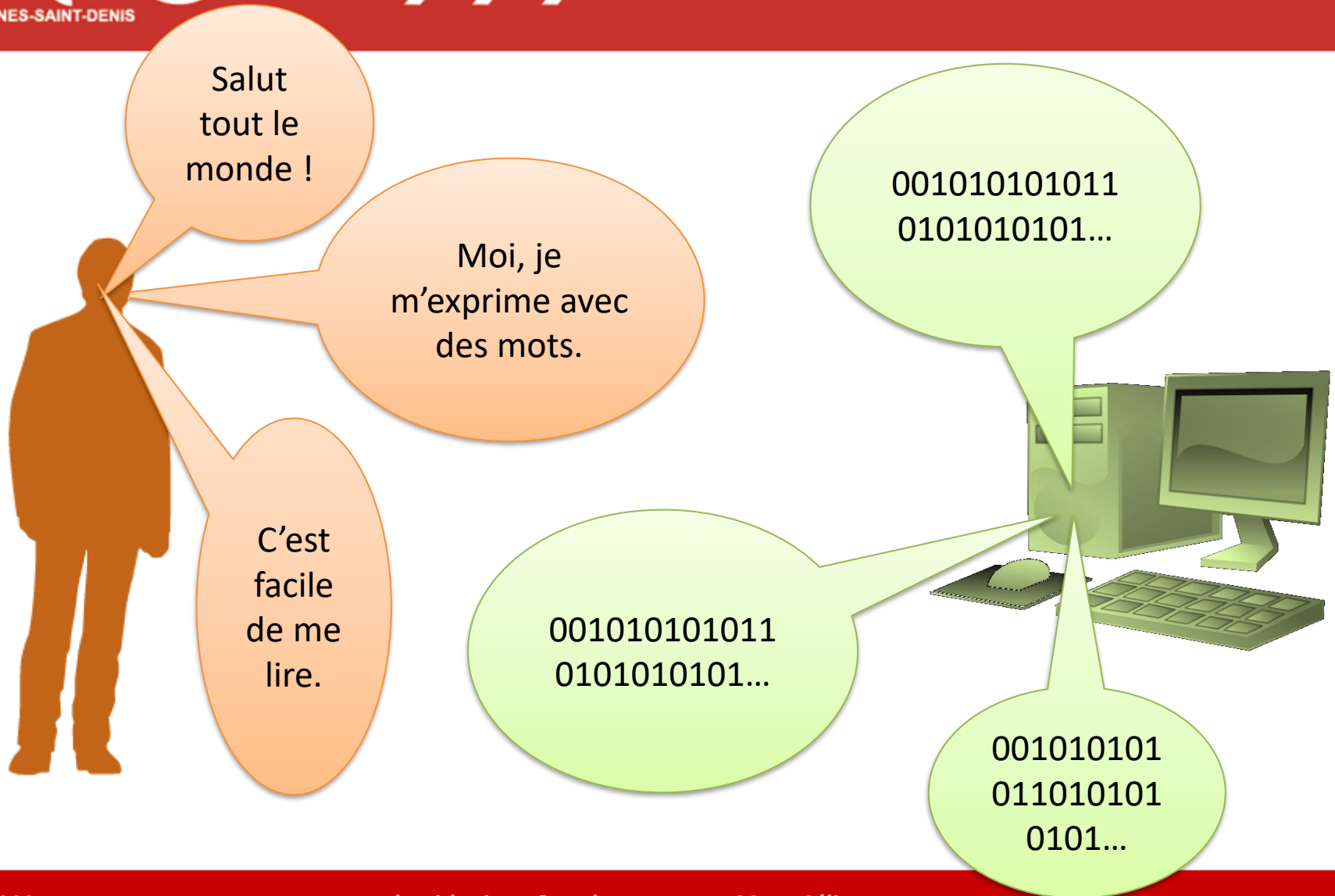


0 0 1 0 1 0 0

POURQUOI L'ALGORITHMIQUE ?

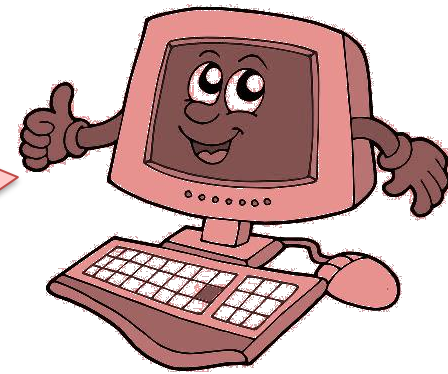


Contexte





Hello, je m'appelle
Algorithmique !
Je suis mi-humain, mi-machine.
Je sais parler le langage des
humains, et le langage de
l'ordinateur. Du coup, c'est
facile pour moi d'exprimer un
problème d'humain à
l'ordinateur et vice-versa.



Ca, c'est l'allégorie
de l'algorithmique.



