UNIVERSITÉ PARIS 8
VINCENNES-SAINT-DENIS

Algorithmique de base et python

Cours 03

Master G2M
2024-2025

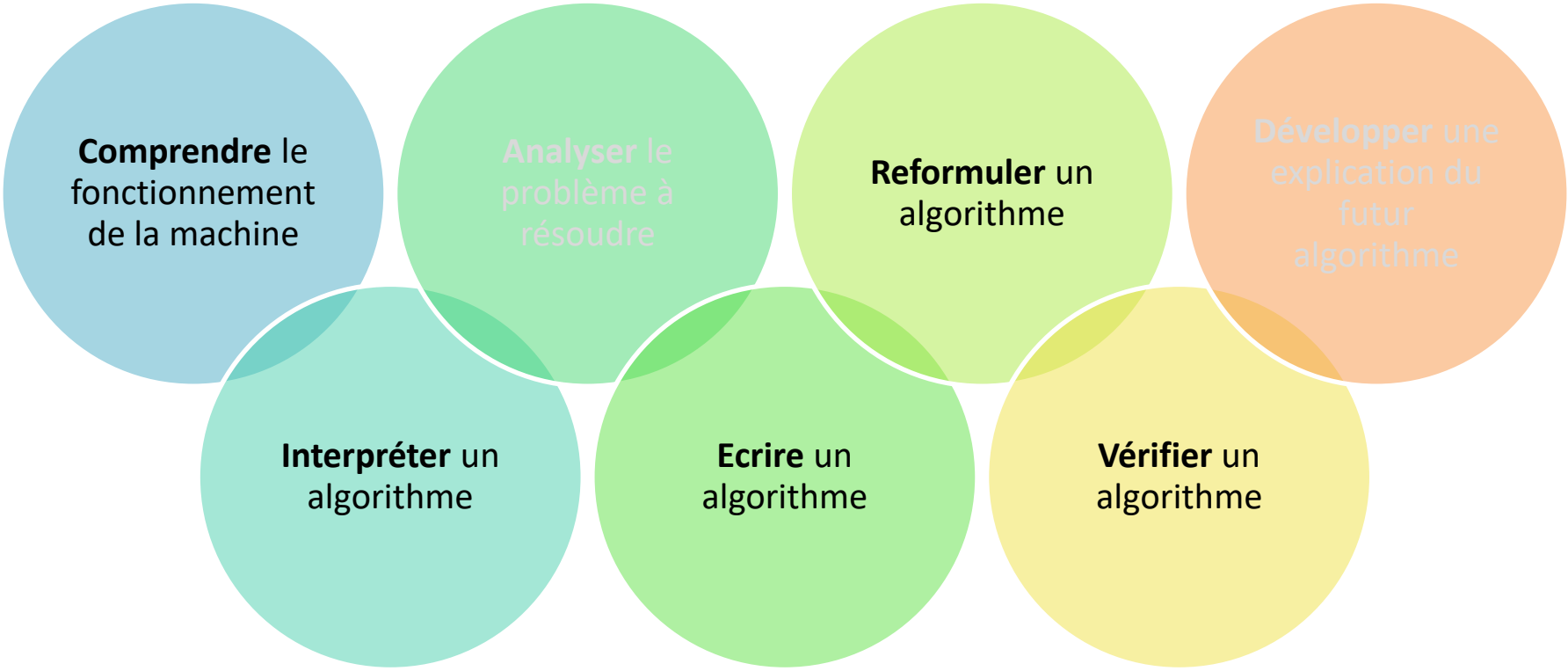
celine.jost@univ-paris8.fr

Comprendre le fonctionnement de la machine

Reformuler, interpréter, vérifier, écrire

Opérateur
Variable
Constante
Déclaration
Affectation
Initialisation

Les savoir-faire



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

Les concepts

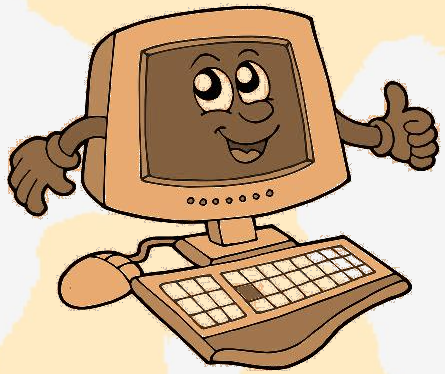
- Constante
- Variable
- Opérateur
- *Tableau (liste, pile)*
- *Arbre*
- ~~Séquence~~
- *Condition*
- *Itération*
- *Fonction*
- *Procédure*
- ~~Permutation~~
- *Concaténation*
- *Tri*

- Déclaration
- Affectation
- Initialisation
- *Incrémentation*
- *Expression*
- ~~Instruction~~
- *Bloc d'instructions*
- *Lecture*
- *Ecriture*
- *Imbrication*
- *Récursivité*
- *Complexité*

Nos outils

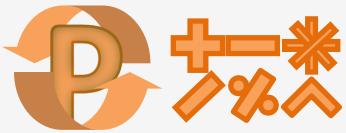
INVENTAIRE

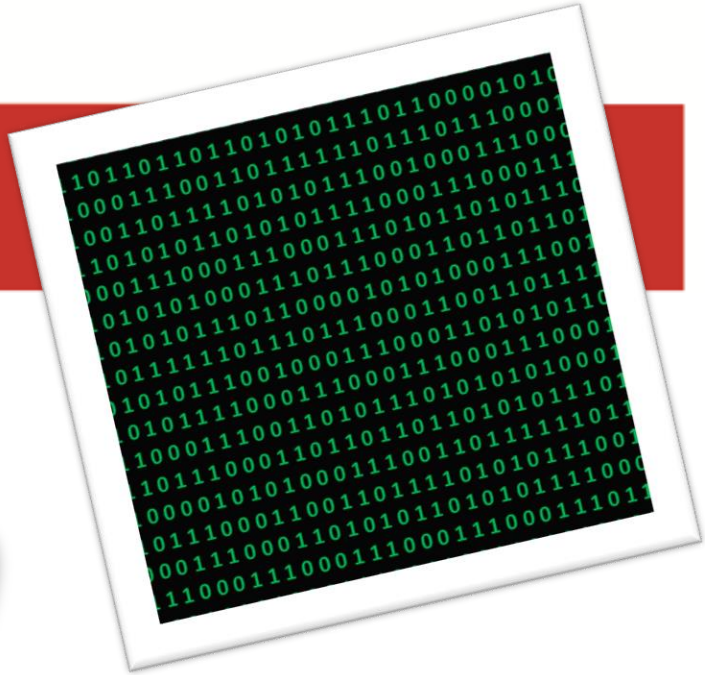
AVATAR



NOM : *Algorithmique*

SAC A DOS





Nouvel outil hautement important !

LA MÉMOIRE

Contexte

- L'ordinateur utilise une mémoire
- Pour faire une opération, il faut lui donner toutes les informations.

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

Contexte

Contexte

Combien ai-je de gâteaux ?

Contexte

Contexte

Contexte

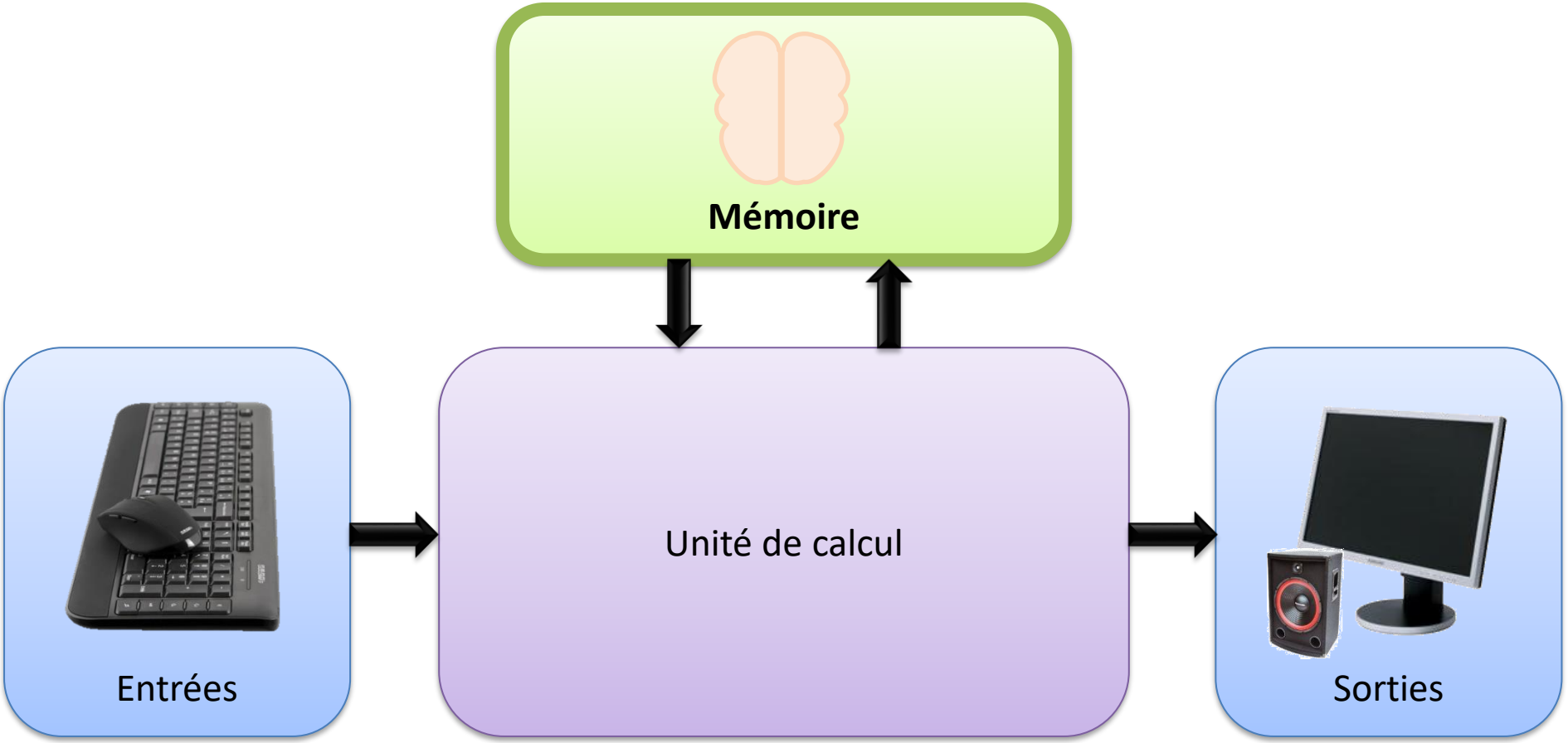
Combien ai-je de gâteaux ?

Contexte

5

Combien ai-je de gâteaux ?

Principe de base



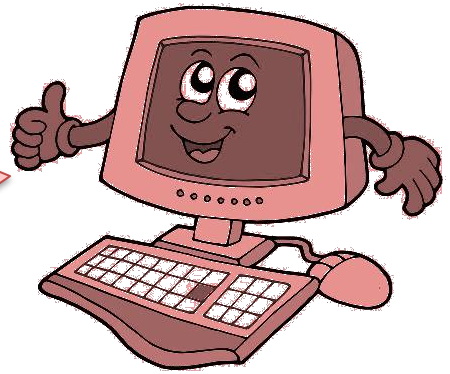
Exemple : la factorielle

Prenons l'exemple de la factorielle.

