Cours 1 : Qu'est-ce que la programmation?

Yann Régis-Gianas
yrg@pps.univ-paris-diderot.fr

Université Paris Diderot – Paris 7

1. Sortez un appareil qui peut se rendre sur l'internet (smartphone, ordinateur, tablette, ...) ;
2. **Si** il n'est pas connecté à internet, **alors** connectez-le ;
3. Ouvrez un navigateur à l'adresse suivante :

   http://www.ip101.fr/quiz/1
4. **Tant que** le prof ne parle pas, ne faites rien.
5. Ecoutez.

(Exécutez les instructions ci-dessus avec application, s'il vous plaît.)

Quiz 1 : Avez-vous déjà programmé ?

Quiz 2 : Avez-vous déjà été programmé ?

1. Sortez un appareil qui peut se rendre sur l'internet (smartphone, ordinateur, tablette, ...) ;
2. **Si** il n'est pas connecté à internet, **alors** connectez-le ;
3. Ouvrez un navigateur à l'adresse suivante :

   `http://www.ip101.fr/quiz/1`
4. **Tant que** le prof ne parle pas, ne faites rien.
5. Ecoutez.

(Vous avez exécuté votre premier programme d'IP1.)

Qu'est-ce qu'un programme ?

Qu'est-ce qu'un programme ?

Définition : Programme

Un **programme** est la **représentation** d'une **suite d'instructions**.

Qu'est-ce qu'un programme ?

Définition : Programme

Un **programme** est la **représentation** d'une **suite d'instructions**.

Cette définition pose question...

- Quelles instructions sont disponibles ?
- Dans quel langage écrire le programme
  pour qu'il n'existe qu'une seule façon de l'interpréter ?

La machine programmable

Définition : Machine

Une **machine** est un dispositif capable d'interagir avec son environnement. À chaque machine, on associe l'ensemble des actions qu'elle peut e□ectuer. Cet ensemble est appelé **jeu d'instructions de la machine**.

Définition : Interprétation d'un programme

Interpréter un programme, c'est (i) décoder la séquence d'instructions représentée par ce programme et ; (ii) exécuter ces instructions.

Définition : Machine programmable

Une machine est **programmable** quand elle est capable d'interpréter tout programme *qui est une représentation bien formée d'une séquence d'instructions appartenant à son jeu d'instructions*.

Exemples d'instructions

1. Calcule “$6 * 7$”.
2. Modifie le secteur `0x00CAFE00`[1] du disque dur.
3. A□che la couleur `#FF00FF` sur le pixel $(0, 1)$.
4. Calcule la somme des 10000 premiers nombres premiers.
5. Cherche Charlie sur la photo.
6. Apprends à parler en anglais.

1. Ceci est un nombre écrit en hexadécimal.

Quiz : Quelles sont les machines programmables ?

L'humain est-il une bonne machine programmable ?