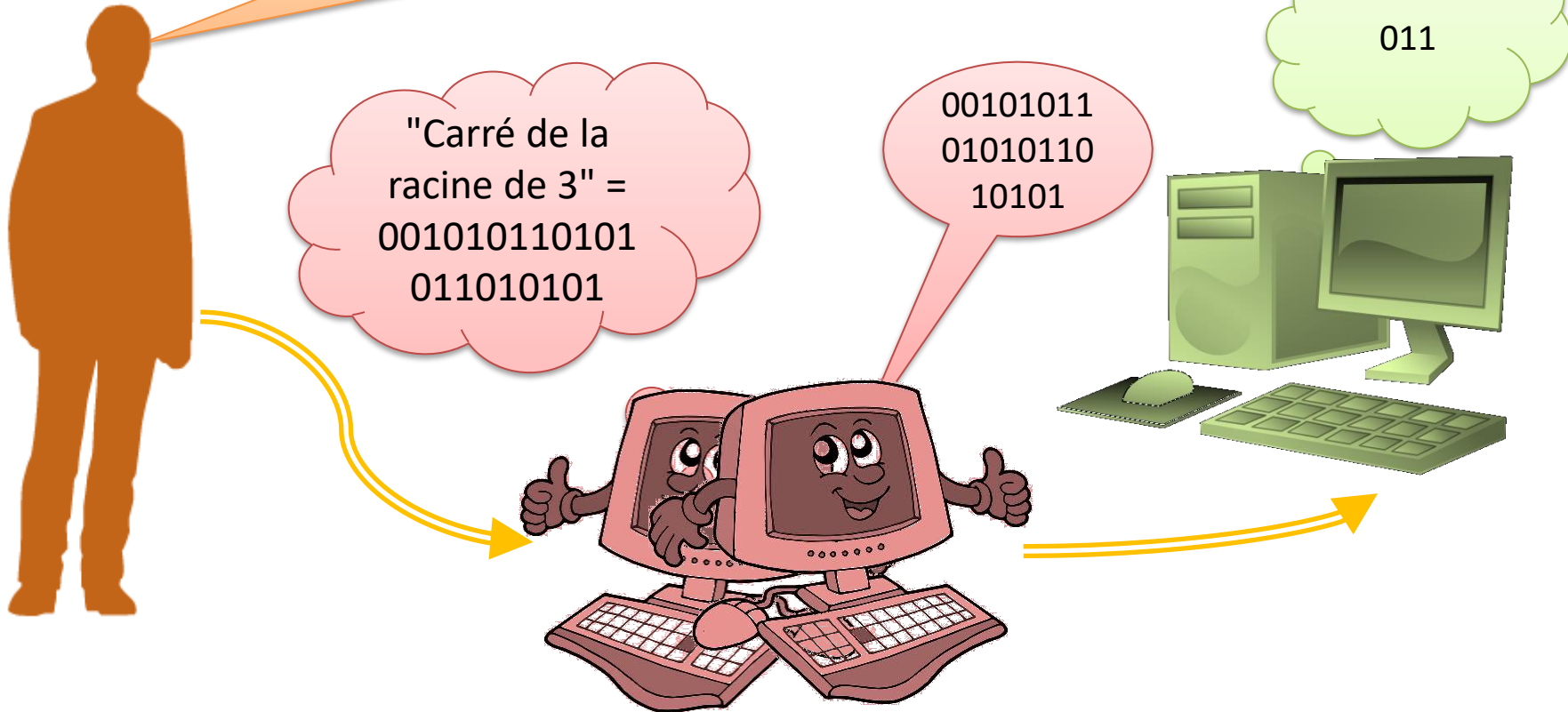
Exemple

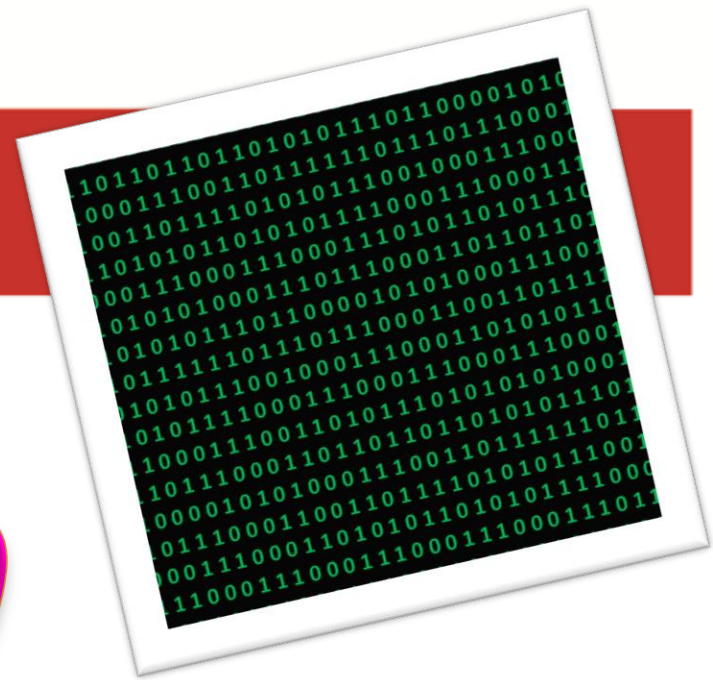
J'aimerais calculer le carré de la racine de 3 !