Rappel :

Soit n un entier naturel. La factorielle de n est définie par

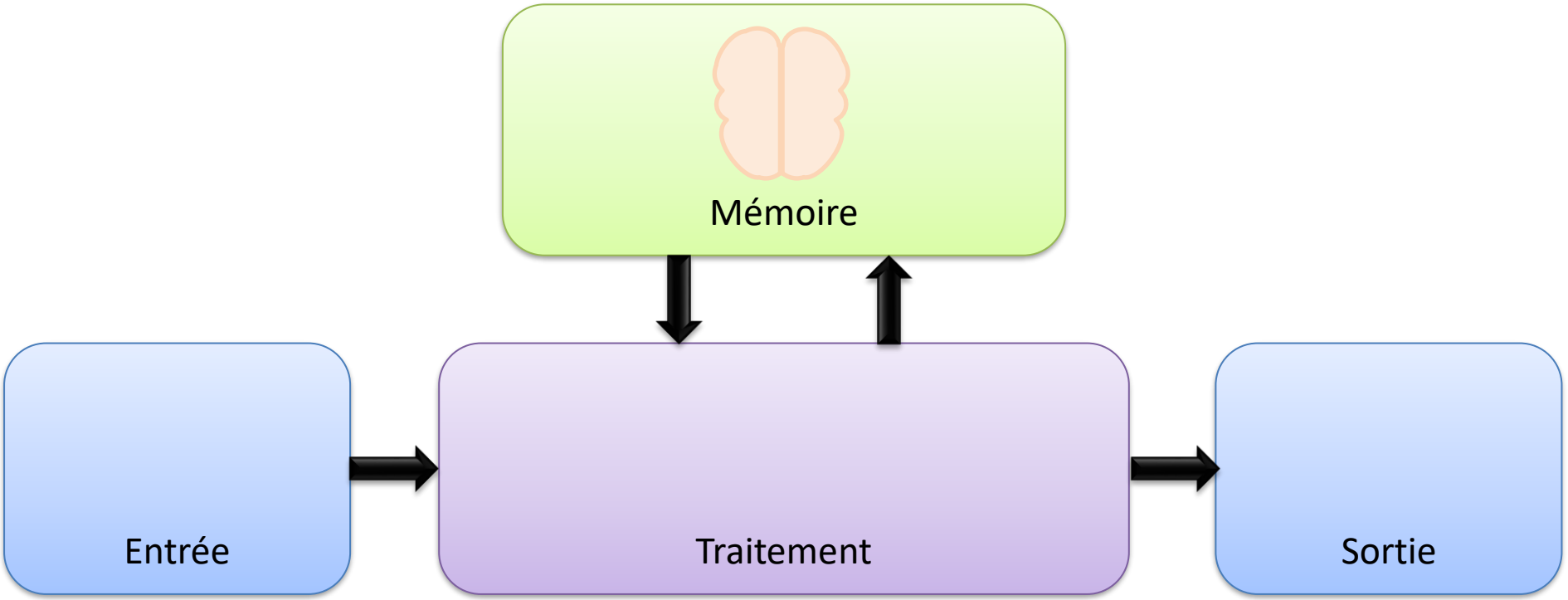
n! = 1 x 2 x 3 x … x (n-1) x n



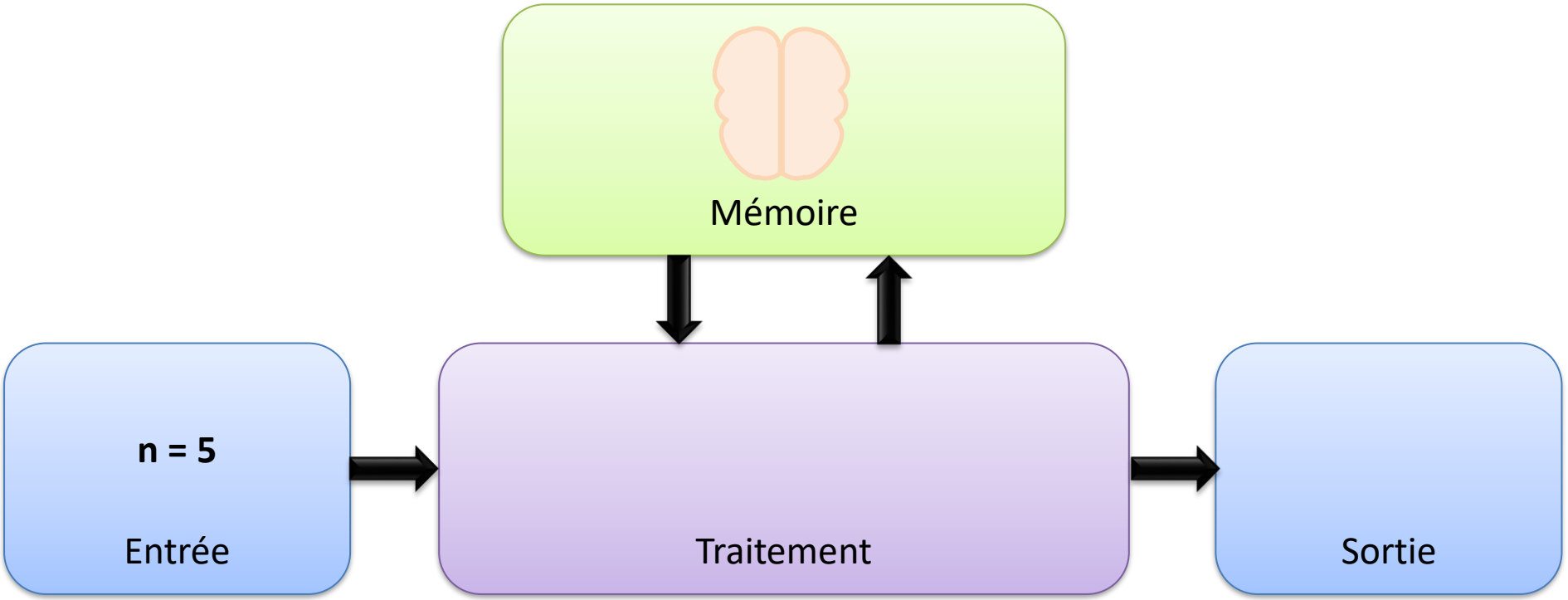
Calcul à effectuer

$$1 \times 2 \times 3 \times \ldots \times (n-1) \times n$$

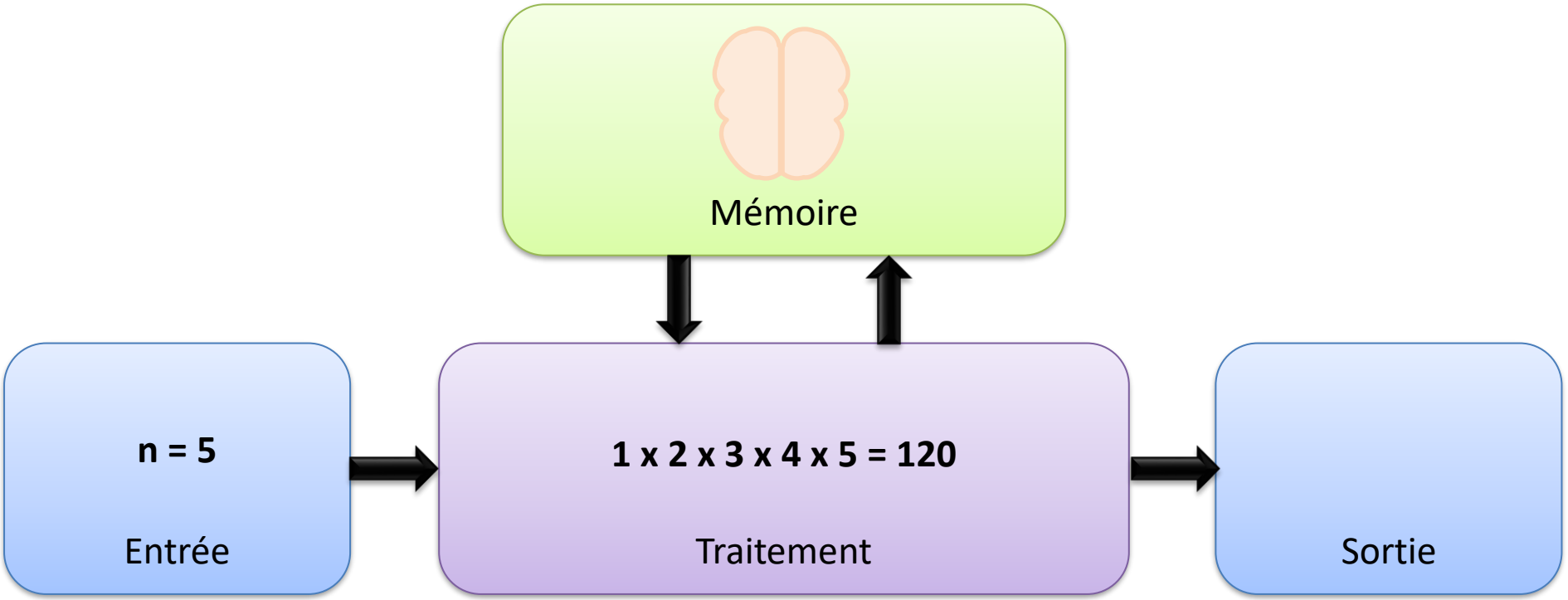
Exécution de l'algorithme



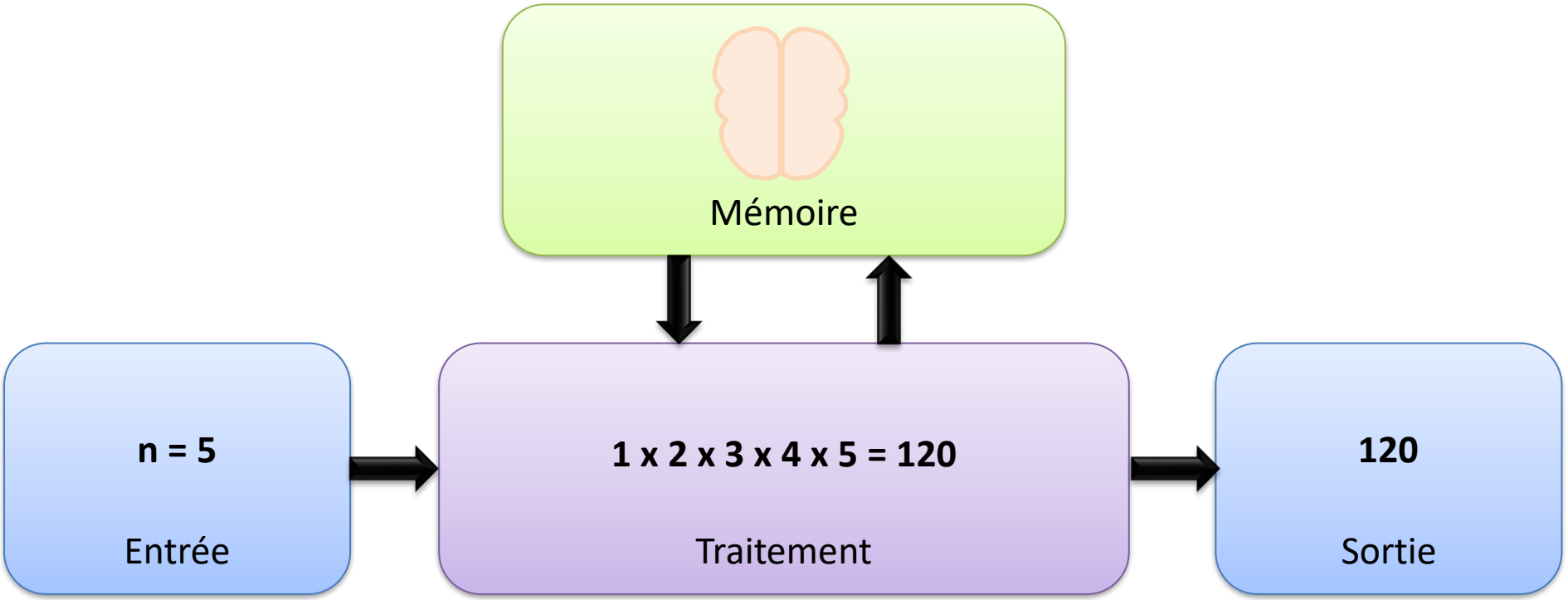
Exécution de l'algorithme

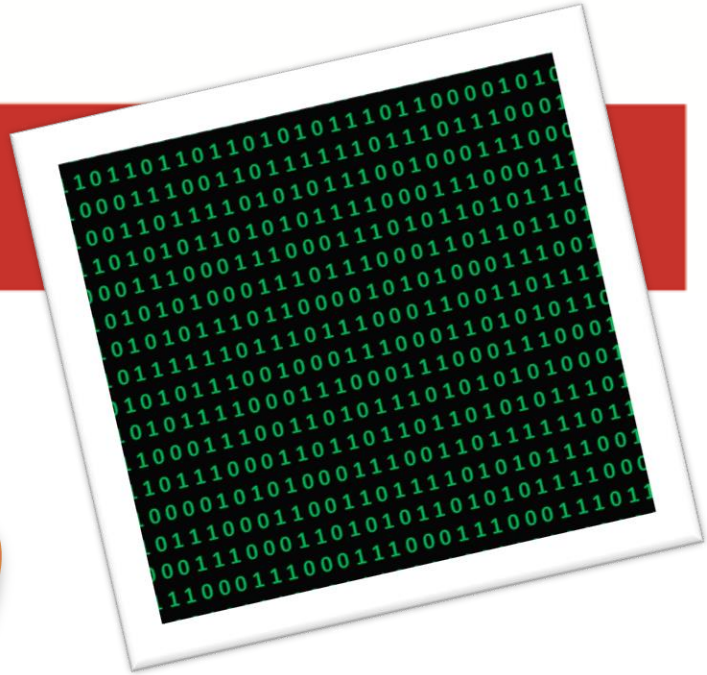


Exécution de l'algorithme



Exécution de l'algorithme





COMMENT ÇA SE PASSE EN VRAI ?

Deux choses à comprendre :

- L'ordinateur effectue un calcul à la fois
- L'ordinateur doit utiliser sa mémoire pour retrouver les valeurs à calculer

Ce qui pourrait se passer…

Calcul à effectuer : 1 x 2 x 3 x 4 x 5

1 2 3 4 5

Ce qui pourrait se passer…

Calcul à effectuer : 1 x 2 x 3 x 4 x 5

1 2 3 4 5

1 x 2 = 2

Ce qui pourrait se passer…

Calcul à effectuer : 1 x 2 x 3 x 4 x 5

1 x 2 = 2

1 2 3 4 5

1 2 3 4 5 2

Ce qui pourrait se passer…

Calcul à effectuer : 1 x 2 x 3 x 4 x 5

1 x 2 = 2

2 x 3 = 6

1 2 3 4 5

1 2 3 4 5 2

Ce qui pourrait se passer…

Calcul à effectuer : 1 x 2 x 3 x 4 x 5

1 x 2 = 2

2 x 3 = 6

1 2 3 4 5

1 2 3 4 5 2

1 2 3 4 5 2 6

Ce qui pourrait se passer…

Calcul à effectuer : 1 x 2 x 3 x 4 x 5

1 x 2 = 2

2 x 3 = 6

6 x 4 = 24

1 2 3 4 5

1 2 3 4 5 2

1 2 3 4 5 2 6

Ce qui pourrait se passer…

Calcul à effectuer : 1 x 2 x 3 x 4 x 5

1 x 2 = 2

2 x 3 = 6

6 x 4 = 24

1 2 3 4 5

1 2 3 4 5 2

1 2 3 4 5 2 6

1 2 3 4 5 2 6 24

Ce qui pourrait se passer…

Calcul à effectuer : 1 x 2 x 3 x 4 x 5

1 x 2 = 2

2 x 3 = 6

6 x 4 = 24

24 x 5= 120

1 2 3 4 5

1 2 3 4 5 2

1 2 3 4 5 2 6

1 2 3 4 5 2 6 24

Ce qui pourrait se passer…

Calcul à effectuer : 1 x 2 x 3 x 4 x 5

1 x 2 = 2

2 x 3 = 6

6 x 4 = 24

24 x 5= 120


1 2 3 4 5


1 2 3 4 5 2


1 2 3 4 5 2 6


1 2 3 4 5 2 6 24


1 2 3 4 5 2 6 24 120

Autre façon de calculer

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

5 4

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

5 4

5 x 4 = 20

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

5 4

5 x 4 = 20

20 4

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

5 4

5 x 4 = 20

20 4

4 - 1 = 3

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

5 4

5 x 4 = 20

20 4

4 - 1 = 3

20 3

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

5 4

5 x 4 = 20

20 4

4 - 1 = 3

20 3

20 x 3 = 60

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

5 4

5 x 4 = 20

20 4

4 - 1 = 3

20 3

20 x 3 = 60

60 3

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

5 4

5 x 4 = 20

20 4

4 - 1 = 3

20 3

20 x 3 = 60

60 3

3 - 1 = 2

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

5 4

5 x 4 = 20