Points faibles

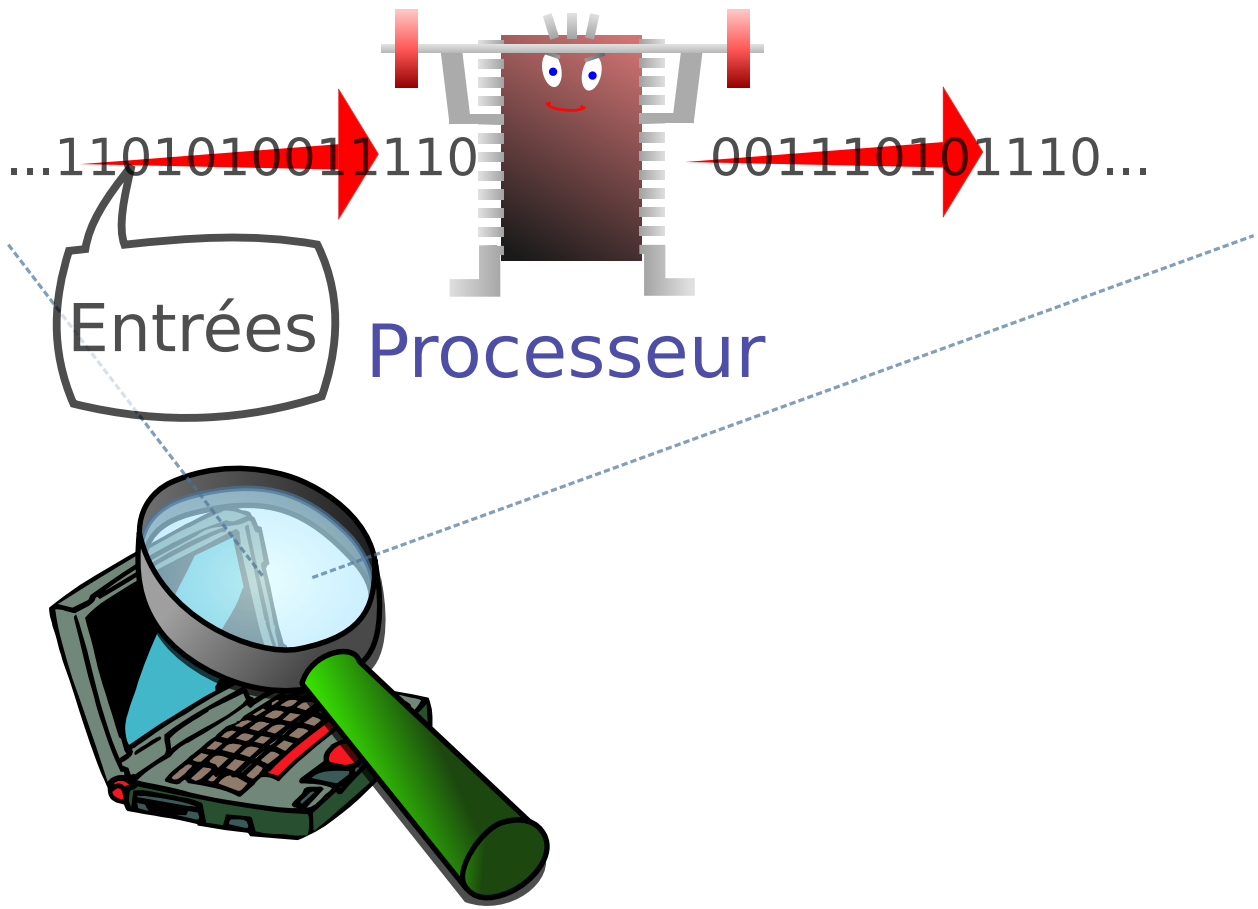
- Lent dans l'analyse et la communication de grosses quantités de données.
- Lent en calcul.
- S'ennuie vite.
- N'apprécie guère qu'on l'appelle une machine ou qu'on le programme.
- ...