"Carré de la
racine de 3" =
001010110101
011010101

00101011
01010110
10101

011





0 0 1 0 1 0 0

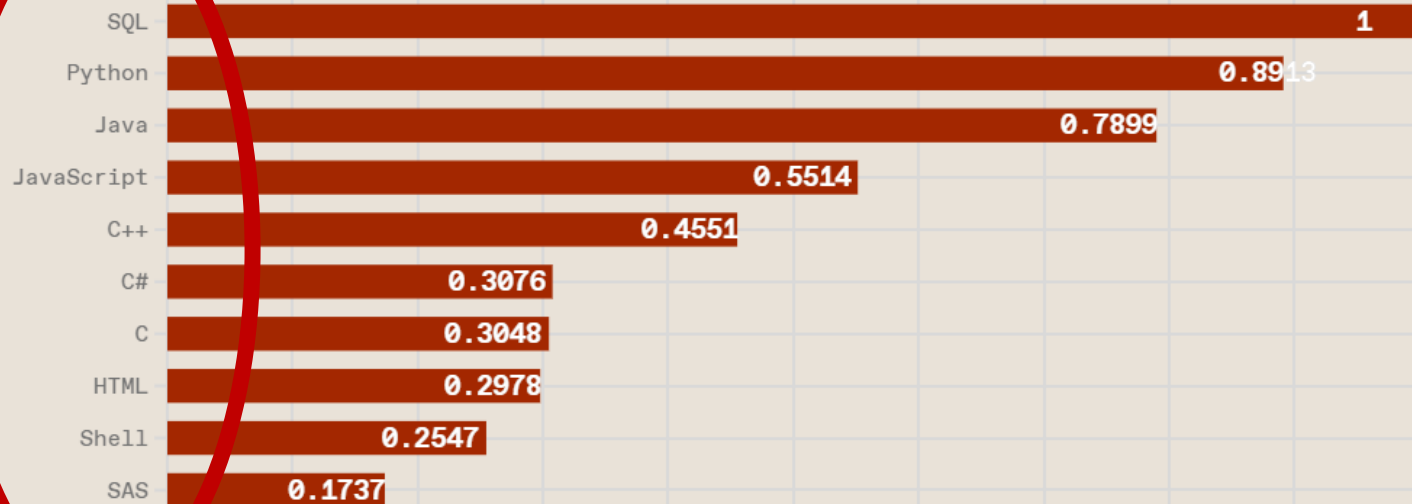
L'ALGORITHMIQUE N'EST PAS UN LANGAGE DE PROGRAMMATION



Top Programming Languages 2023

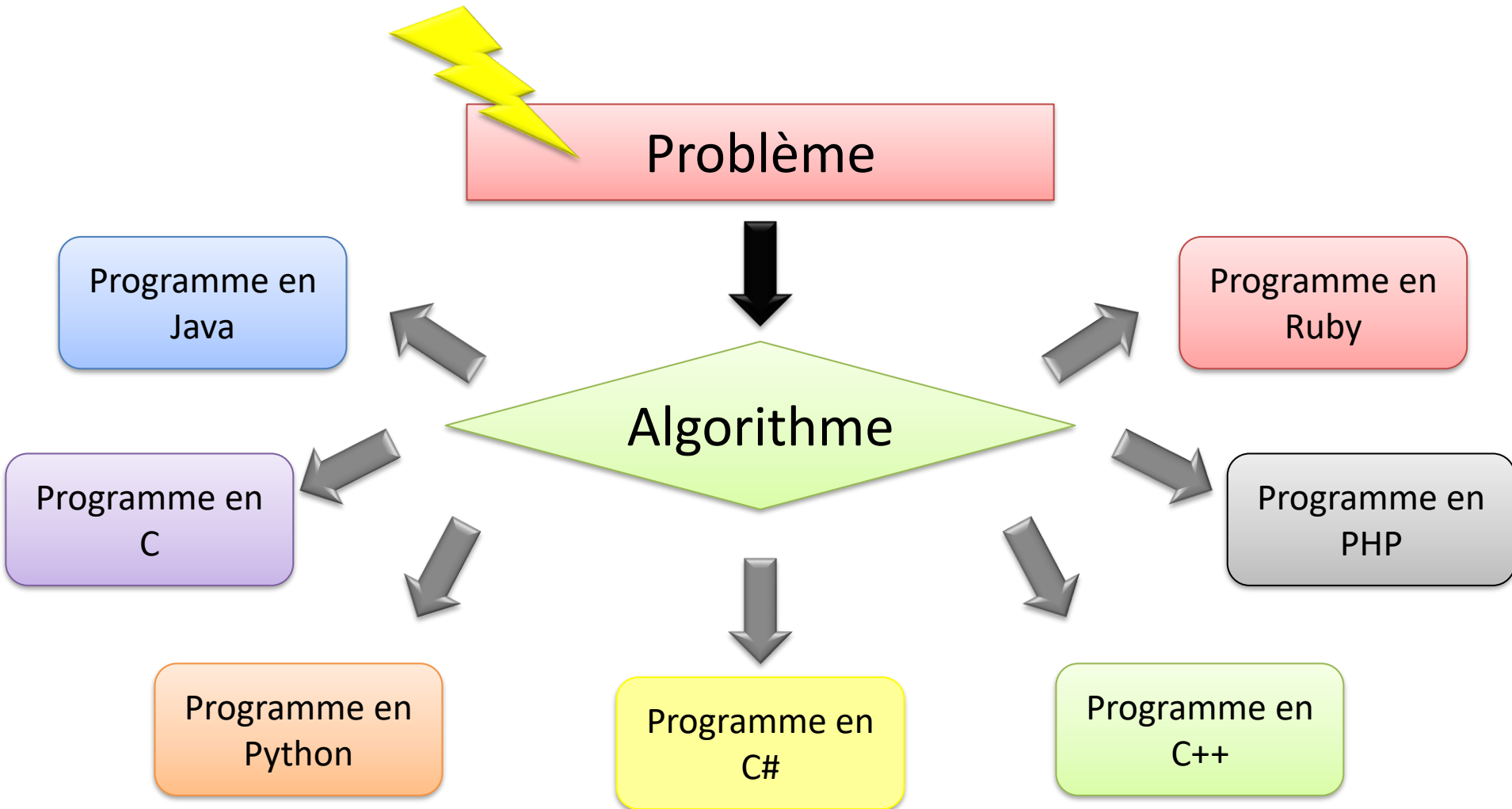
Click a button to see a differently weighted ranking

Spectrum **Jobs** Trending



Top 10 des langages les plus demandés par les employeurs

Source : <https://spectrum.ieee.org/the-top-programming-languages-2023>



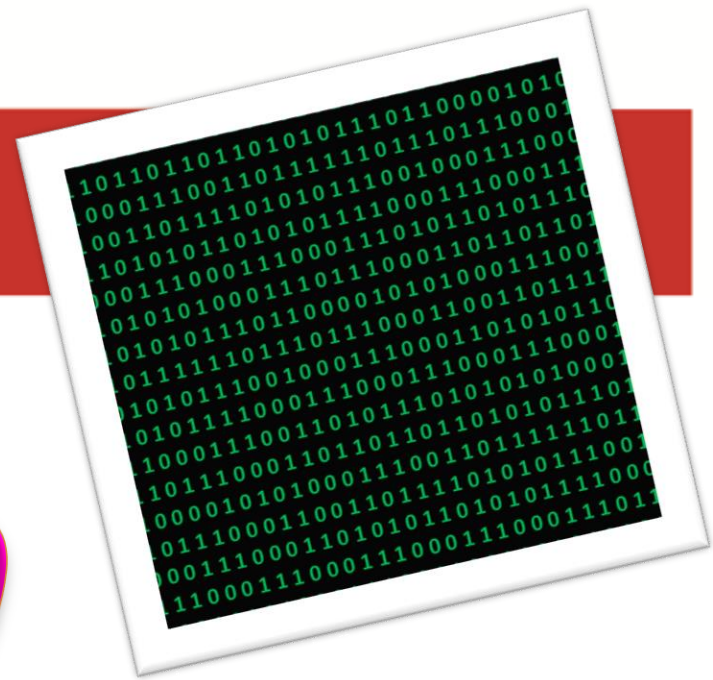


Algorithmique

- Fabrique des algorithmes
- Compréhensible par l'humain
- Langage non formel
- Créé à l'aide du cerveau sur papier
- Exécution simulée à l'aide du cerveau
- Indépendant des langages de programmation

Langage de programmation

- Code les algorithmes
- Compréhensible par la machine
- Langage formel
- Créé à l'aide de l'ordinateur
- Exécution directement par la machine
- ... C'est un langage de programmation !