20 4

4 - 1 = 3

20 3

20 x 3 = 60

60 3

3 - 1 = 2

60 2

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

5 4

5 x 4 = 20

20 4

4 - 1 = 3

20 3

20 x 3 = 60

60 3

3 - 1 = 2

60 2

60 x 2 = 120

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

5 4

5 x 4 = 20

20 4

4 - 1 = 3

20 3

20 x 3 = 60

60 3

3 - 1 = 2

60 2

60 x 2 = 120

120 2

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4

5 4

5 x 4 = 20

20 4

4 - 1 = 3

20 3

20 x 3 = 60

60 3

3 - 1 = 2

60 2

60 x 2 = 120

120 2

2 – 1 = 1

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4		5 4
5 x 4 = 20		20 4
4 - 1 = 3		20 3
20 x 3 = 60		60 3
3 - 1 = 2		60 2
60 x 2 = 120		120 2
2 – 1 = 1		120 1

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4		5 4
5 x 4 = 20		20 4
4 - 1 = 3		20 3
20 x 3 = 60		60 3
3 - 1 = 2		60 2
60 x 2 = 120		120 2
2 – 1 = 1		120 1
120 x 1 = 120		

Autre façon de calculer

Algorithme : 5 x 4 x 3 x 2 x 1

Valeur de départ : 5

5 - 1 = 4		5 4
5 x 4 = 20		20 4
4 - 1 = 3		20 3
20 x 3 = 60		60 3
3 - 1 = 2		60 2
60 x 2 = 120		120 2
2 – 1 = 1		120 1
120 x 1 = 120		120 1

Exemple d'une addition

Ceci est une simplification de la réalité. Vous comprendrez mieux un peu plus tard.

Exemple d'une addition

Opération à effectuer : 3 + 4

Ceci est une simplification de la réalité. Vous comprendrez mieux un peu plus tard.

Exemple d'une addition

Opération à effectuer : 3 + 4

Stocke 3 en mémoire  3

Ceci est une simplification de la réalité. Vous comprendrez mieux un peu plus tard.

Exemple d'une addition

Opération à effectuer : 3 + 4

Stocke 3 en mémoire  3

Stocke 4 en mémoire  3 4

Ceci est une simplification de la réalité. Vous comprendrez mieux un peu plus tard.

Exemple d'une addition

Opération à effectuer : 3 + 4

Stocke 3 en mémoire  3

Stocke 4 en mémoire  3 4

3 + 4 = 7  3 4 7

Ceci est une simplification de la réalité. Vous comprendrez mieux un peu plus tard.

Exemple d'une addition

Opération à effectuer : 3 + 4

Stocke 3 en mémoire  3

Stocke 4 en mémoire  3 4

3 + 4 = 7  3 4 7

Annonce du résultat  3 4 7

Ceci est une simplification de la réalité. Vous comprendrez mieux un peu plus tard.

Exemple d'une addition

Opération à effectuer : 3 + 4

Stocke 3 en mémoire  3

Stocke 4 en mémoire  3 4

3 + 4 = 7  3 4 7

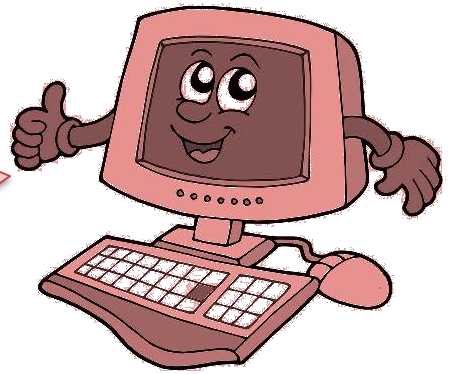
Annonce du résultat  3 4 7

On vide la mémoire 

Ceci est une simplification de la réalité. Vous comprendrez mieux un peu plus tard.

Synthèse

En algorithmique, on manipule des valeurs qui sont stockées dans la mémoire.
Interpréter un algorithme signifie que l'on est capable de déterminer à chaque étape les valeurs stockées en mémoire.



Il faut toujours avoir en tête une représentation de son algorithme !

