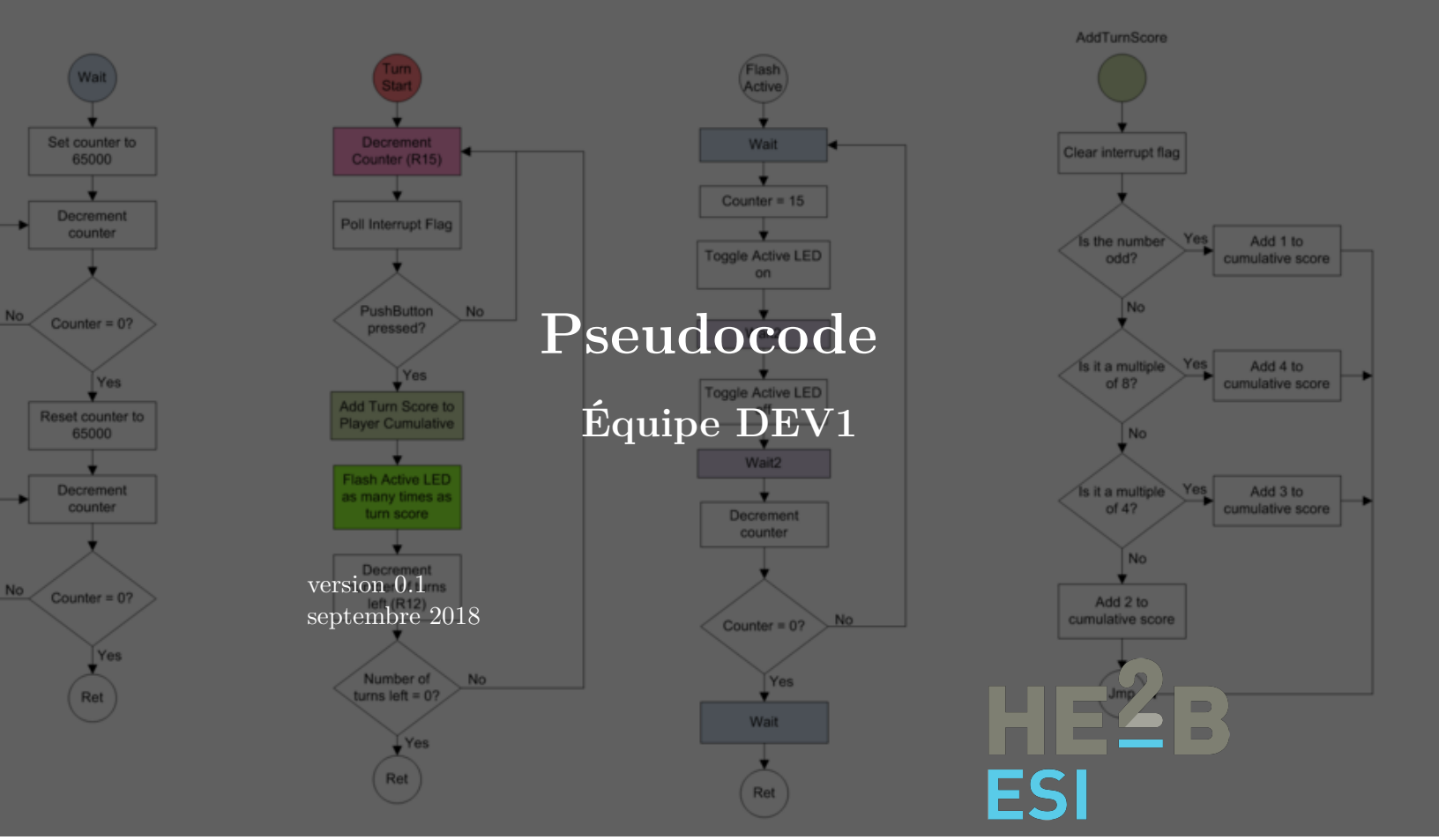



# Pseudocode

## Équipe DEV1

version 0.1
septembre 2018



Le **pseudocode** est une manière de décrire un algorithme en langage presque naturel. C'est un ensemble de phrases représentant l'enchainement des opérations nécessaires à la résolution du problème.

Par exemple, un jardinier pourrait dire:

> Tant que l'on est pas arrivé à la fin de la route, faire un trou et repiquer un poireau.

Nous allons formaliser un peu tout ça.

On pourrait croire à priori, qu'il existe autant de pseudocodes que de personnes décrivant un algorithme. C'est un peu vrai... et faux. Dès lors que l'on veut décrire un algorithme, c'est pour le partager avec d'autres... Ceci implique que certaines régles soient définies.

Nous présentons ici ce que nous pensons être le sous-ensemble minimal de règles à respecter pour ne pas être (trop) ambigü et pour ne pas devoir *apprendre* le pseudocode. Ceci dit, si le lecteur trouve que c'est ambigü, c'est ambigü.

## Un traitement, un mot, un algorithme

Pour faire appel à un traitement, une opération, un algorithme, nous utilisons un mot en *mixedCase*.

Un mot en *mixedCase* est un mot composé de plusieurs mots. Collés. Chaque mot commencant par une majuscule excepté le premier. Par exemple: faireUnTrou, remplirLeFiltre, putLeekInHole...

```
faireUnTrou()
remplirLeFiltre()
putLeekInHole()
```

*pseudocode*

Certaines actions sont des actions élémentaires qui ne demandent aucune explication, d'autres sont plus complexes et doivent être expliquées. Elles le sont dans un algorithme. Un algorithme est une suite d'opérations... qui sont des actions élémentaires ou des opérations plus complexes qui doivent être expliquées... et ainsi de suite.