0 0 1 0 1 0 0

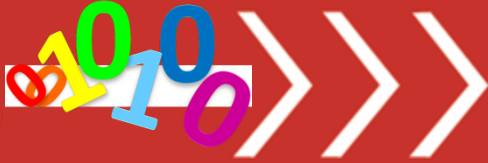
OBJECTIFS DU COURS



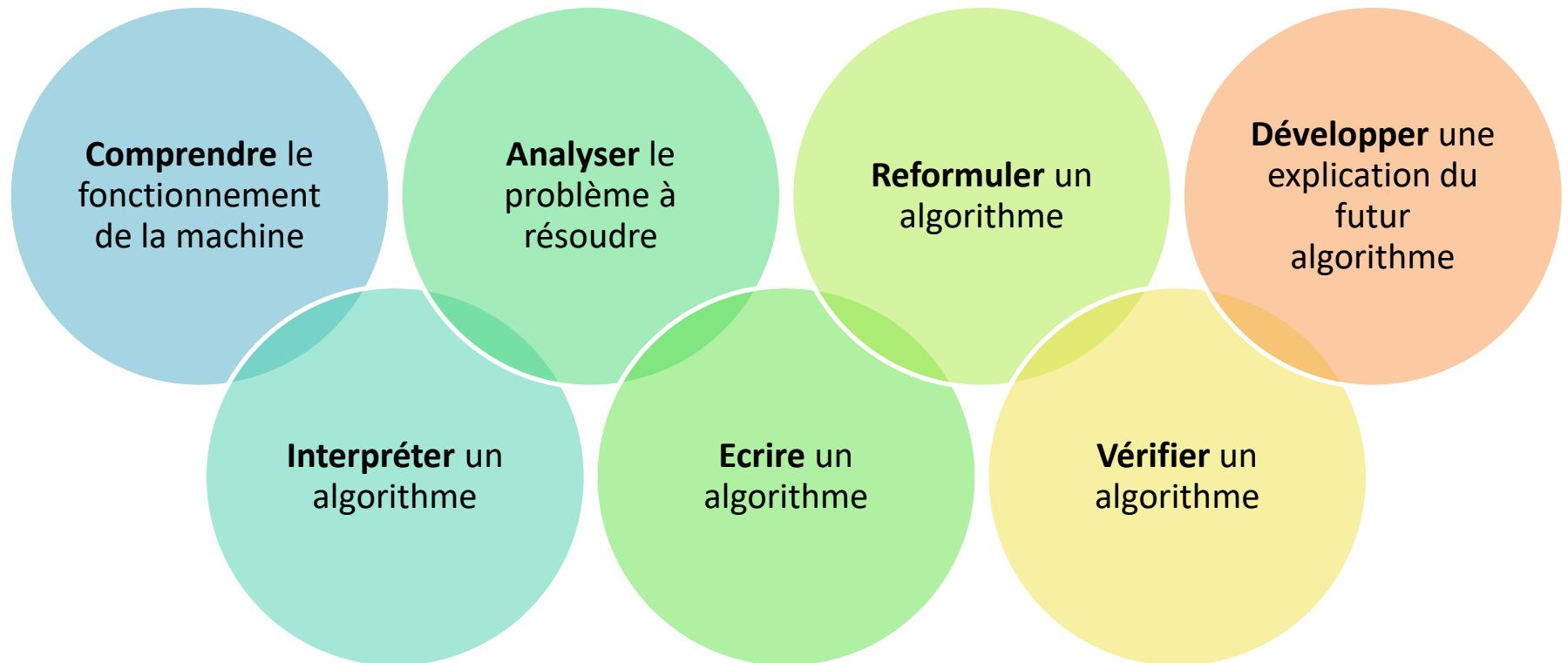
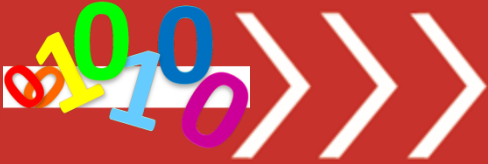
Deux objectifs principaux

Développer les
savoir-faire
indispensables

Comprendre
et utiliser les
concepts principaux
de l'algorithmique



- *Constante*
- *Variable*
- *Opérateur*
- *Tableau (liste, pile)*
- *Arbre*
- **Séquence**
- *Condition*
- *Itération*
- *Fonction*
- *Procédure*
- *Permutation*
- *Concaténation*
- *Tri*
- *Déclaration*
- *Affectation*
- *Initialisation*
- *Incrémentation*
- *Expression*
- **Instruction**
- *Bloc d'instructions*
- *Lecture*
- *Ecriture*
- *Imbrication*
- *Récursivité*
- *Complexité*



Ce n'est pas une définition officielle. Cela constitue mes choix pédagogiques.



Comprendre le fonctionnement de la machine

Créer un algorithme

Aanalyser
Développer
Ecrire
Interpréter
Vérifier

Modifier un algorithme

Reformuler
Interpréter
Vérifier

Ce n'est pas une définition officielle. Cela constitue mes choix pédagogiques.



Les algorithmes sont prévus pour être interprétés par des ordinateurs.

Ils se basent donc sur le fonctionnement de la machine.

Donc, pour écrire des algorithmes, il faut avoir un minimum d'information sur la machine.



Analyse du problème

Comment font les experts ?

Que devra faire le système informatique ?



Développement d'une explication qui s'appuie sur l'analyse.

On transmet souvent les algorithmes à l'oral en les expliquant.

Comment on explique à une personne tout ce que fera le système informatique, étape par étape ?



Comment traduire le développement dans un langage informatique ?



Quel résultat produit chaque ligne de l'algorithme ?

Ici, il s'agit de faire. On construit le résultat tel que le ferait le système informatique.



Ceci est l'étape la plus importante !

Comment savoir si l'algorithme est correct (ou pas) ?

Il ne faut pas livrer un algorithme qui risque d'induire des bugs !



Manière de lire un algorithme.

On explique chaque ligne de l'algorithme
comme si on le traduisait à quelqu'un qui ne
connait pas l'algorithmique.

Ce savoir-faire est indispensable pour pouvoir
Interpréter.