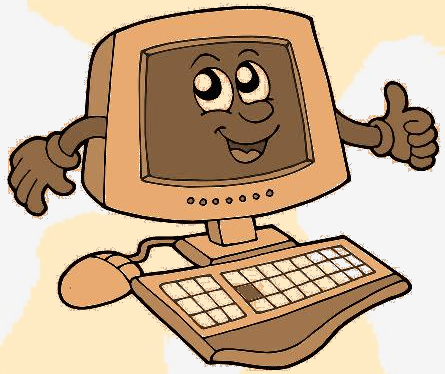
Exécution pas à pas ?

```
debut
a <- 5
b <- 3
a <- a + b
b <- a - b
a <- a - b
fin
```

Nos outils

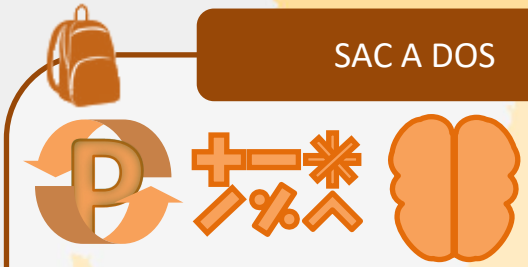
INVENTAIRE

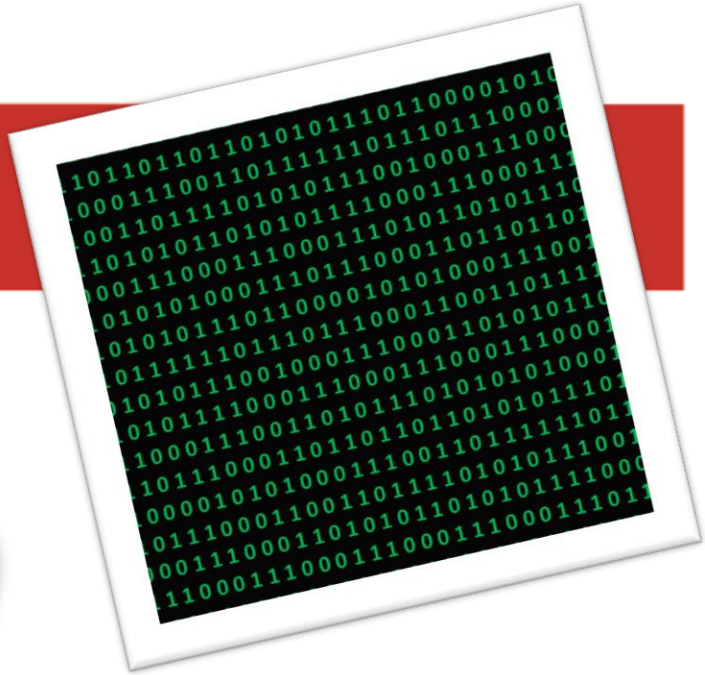
AVATAR



NOM : *Algorithmique*

SAC A DOS

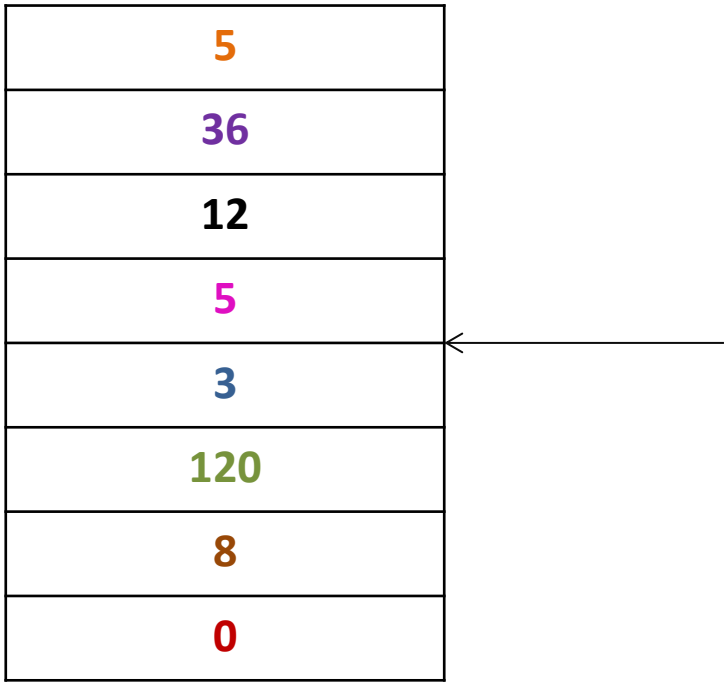




MÉMOIRE ET CONCEPTS

Mémoire : représentation

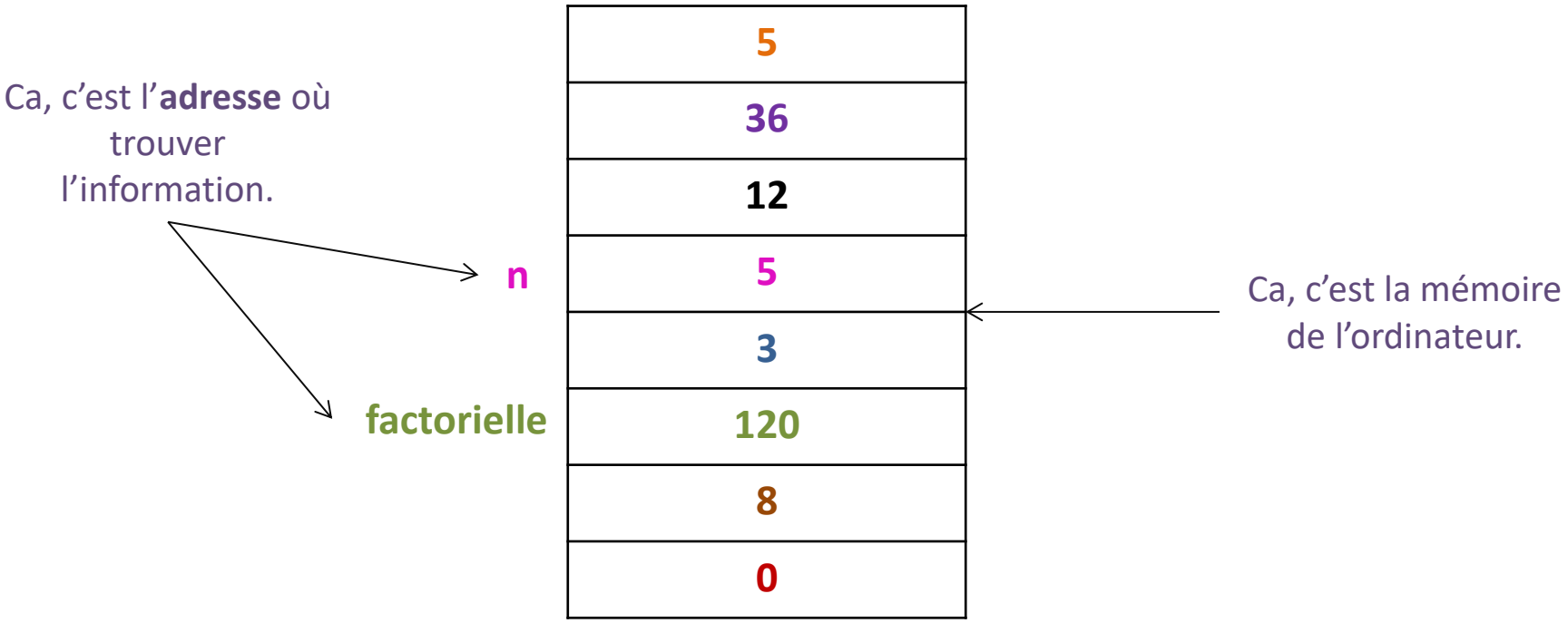
En vrai, la mémoire se représente sous la forme d'un tableau.



C'est une version simplifiée de la réalité.

Mémoire : représentation

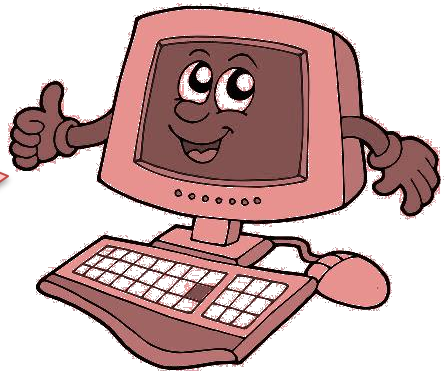
En vrai, la mémoire se représente sous la forme d'un tableau.



C'est une version simplifiée de la réalité.

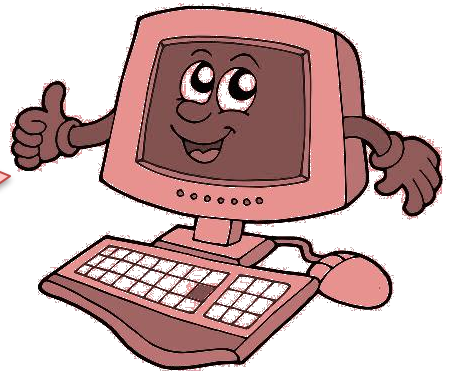
Variables

Et voilà, ça c'est le rôle des variables.
Stocker des informations dans la mémoire de l'ordinateur pour pouvoir utiliser ces informations dans l'algorithme.



Adresse mémoire

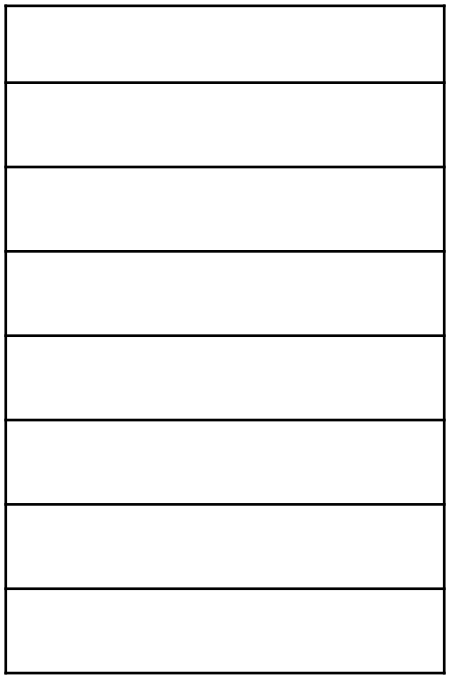
Une variable est une sorte d'adresse qui permet de retrouver une valeur en mémoire.
Pas besoin de savoir où la donnée est stockée dans la mémoire.



Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire

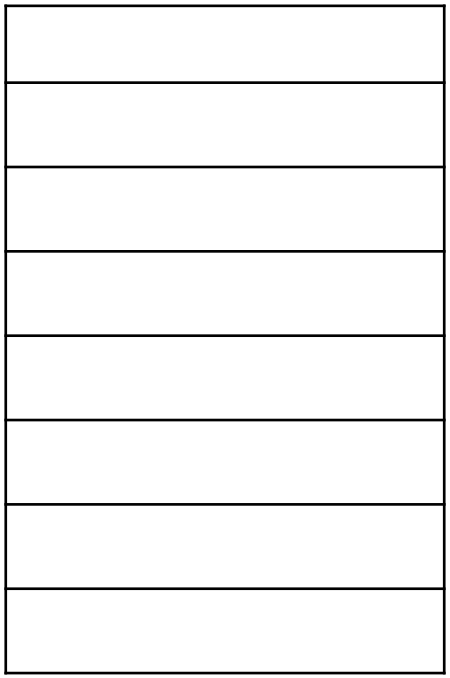


Liste des variables

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire



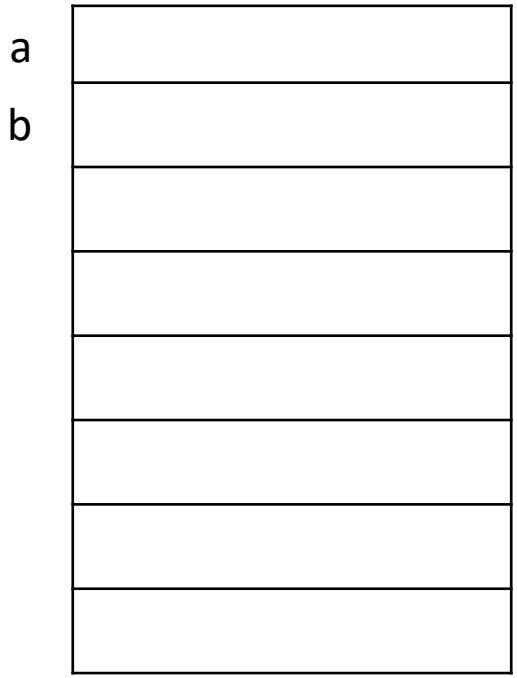
Liste des variables

J'ai besoin de deux nombres **a** et **b** afin de diviser a par b.