Définir un algorithme, c'est:

- lui donner un nom représentatif de ce qu'il fait;
- commencer par le mot algorithm (ou algorithme ou encore algo);
- **indenter** les opérations de manière à **marquer clairement le bloc** d'opérations (avec une ligne verticale blanche, ou au crayon, ou sans, ou... du moment que l'ensemble est cohérent).

```
algorithm plantOneLeek()
    makeHole()
    putLeekInHole()
```

*pseudocode*

# Les structures conditionnelles, if

Pour représenter le **si** (**if**) nous utiliserons cette notation:

```
if condition then
    statement
```

*pseudocode*

où:

- condition est une expression booléenne... une expression vraie ou fausse;
- statement est une instruction (une opération) ou plusieurs.

Exemple:

```
if thereAreLeeks then
    plantOneLeek
```

*pseudocode*

Remarques:

- dans les notes, nous utiliserons l'anglais mais le français est bien aussi;
- il est important de marquer le bloc d'instructions. Nous utilisons une barre verticale mais un endIf pourrait faire l'affaire;
- nous utilisons if-then mais nous comprenons si-alors[1];
- ...

Les autres structures conditionnelles se représentent comme suit:

```
if condition then
    statement
else
    statement
```

*pseudocode*

[1] Nous comprenons aussi if-alors ou si-then... mais bon, *faut pas pousser* !

```pseudocode
if condition then
    statement
else if condition then
    statement
else
    statement
```

*pseudocode*

### La structure **switch**

Le *selon que* (switch), s'écrit:

```pseudocode
switch dayNumber
    case 1: dayName = "lundi"
    case 2: dayName = "mardi"
    case 3: dayName = "mercredi"
    case 4: dayName = "jeudi"
    case 5: dayName = "vendredi"
    case 6: dayName = "samedi"
    case 7: dayName = "dimanche"
```

*pseudocode*

Remarques:

- nous utilisons *switch* mais nous comprenons *selon que*;
- nous ajoutons un case pour chaque cas mais nous comprenons *cas* ou l'utilisation d'un tiret;
- en langage Java le switch est associé au break. Nous n'en utilisons pas mais nous comprenons s'il y en a.

. . . du moment que l'ensemble est cohérent.

## Les structures répétitives

Nous formalisons les structrures répétitives les plus courantes: *tant que* (*while*), *faire tant que* (*do while*) et *pour* (*for*).

```
while condition
    statement
```
*pseudocode*

```
do
    statement
while condition
```
*pseudocode*

```
for i from 1 until n
    statement
```
*pseudocode*

Remarques:

- nous utilisons des mots anglais mais comprenons les équivalents français;
- todo

## Les paramètes

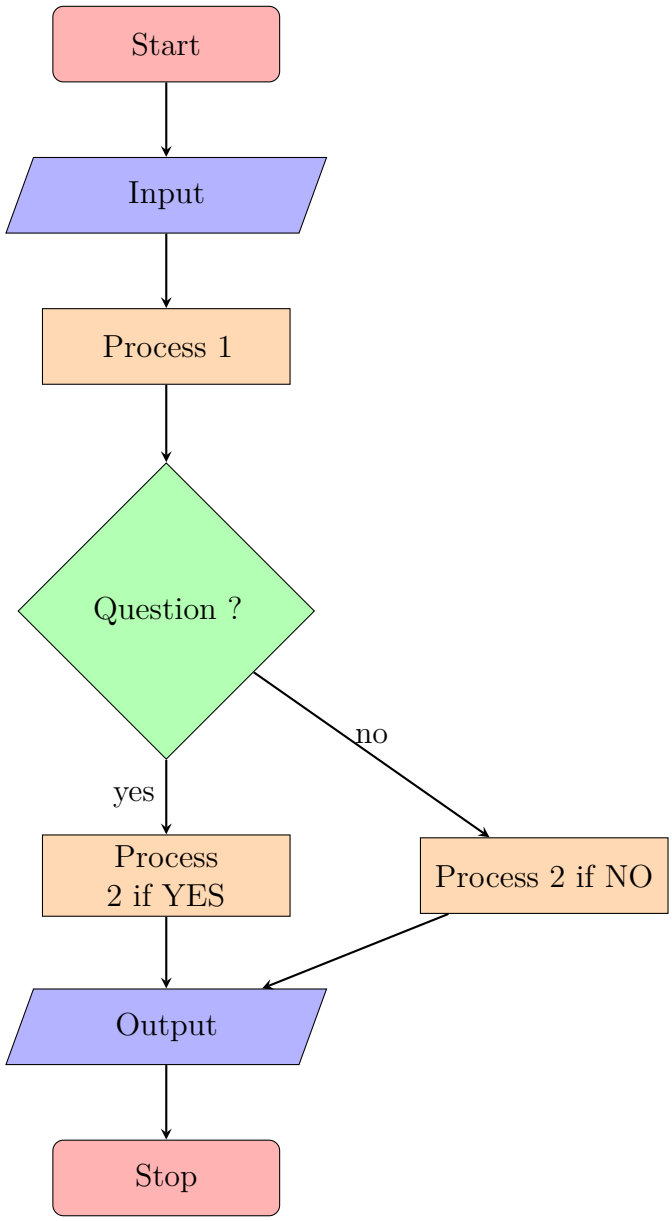
todo

## Les entrées-sorties

todo

## Exemple

todo à revoir




ORGANIGRAMME 1 – Exemple récapitulatif

*Crédit photo chez DeviantArt*[2] *par Susyspider*[3]*.Code des symboles issus de ShareLaTeX*[4]*.*



[2]http://deviantart.com
[3]https://www.deviantart.com/art/Simple-MSP430-Game-Subroutine-Flowcharts-302014732
[4]https://fr.sharelatex.com/blog/2013/08/29/tikz-series-pt3.html