12

Séances de
cours de 3h

7

Savoir-faire
indispensable
pour faire de
l'algorithmique

25

Concepts
absolument
indispensables
(et il y en a
d'autres)

Il y a du boulot !

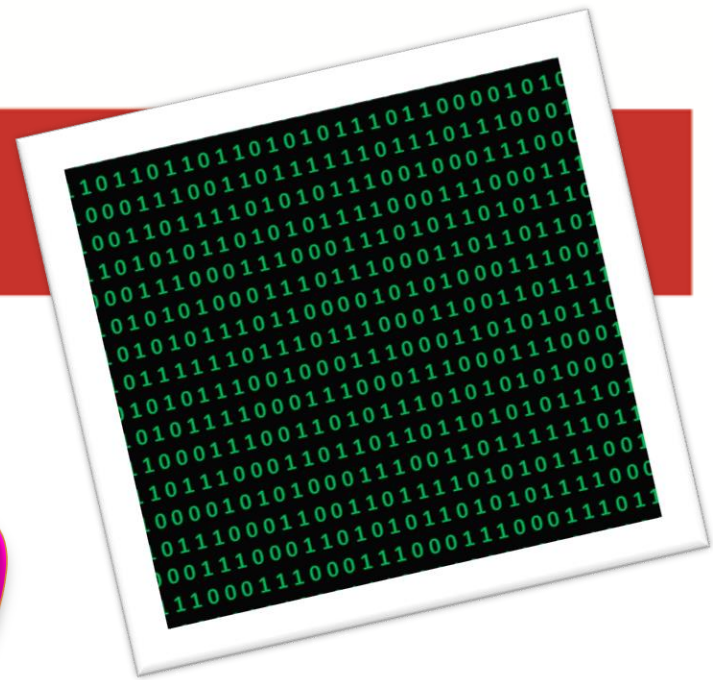


- Constante
- Variable
- Opérateur
- Tableau (liste, pile)
- Arbre
- Séquence
- Condition
- Itération
- Fonction
- Procédure
- Permutation
- Concaténation
- Tri
- Déclaration
- Affectation
- Initialisation
- Incrémentation
- Expression
- Instruction
- Bloc d'instructions
- Lecture
- Ecriture
- Imbrication
- Récursivité
- Complexité



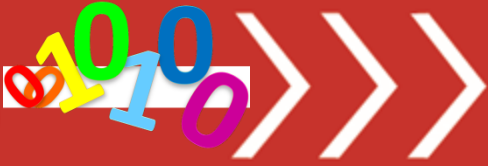
- L'algorithme est une façon de raisonner
- Cela se passe dans vos cerveaux
- Et avec du papier et un crayon !
- Vous rédigerez beaucoup





0 0 1 0 1 0 0

DÉFINITIONS FORMELLES



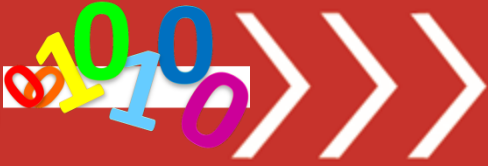
Algorithmique : résolution d'un problème par la production d'**un** ou plusieurs algorithmes.

(Définition non officielle)

Problème
simple

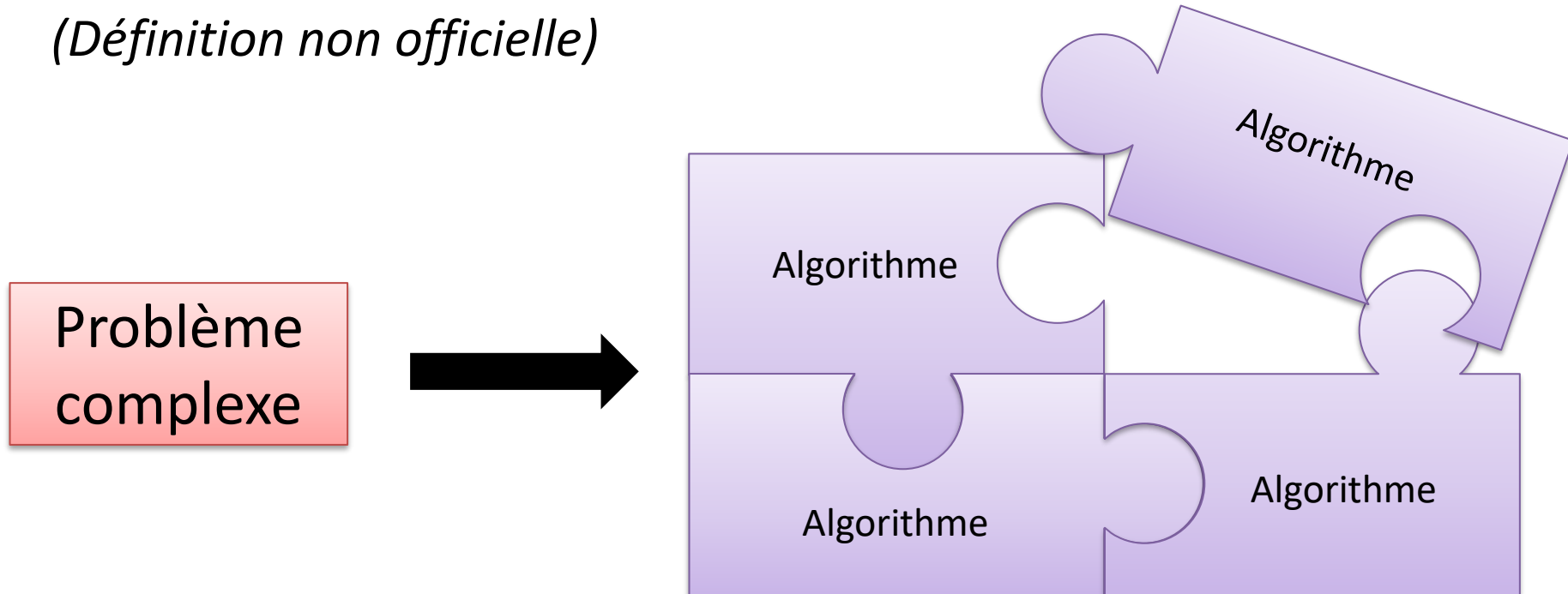


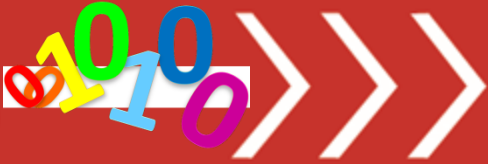
Algorithme



Algorithmique : résolution d'un problème par la production d'un ou **plusieurs** algorithmes.