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire



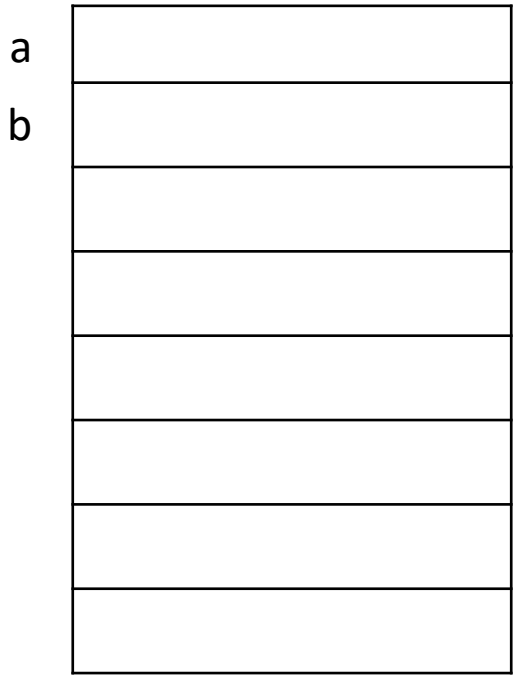
Liste des variables

J'ai besoin de deux nombres **a** et **b** afin de diviser a par b.

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire



Liste des variables

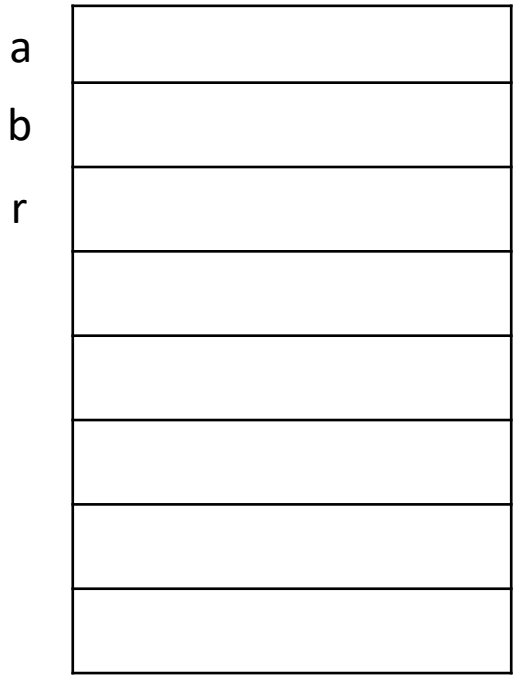
J'ai besoin de deux nombres **a** et **b** afin de diviser a par b.

J'ai besoin de stocker le reste **r** de la division euclidienne.

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire



Liste des variables

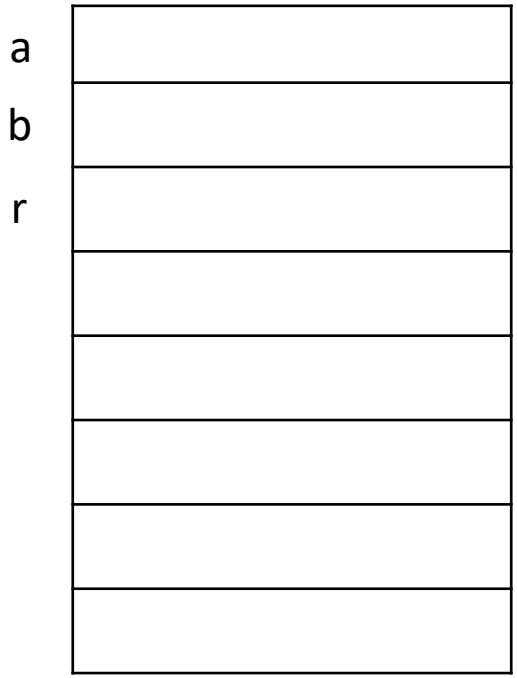
J'ai besoin de deux nombres **a** et **b** afin de diviser a par b.

J'ai besoin de stocker le reste **r** de la division euclidienne.

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire



Liste des variables

J'ai besoin de deux nombres **a** et **b** afin de diviser a par b.

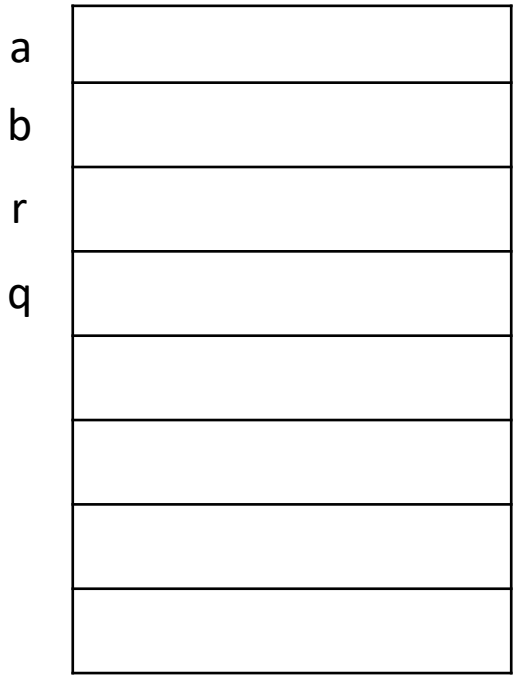
J'ai besoin de stocker le reste **r** de la division euclidienne.

J'ai besoin de stocker le quotient **q** de la division euclidienne.

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire



Liste des variables

J'ai besoin de deux nombres **a** et **b** afin de diviser a par b.

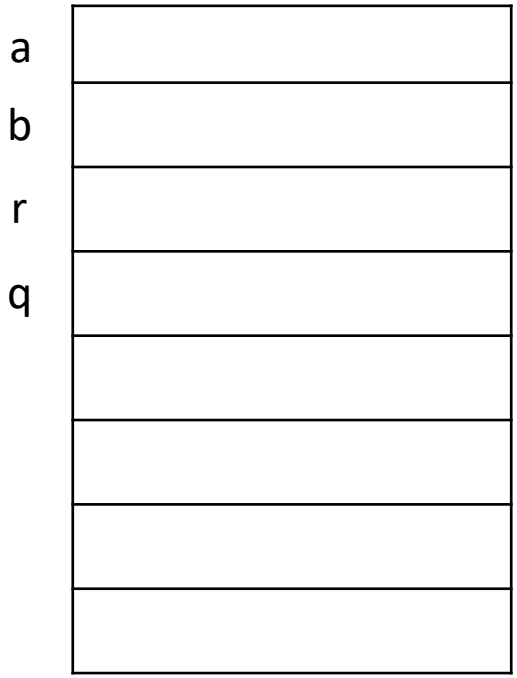
J'ai besoin de stocker le reste **r** de la division euclidienne.

J'ai besoin de stocker le quotient **q** de la division euclidienne.

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire

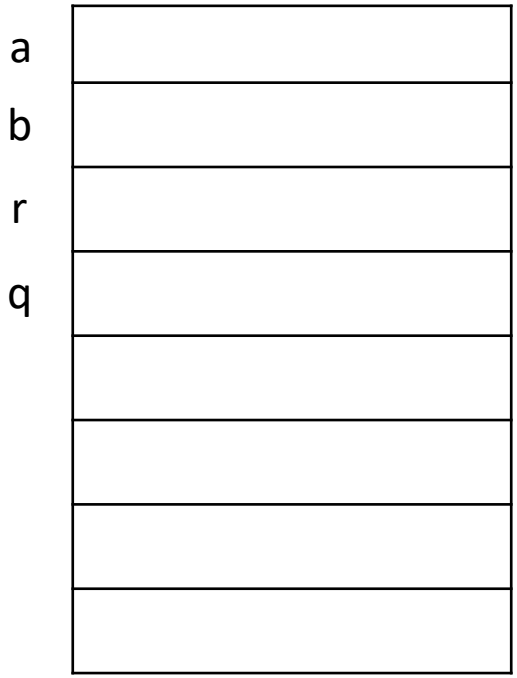


Opérations

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire



Opérations

```
a <- 16
```

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire

a	16
b	
r	
q	

Opérations

```
a <- 16
```

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire

a	16
b	
r	
q	

Opérations

```
a <- 16

b <- 5
```

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire

a	16
b	5
r	
q	

Opérations

```
a <- 16

b <- 5
```

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire

a	16
b	5
r	
q	

Opérations

```
a <- 16

b <- 5

r <- a%b
```

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire

a	16
b	5
r	1
q	

Opérations

```
a <- 16

b <- 5

r <- a%b
```

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire

a	16
b	5
r	1
q	

Opérations

```
a <- 16

b <- 5

r <- a%b

q <- a/b
```

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire

a	16
b	5
r	1
q	3

Opérations

```
a <- 16

b <- 5

r <- a%b

q <- a/b
```

Exemple de variables

Prenons l'exemple d'une division euclidienne

Mémoire

a	16
b	5
r	1
q	3

Opérations

```
a <- 16

b <- 5

r <- a%b

q <- a/b
```

Vérification ?

Exemple de variables

La zone en mémoire est réservée tant qu'on l'utilise

Mémoire

a	16
b	5
r	3
q	1

Opérations

Exemple de variables

La zone en mémoire est réservée tant qu'on l'utilise

Mémoire

a	16
b	5
r	3
q	1

Opérations

```
a <- 25
```

Exemple de variables

La zone en mémoire est réservée tant qu'on l'utilise

Mémoire

a	25
b	5
r	3
q	1

Opérations

```
a <- 25
```

Exemple de variables

La zone en mémoire est réservée tant qu'on l'utilise

Mémoire

a	25
b	5
r	3
q	1

Opérations

```
a <- 25

b <- 10
```

Exemple de variables

La zone en mémoire est réservée tant qu'on l'utilise

Mémoire

a	25
b	10
r	3
q	1

Opérations

```
a <- 25

b <- 10
```

Exemple de variables

La zone en mémoire est réservée tant qu'on l'utilise

Mémoire

a	25
b	10
r	3
q	1

Opérations

```
a <- 25

b <- 10

r <- 2*a + b
```

Exemple de variables

La zone en mémoire est réservée tant qu'on l'utilise

Mémoire

a	25
b	10
r	60
q	1

Opérations

```
a <- 25

b <- 10

r <- 2*a + b
```

Exemple de variables

La zone en mémoire est réservée tant qu'on l'utilise

Mémoire

a	25
b	10
r	60
q	1

Opérations

```
a <- 25

b <- 10

r <- 2*a + b

r <- 0
```

Exemple de variables

La zone en mémoire est réservée tant qu'on l'utilise

Mémoire

a	25
b	10
r	0
q	1

Opérations

```
a <- 25

b <- 10

r <- 2*a + b

r <- 0
```

Question

Comprendre le fonctionnement de la machine
Reformuler, interpréter, vérifier, écrire

Opérateur
Variable
Constante
Déclaration
Affectation
Initialisation

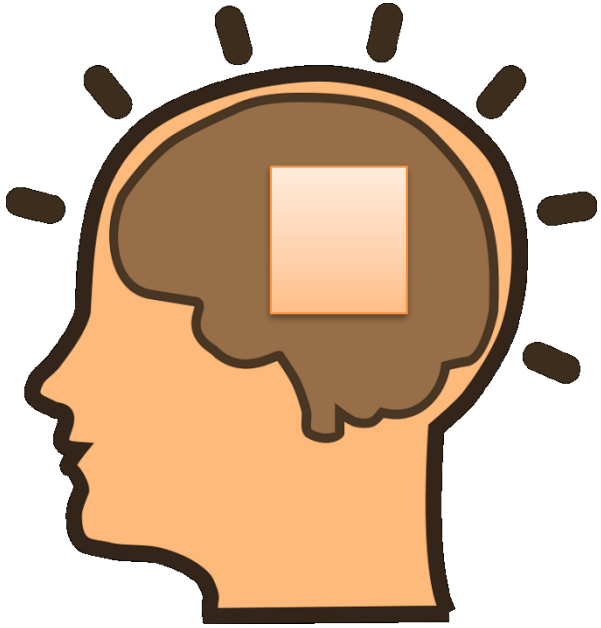
Et, c'est quoi une constante ?

VOCABULAIRE CLEF POUR LA REFORMULATION

Comprendre le fonctionnement de la machine
Reformuler, interpréter, vérifier, écrire

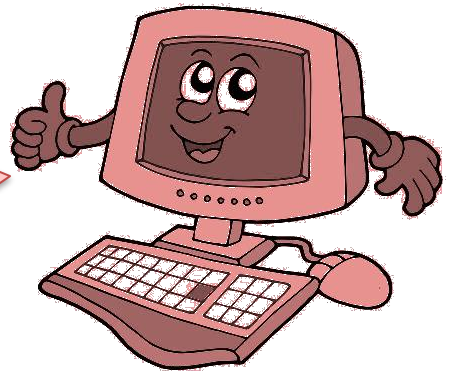
Opérateur
Variable
Constante
Déclaration
Affectation
Initialisation

Déclaration

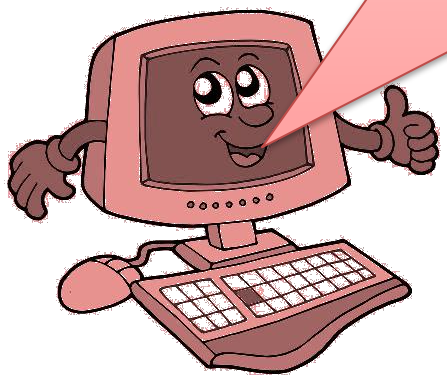


La vérité sur la déclaration

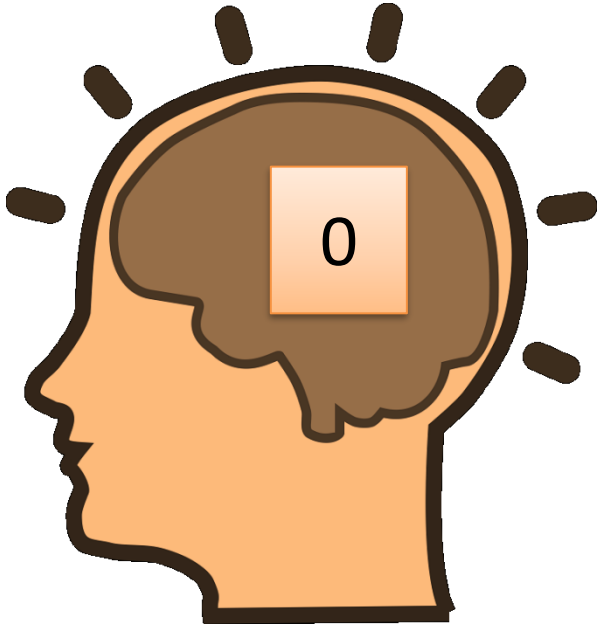
La **déclaration** permet de réserver une zone en mémoire. Mais attention, la mémoire n'est pas vide. La zone mémoire est toujours remplie de ce qu'elle contenait avant.



Affectation



Hey, tu peux stocker une valeur dans la zone mémoire que j'ai réservé s'il te plaît ?
Pour l'instant ma **somme** vaut 0.



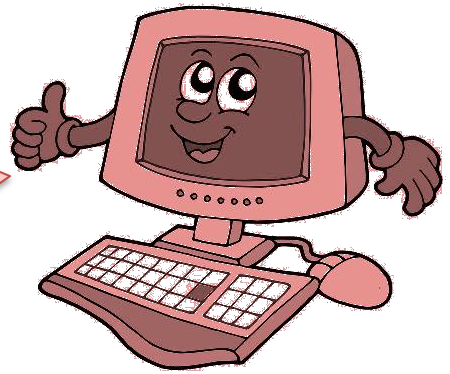
Exemple d'**affectation** :

somme <- 0

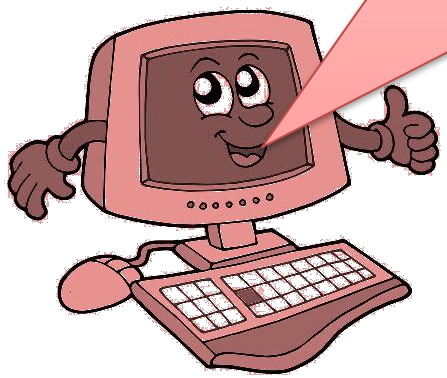
Initialisation

L'**initialisation**, c'est le fait d'affecter une valeur à une variable la première fois. C'est la toute première affectation.

Et : **il est obligatoire d'initialiser une variable !**



Expression



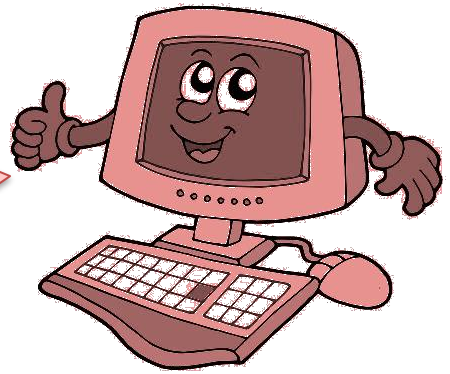
Hey, tu peux calculer **2+5** et mettre le résultat de l'opération dans la zone mémoire réservée à la **somme** s'il te plaît ?

Exemple d'expression :

3 * (7 + 9)

Précision sur l'expression

On peut dire qu'une expression c'est un ensemble d'opérations amenant à un résultat, c'est-à-dire un calcul effectué avec un ou plusieurs opérateurs.

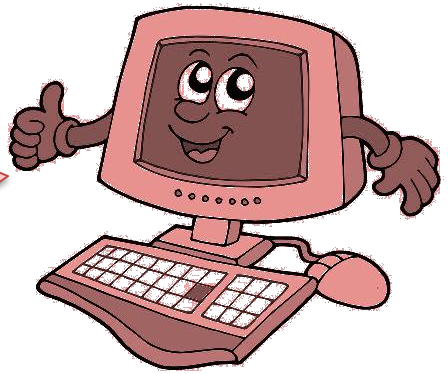


Ordre de calcul

Dans « somme <- a + b »,
l'expression est « a + b ».

L'ordinateur calcule d'abord
l'expression a+b et ensuite
affecte le résultat dans somme.

C'est toujours cet ordre là !



Synthèse 1/2

Synthèse 1/2

Algorithme : **déclarations + initialisations + instructions**

Synthèse 1/2

Algorithme : **déclarations + initialisations + instructions**

Appliquer des **données** à un algorithme fournit un **résultat**

Synthèse 1/2

Algorithme : **déclarations + initialisations + instructions**

Appliquer des **données** à un algorithme fournit un **résultat**

Instructions effectuées selon **ordre séquentiel** (les unes après les autres)

Synthèse 1/2

Algorithme : **déclarations + initialisations + instructions**

Appliquer des **données** à un algorithme fournit un **résultat**

Instructions effectuées selon **ordre séquentiel** (les unes après les autres)

Reformuler : **expliquer** ce que l'on voit, **comprendre** l'algorithme

Synthèse 1/2

Algorithme : **déclarations + initialisations + instructions**

Appliquer des **données** à un algorithme fournit un **résultat**

Instructions effectuées selon **ordre séquentiel** (les unes après les autres)

Reformuler : **expliquer** ce que l'on voit, **comprendre** l'algorithme

Interpréter : appliquer, calculer, **exécuter** (avec la tête), dérouler, chercher le résultat…

Synthèse 1/2

Algorithme : **déclarations + initialisations + instructions**

Appliquer des **données** à un algorithme fournit un **résultat**

Instructions effectuées selon **ordre séquentiel** (les unes après les autres)

Reformuler : **expliquer** ce que l'on voit, **comprendre** l'algorithme

Interpréter : appliquer, calculer, **exécuter** (avec la tête), dérouler, chercher le résultat…

Permutation

Synthèse 2/2

Opérateurs : 6 opérateurs arithmétiques : +, -, *, /, ^, %

Modulo % : un opérateur qui nous retourne le reste de la division euclidienne de deux nombres

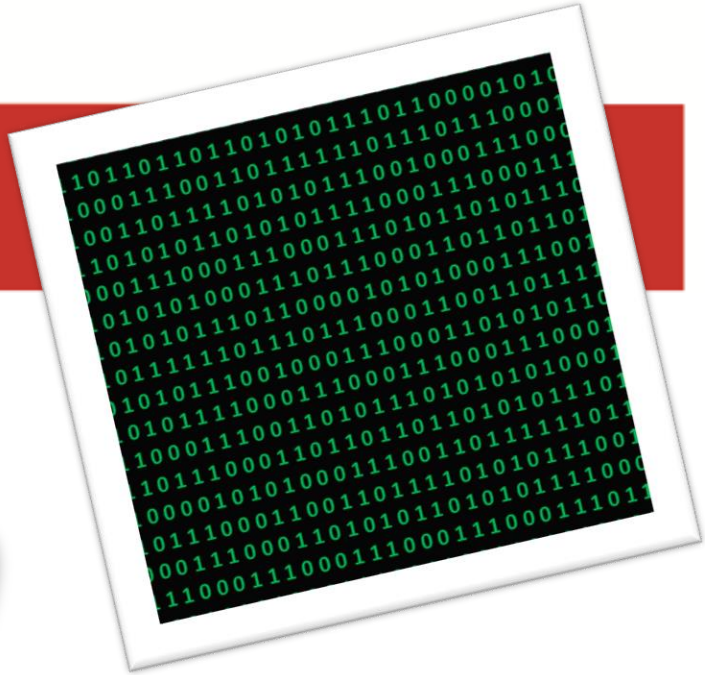
a = bq + r	a/b = q	a%b = r
*13 = 5*2 + 3*	*13/5 = 2*	*13%5 = 3*

Mémoire : tableau, jamais vide, stocke le résultat des opérations, une valeur par case, accès par variable, pas d'historique. On ne sait pas ce qu'il y a dans une zone qu'on réserve.