(Définition non officielle)





Algorithme : suite d'**instructions** qui, appliquée à un **ensemble de données**, conduit à un **résultat**. (*Définition plutôt officielle*)

Instructions

$$x = 3a + b$$

$$y = 2x + c$$

Données

$$a = 2$$

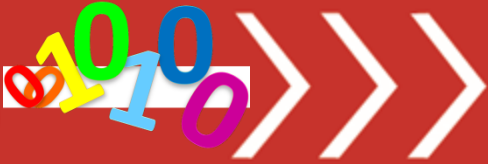
$$b = 3$$

$$c = 4$$

Résultat

$$x = 9$$

$$y = 22$$



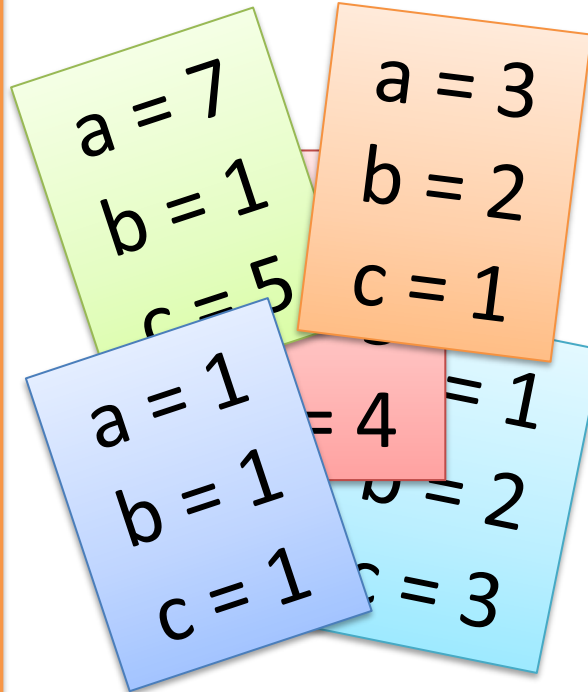
Exemples

Instructions

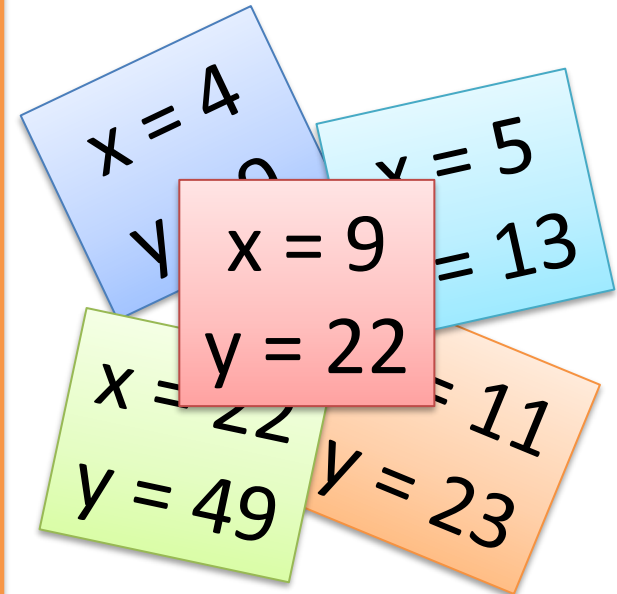
$$x = 3a + b$$

$$y = 2x + c$$

Données

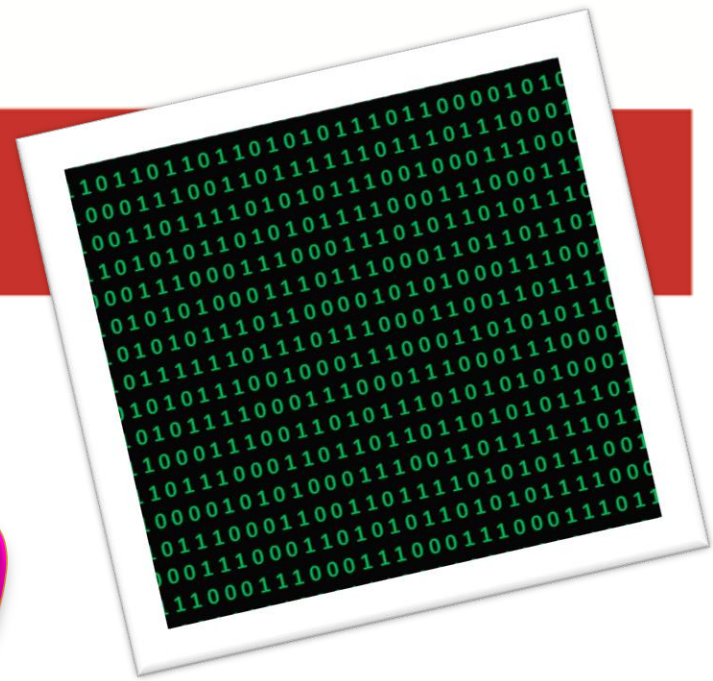


Résultat





- Algorithme : suite d'instructions qui, appliquée à un ensemble de données, conduit à un résultat.
- Algorithme : raisonnement **général** que l'on peut appliquer à différents cas.
- Mais c'est aussi : une manière de penser !
- Indépendant des langages de programmation
- Compréhensible pour l'humain
- Compréhensible pour la machine
- Demande beaucoup de temps de pratique !



0 0 1 0 1 0 0

Comprendre

QU'EST-CE QU'UN ALGORITHME ?



Exemple 1/3



MOUSSE AU CHOCOLAT VEGAN



Une mousse au chocolat vegan sans oeufs

Imprimer la recette

Temps de
préparation
20 minutes

Temps total
20 minutes

Portions
4

Ingrédients

- 220 gr de jus de pois chiche
- 150 gr de chocolat noir ou au lait

Instructions

1. Faire fondre le chocolat.
2. Monter le jus de pois chiche en neige.
3. Incorporer petit à petit le jus de pois chiche en neige au chocolat.
4. Mélanger délicatement avec une maryse.