Variable : adresse qui pointe vers une case de la mémoire qui contient une valeur

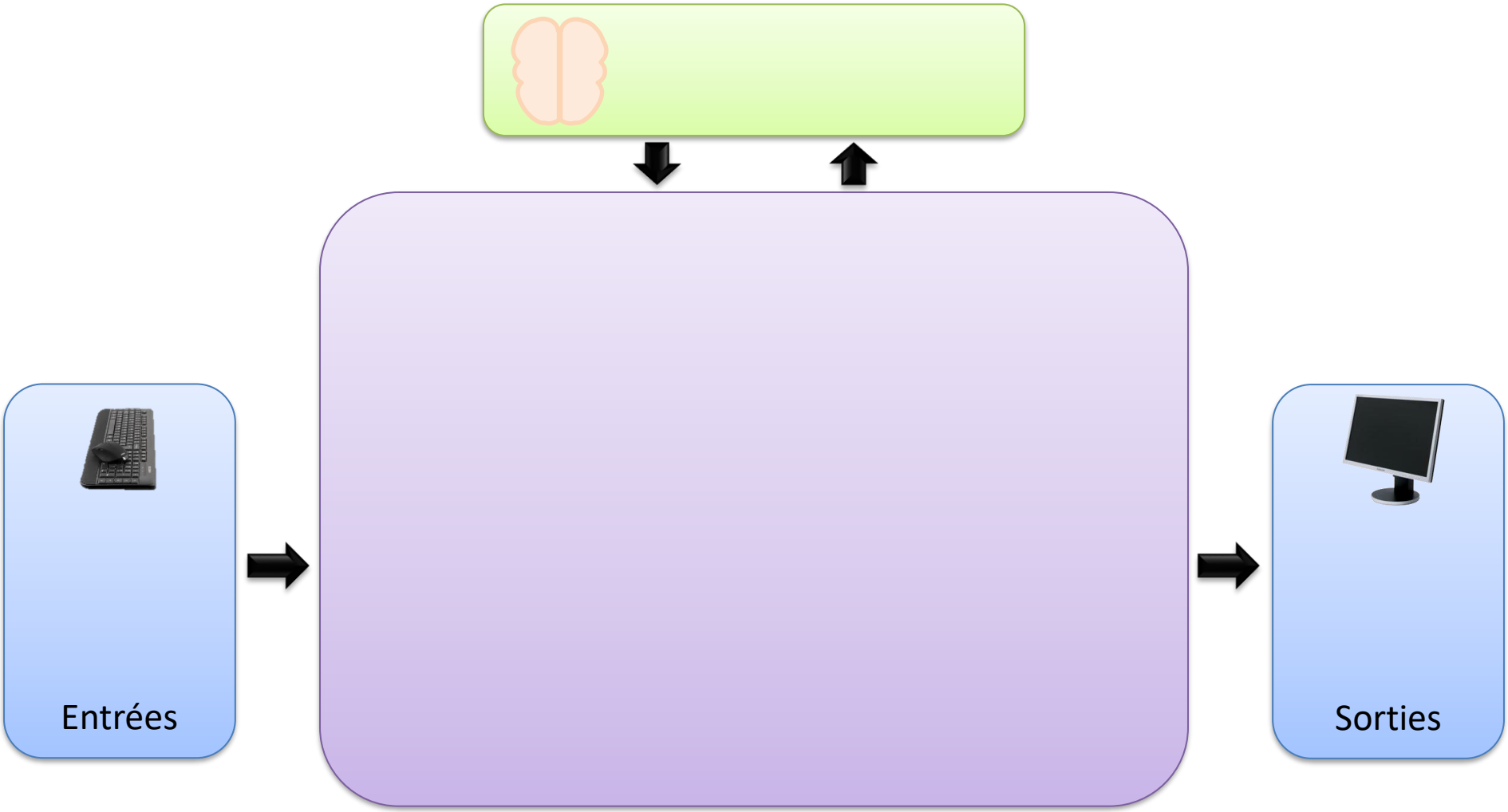
Constante : variable accessible en lecture seule

Vocabulaire : déclaration, initialisation, affectation (attention à l'ordre), expression, instruction, variable, constante.

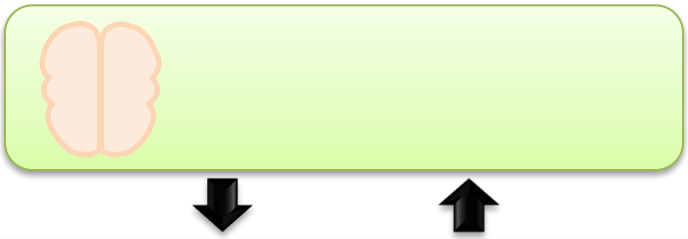


LECTURE ET ÉCRITURE

Lecture et écriture



Lecture et écriture

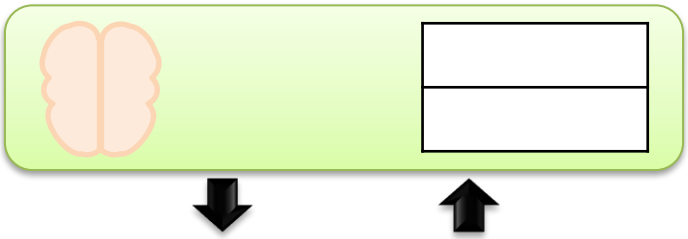


```
var reponse : chaine
```

Entrées

Sorties

Lecture et écriture

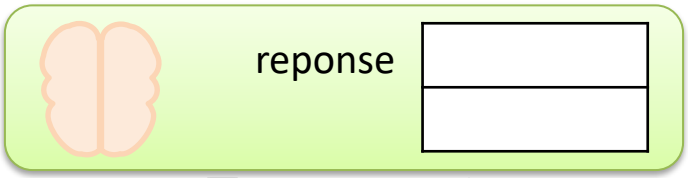


```
var reponse : chaine
```

Entrées

Sorties

Lecture et écriture

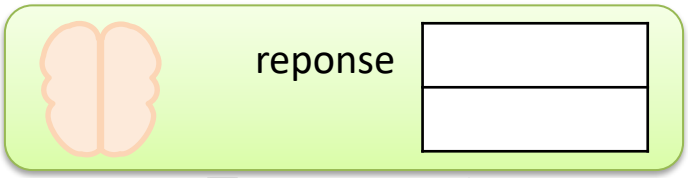



```
var reponse : chaine
```

Entrées

Sorties

Lecture et écriture

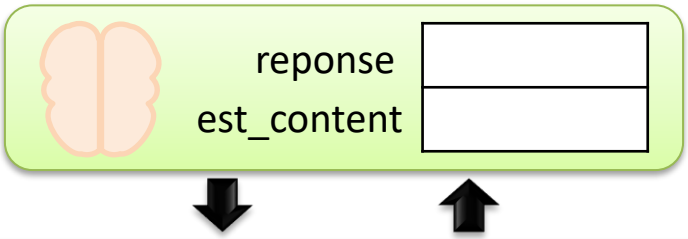



```
var reponse : chaine
var est_content : booleen
```

Entrées

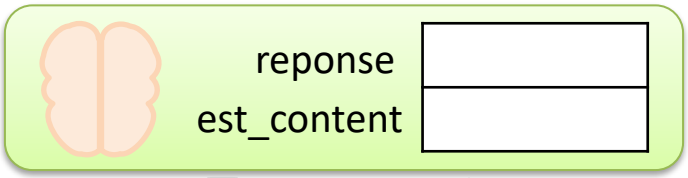
Sorties

Lecture et écriture




```
var reponse : chaine
var est_content : booleen
```

Lecture et écriture



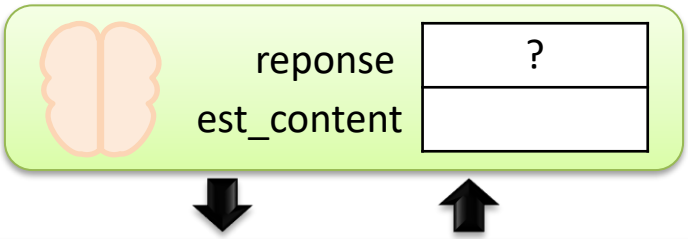

```
var reponse : chaine
var est_content : booleen

reponse <- "?"
```

Entrées

Sorties

Lecture et écriture



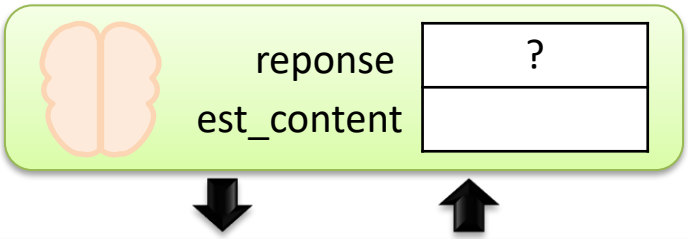

```
var reponse : chaine
var est_content : booleen

reponse <- "?"
```

Entrées

Sorties

Lecture et écriture




```
var reponse : chaine
var est_content : booleen

reponse <- "?"
est_content <- faux
```

Entrées

Sorties

Lecture et écriture

reponse	?
est_content	faux

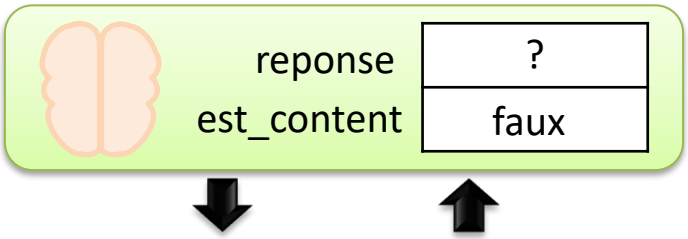
```
var reponse : chaine
var est_content : booleen

reponse <- "?"
est_content <- faux
```

Entrées

Sorties

Lecture et écriture




```
var reponse : chaine
var est_content : booleen

reponse <- "?"
est_content <- faux

ecrire "Etes-vous content ?"
```

Entrées

Sorties

Lecture et écriture

reponse	?
est_content	faux

```
var reponse : chaine
var est_content : booleen

reponse <- "?"
est_content <- faux

ecrire "Etes-vous content ?"
```

Entrées

Etes-vous content ?

Sorties

Lecture et écriture

reponse	?
est_content	faux

```
var reponse : chaine
var est_content : booleen

reponse <- "?"
est_content <- faux

ecrire "Etes-vous content ?"
lire reponse
```

Entrées

Etes-vous content ?

Sorties

Lecture et écriture

reponse	oui
est_content	faux

```
var reponse : chaine
var est_content : booleen

reponse <- "?"
est_content <- faux

ecrire "Etes-vous content ?"
lire reponse
```

oui

Entrées

Etes-vous content ?

Sorties

Lecture et écriture

reponse	oui
est_content	faux

```
var reponse : chaine
var est_content : booleen

reponse <- "?"
est_content <- faux

ecrire "Etes-vous content ?"
lire reponse
si reponse="non" alors
```

oui

Entrées

Etes-vous content ?

Sorties

Lecture et écriture

reponse	oui
est_content	faux

```
var reponse : chaine
var est_content : booleen

reponse <- "?"
est_content <- faux

ecrire "Etes-vous content ?"
lire reponse
si reponse="non" alors
  est_content <- faux
sinon si reponse="oui" alors
```

oui

Entrées

Etes-vous content ?

Sorties

Lecture et écriture

reponse	oui
est_content	faux

```
var reponse : chaine
var est_content : booleen

reponse <- "?"
est_content <- faux

ecrire "Etes-vous content ?"
lire reponse
si reponse="non" alors
  est_content <- faux
sinon si reponse="oui" alors
  est_content <- vrai
```

oui

Entrées

Etes-vous content ?

Sorties

Lecture et écriture

reponse	oui
est_content	vrai

```
var reponse : chaine
var est_content : booleen

reponse <- "?"
est_content <- faux

ecrire "Etes-vous content ?"
lire reponse
si reponse="non" alors
  est_content <- faux
sinon si reponse="oui" alors
  est_content <- vrai
```

oui

Entrées

Etes-vous content ?

Sorties

Lecture et écriture

reponse	oui
est_content	vrai

```
var reponse : chaine
var est_content : booleen

reponse <- "?"
est_content <- faux

ecrire "Etes-vous content ?"
lire reponse
si reponse="non" alors
  est_content <- faux
sinon si reponse="oui" alors
  est_content <- vrai
sinon
```

oui

Entrées

Etes-vous content ?

Sorties

Lecture et écriture

reponse	oui
est_content	vrai

```
var reponse : chaine
var est_content : booleen

reponse <- "?"
est_content <- faux

ecrire "Etes-vous content ?"
lire reponse
si reponse="non" alors
  est_content <- faux
sinon si reponse="oui" alors
  est_content <- vrai
sinon
  ecrire "Pouvez-vous écrire oui ou non ?"
finsi
```

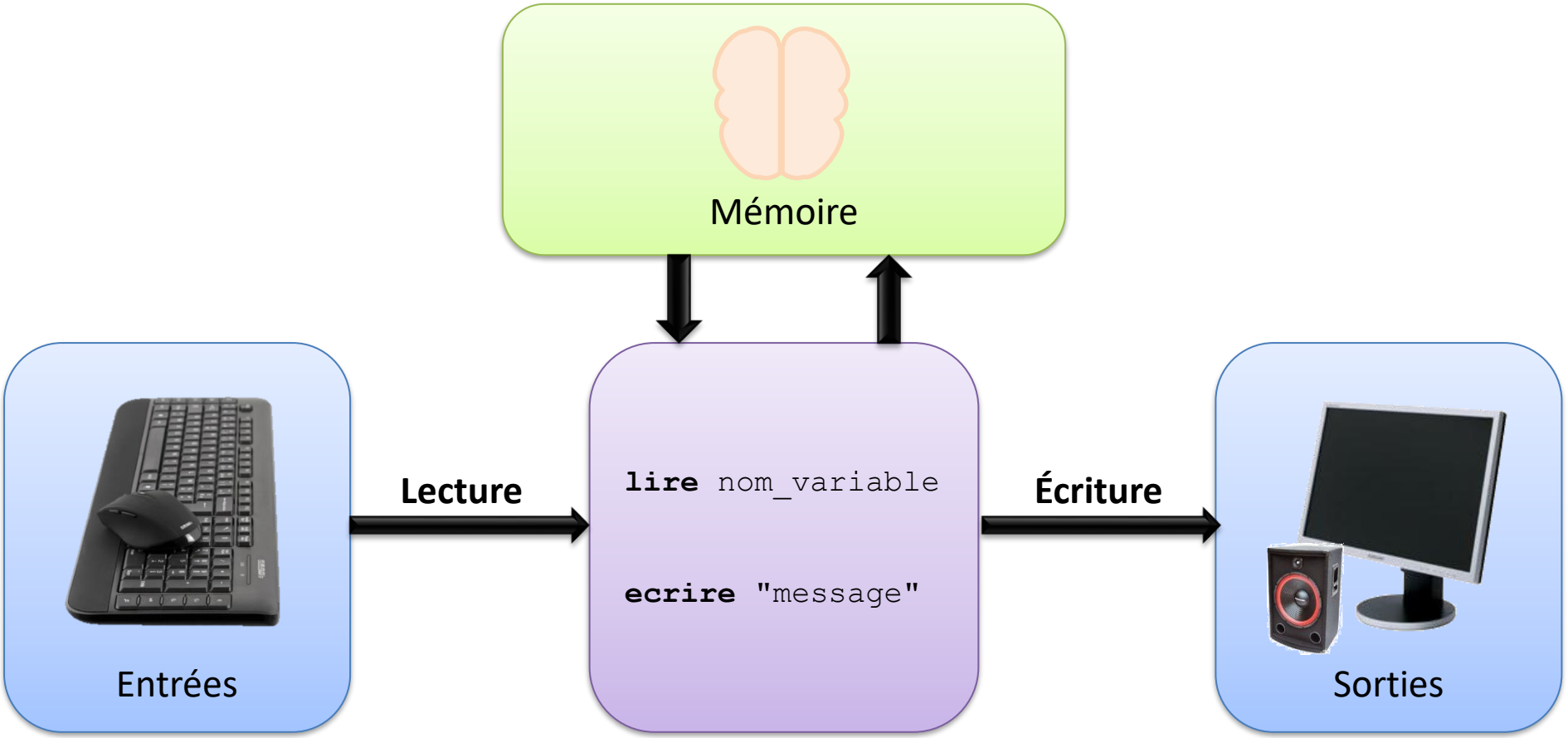
oui

Entrées

Etes-vous content ?

Sorties

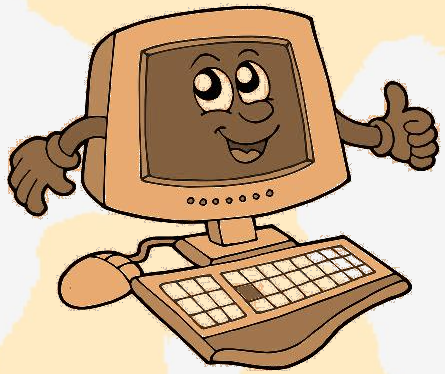
Lecture et écriture



Nos outils

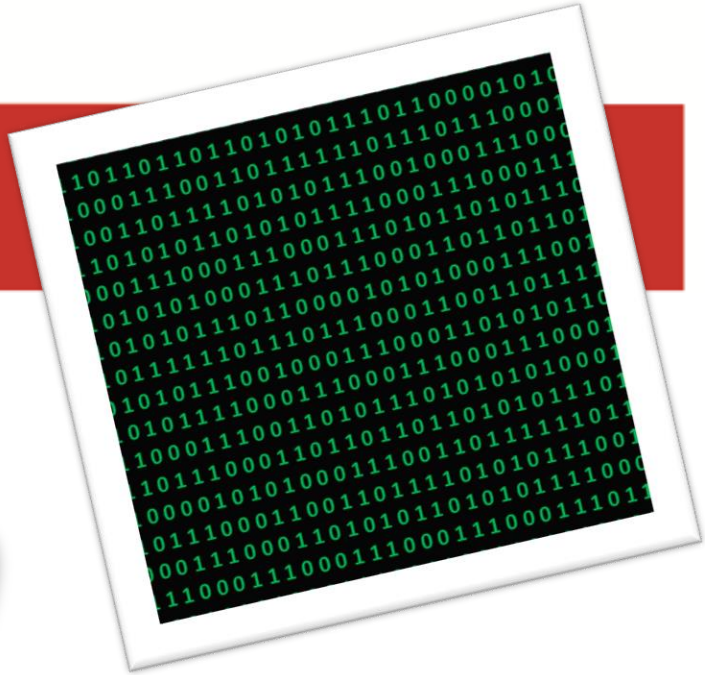
INVENTAIRE

AVATAR



NOM : *Algorithmique*

SAC A DOS



VERS LE PROCHAIN COURS

Ouverture...

```
debut

fin
```

Ouverture…

```
debut
   imc <- 27.0




fin
```

Ouverture...

```
debut
   imc <- 27.0
   resultat <- ""




fin
```

Ouverture...

```
debut
   imc <- 27.0
   resultat <- ""
   si imc < 18.5 alors




fin
```

Ouverture...

```
debut
  imc <- 27.0
  resultat <- ""
  si imc < 18.5 alors
    resultat <- "sous-poids"

fin
```

Ouverture...

```
debut
   imc <- 27.0
   resultat <- ""
   si imc < 18.5 alors
      resultat <- "sous-poids"
   sinon si imc < 25.0 alors




fin
```

```
debut
  imc <- 27.0
  resultat <- ""
  si imc < 18.5 alors
    resultat <- "sous-poids"
  sinon si imc < 25.0 alors
    resultat <- "normal"

fin
```

Ouverture...

```
debut
  imc <- 27.0
  resultat <- ""
  si imc < 18.5 alors
    resultat <- "sous-poids"
  sinon si imc < 25.0 alors
    resultat <- "normal"
  sinon si imc < 30.0 alors




fin
```

Ouverture...

```
debut
  imc <- 27.0
  resultat <- ""
  si imc < 18.5 alors
    resultat <- "sous-poids"
  sinon si imc < 25.0 alors
    resultat <- "normal"
  sinon si imc < 30.0 alors
    resultat <- "surpoids"




fin
```

Ouverture...

```
debut
  imc <- 27.0
  resultat <- ""
  si imc < 18.5 alors
    resultat <- "sous-poids"
  sinon si imc < 25.0 alors
    resultat <- "normal"
  sinon si imc < 30.0 alors
    resultat <- "surpoids"
  sinon si imc < 35.0 alors




fin
```

Ouverture...

```
debut
  imc <- 27.0
  resultat <- ""
  si imc < 18.5 alors
    resultat <- "sous-poids"
  sinon si imc < 25.0 alors
    resultat <- "normal"
  sinon si imc < 30.0 alors
    resultat <- "surpoids"
  sinon si imc < 35.0 alors
    resultat <- "obésité"



fin
```

Ouverture...

```
debut
  imc <- 27.0
  resultat <- ""
  si imc < 18.5 alors
    resultat <- "sous-poids"
  sinon si imc < 25.0 alors
    resultat <- "normal"
  sinon si imc < 30.0 alors
    resultat <- "surpoids"
  sinon si imc < 35.0 alors
    resultat <- "obésité"
  sinon



fin
```

Ouverture...

```
debut
  imc <- 27.0
  resultat <- ""
  si imc < 18.5 alors
    resultat <- "sous-poids"
  sinon si imc < 25.0 alors
    resultat <- "normal"
  sinon si imc < 30.0 alors
    resultat <- "surpoids"
  sinon si imc < 35.0 alors
    resultat <- "obésité"
  sinon
    resultat <- "obésité sévère"


fin
```

Ouverture...

```
debut
  imc <- 27.0
  resultat <- ""
  si imc < 18.5 alors
    resultat <- "sous-poids"
  sinon si imc < 25.0 alors
    resultat <- "normal"
  sinon si imc < 30.0 alors
    resultat <- "surpoids"
  sinon si imc < 35.0 alors
    resultat <- "obésité"
  sinon
    resultat <- "obésité sévère"
  finsi

fin
```

Ouverture...

```
debut
  imc <- 27.0
  resultat <- ""
  si imc < 18.5 alors
    resultat <- "sous-poids"
  sinon si imc < 25.0 alors
    resultat <- "normal"
  sinon si imc < 30.0 alors
    resultat <- "surpoids"
  sinon si imc < 35.0 alors
    resultat <- "obésité"
  sinon
    resultat <- "obésité sévère"
  finsi
  ecrire resultat
fin
```