Source : <https://www.lolibox.fr/mousse-chocolat-vegan/>



1. Garer la voiture dans l'espace réservé
2. Fermer les vitres et retirer l'antenne
3. Sélectionner le programme désiré
4. Mettre le jeton dans la machine
5. Sauter dans sa voiture, s'installer confortablement avec pop-corn et coca, et profiter du spectacle pendant 6 minutes.



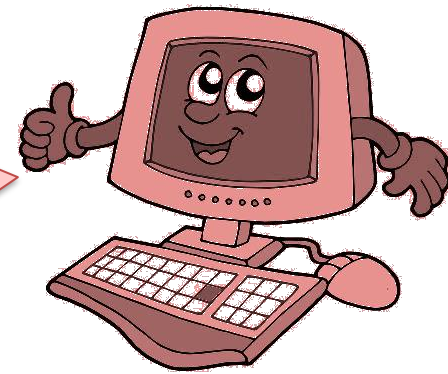
Le jeu commence.

- Etape 1 : le joueur 1 choisit un nombre secret compris entre 1 et 1000.
- Etape 2 : le joueur 2 propose un nombre.
 - Si le nombre proposé est plus petit que le nombre secret, le joueur 1 répond « C'est plus ».
 - Si le nombre proposé est plus grand que le nombre secret, le joueur 1 répond « C'est moins ».
 - Si le nombre proposé est égal au nombre secret, le joueur 1 répond « C'est gagné » et le jeu s'arrête.

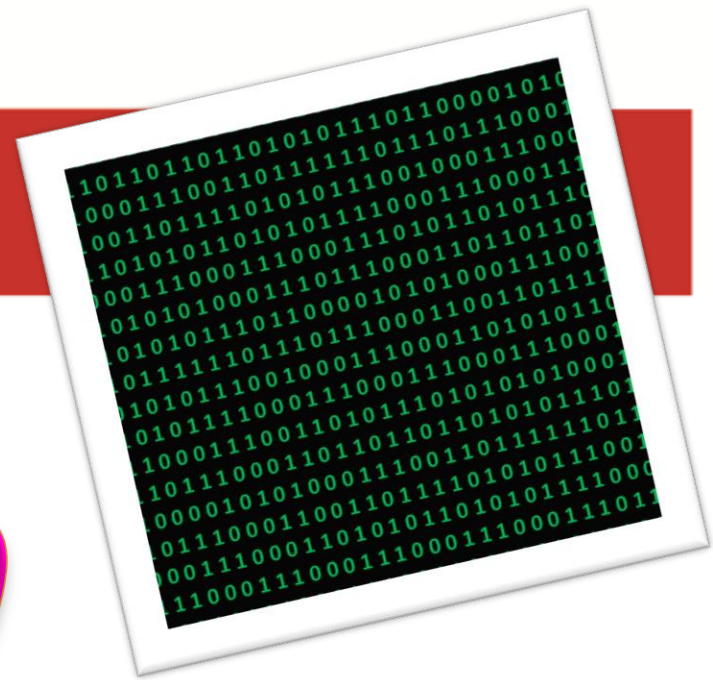
L'étape 2 se répète jusqu'à ce que le joueur 2 gagne le jeu.



Vous connaissez déjà plein d'algorithmes. Le défi maintenant consiste simplement à les écrire de manière optimisée pour l'ordinateur !
Je* suis là pour vous y aider 😊



*Ca, c'est l'allégorie de l'algorithmique.



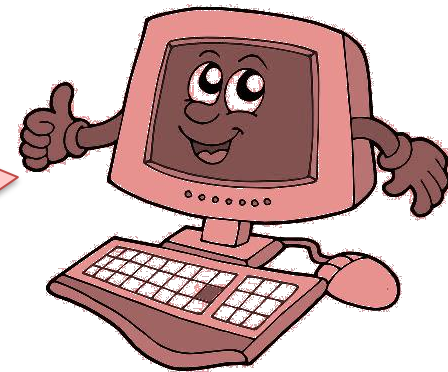
0 0 1 0 1 0 0

COMMENT ÉCRIRE UN ALGORITHME ?



L'algorithme d'Euclide

Euclide est un mathématicien qui a vécu vers -300 avant J.C et qui a créé un algorithme permettant de calculer le PGCD de deux nombres. Regardons sur Internet les différentes représentations de son algorithme.

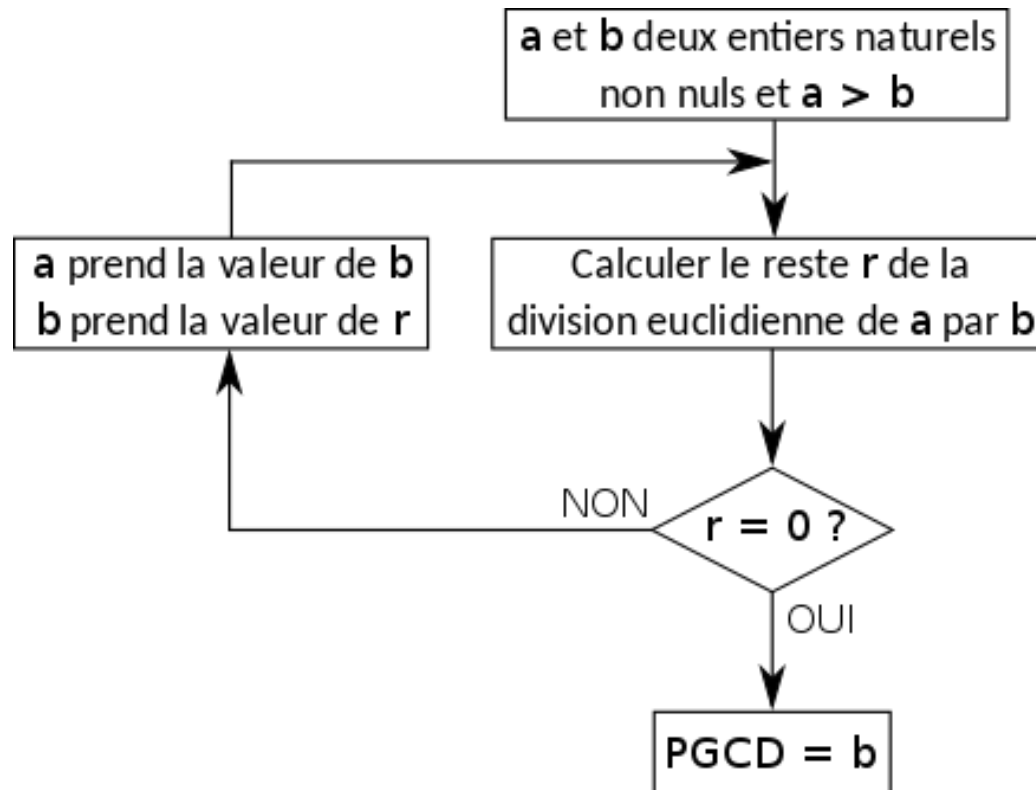




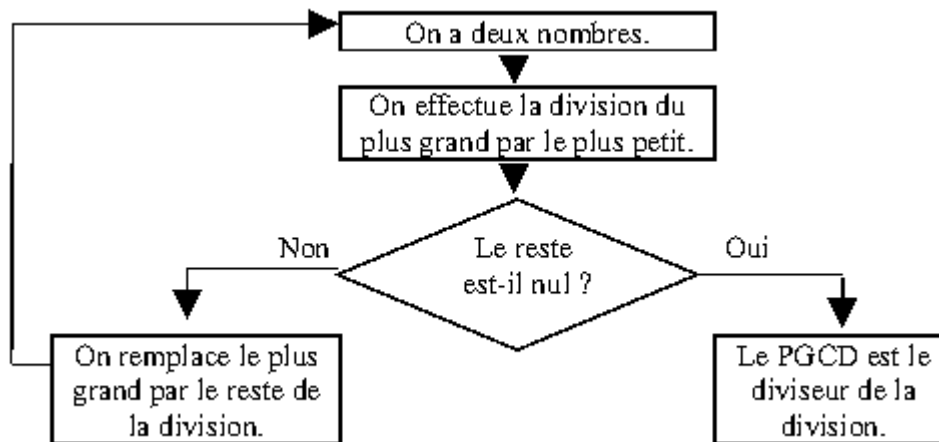
- Plus Grand Diviseur Commun
- Exemple :
 - PGCD de 15 et 25 vaut 5
 - PGCD de 24 et 38 vaut 2
 - PGCD de 1686 et 936 vaut 6
 - PGCD de 50 et 250 vaut 50
 - PGCD de 936 et 450 vaut 18
 - PGCD de 936 et 451 vaut 1



Algorigramme



Source 2019 : Wikipedia



Source 2019 : <http://tpe-ia2014-2015.e-monsite.com/pages/1-la-robotisation/les-algorithmes.html>



Variables :

a est un nombre entier

b est un nombre entier

r est un nombre entier

Début algorithme

Lire a

Lire b

Tant que $b > 0$

reste de $a \text{ div } b \rightarrow r$

$b \rightarrow a$

$r \rightarrow b$

Fin de Tant que

Afficher a

Fin algorithme



Code de l'algorithme

```
1  VARIABLES
2    A EST_DU_TYPE NOMBRE
3    B EST_DU_TYPE NOMBRE
4    R EST_DU_TYPE NOMBRE
5  DEBUT_ALGORITHME
6    AFFICHER "Ce programme va calculer le PGCD de deux entiers entrés par l'utilisateur."
7    AFFICHER "Entrez le premier des deux entiers : "
8    LIRE A
9    AFFICHER "Entrez le second des deux entiers "
10   LIRE B
11   SI (A<B) ALORS
12     DEBUT_SI
13       R PREND_LA_VALEUR A
14       A PREND_LA_VALEUR B
15       B PREND_LA_VALEUR R
16     FIN_SI
17   R PREND_LA_VALEUR A%B
18   TANT_QUE (R!=0) FAIRE
19     DEBUT_TANT_QUE
20       A PREND_LA_VALEUR B
21       B PREND_LA_VALEUR R
22       R PREND_LA_VALEUR A%B
23     FIN_TANT_QUE
24   AFFICHER "Le PGCD des deux nombres que vous avez entrés est : "
25   AFFICHER B
26  FIN_ALGORITHME
```



Et encore du texte

```
TANT_QUE (r != 0) FAIRE
  DEBUT_TANT_QUE
    q PREND_LA_VALEUR floor(a/b)
    r PREND_LA_VALEUR a%b
    a PREND_LA_VALEUR b
    b PREND_LA_VALEUR r
  FIN_TANT_QUE
```

Source 2019 : <http://culturemath.ens.fr/content/lalgorithme-deuclide>



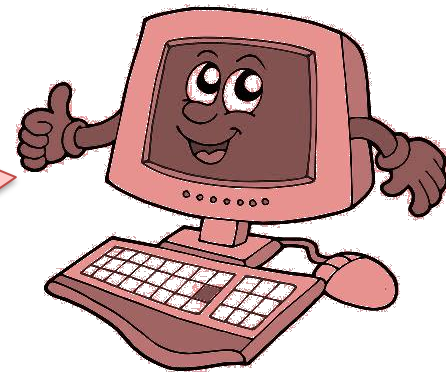
Logiciel SCRATCH



Source 2019 : <http://slideplayer.fr/slide/3704454/>



Il y a beaucoup de façons
d'écrire un algorithme.
Ce qui est important, c'est la
logique.





Une instruction est un ordre simple.

Par exemple :

- Mélanger délicatement
- Fermer les vitres
- Joueur 1 choisit un nombre
- Calculer le reste de la division euclidienne de a par b



Cela veut dire que les actions sont exécutées les unes après les autres.

C'est un **ordre séquentiel**.



- *Constante*
- *Variable*
- *Opérateur*
- *Tableau (liste, pile)*
- *Arbre*
- **Séquence**
- *Condition*
- *Itération*
- *Fonction*
- *Procédure*
- *Permutation*
- *Concaténation*
- *Tri*
- *Déclaration*
- *Affectation*
- *Initialisation*
- *Incrémentation*
- *Expression*
- **Instruction**
- *Bloc d'instructions*
- *Lecture*
- *Ecriture*
- *Imbrication*
- *Récursivité*
- *Complexité*



Ecrire l'algorithme de la fabrication d'une omelette.