Points forts

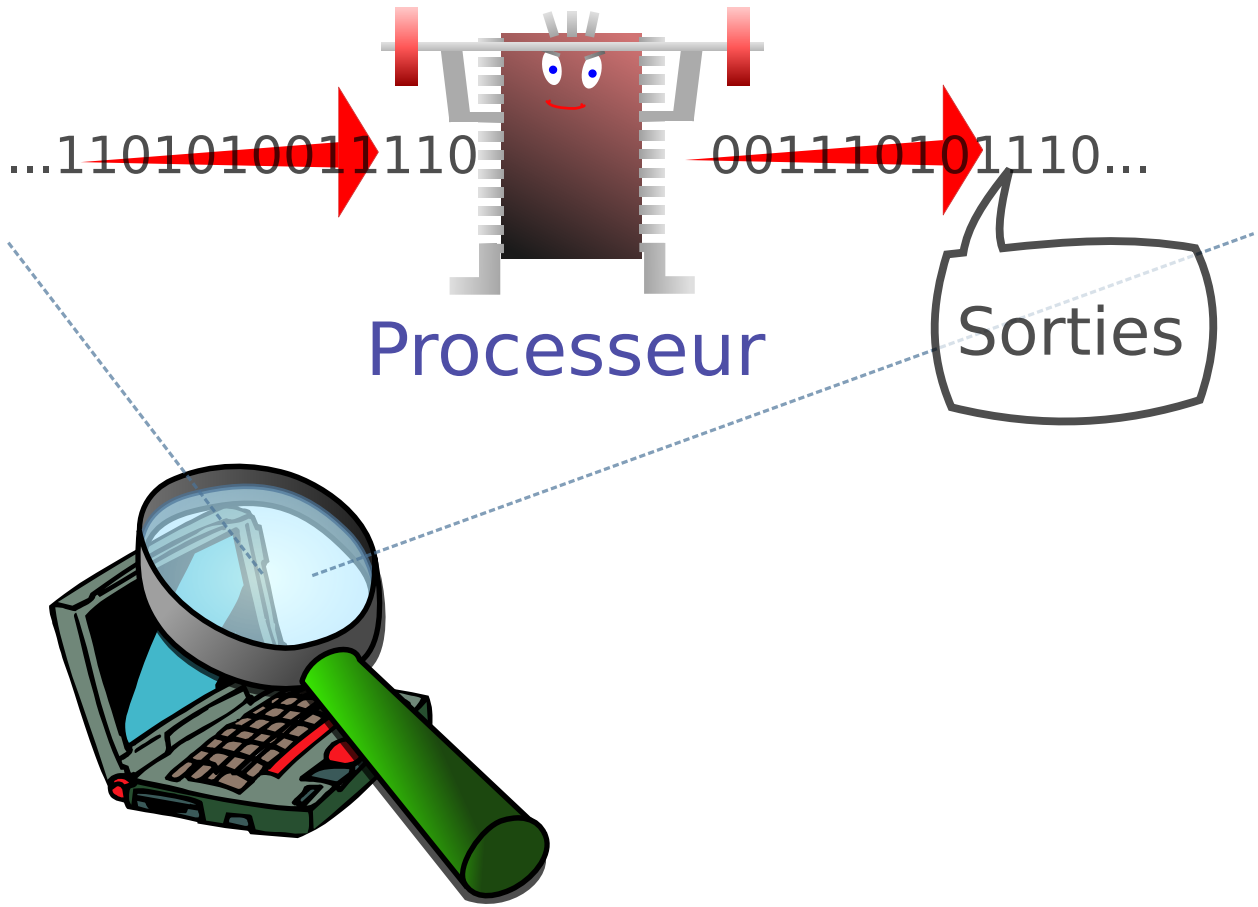
- E□cace en prise de décision en présence d'information partielle.
- E□cace en reconnaissance de motifs.
- E□cace en apprentissage.
- E□cace pour synthétiser des règles générales à partir d'exemples.
- ...

Qu'est-ce qu'un ordinateur ?

Qu'est-ce qu'un ordinateur ?



Qu'est-ce qu'un ordinateur ?



Quiz : A quelle vitesse calcule le processeur de votre smartphone ?

Le jeu d'instructions des ordinateurs

Observons l'intérieur de la machine...

- Regardons à l'intérieur de la machine.
- Regardons à l'intérieur d'un programme.
- (Allons au cours PF1.)

Le jeu d'instructions des ordinateurs

Observons l'intérieur de la machine...

- Regardons à l'intérieur de la machine.
- Regardons à l'intérieur d'un programme.
- (Allons au cours PF1.)

... et donc ?

- L'ordinateur est fait de circuits électroniques.
- L'information numérique y est représentée de façon analogique : 1 si il y a du courant, 0 sinon.
- On dit que la représentation est **binaire**.
- Le programme est lui-même représenté en binaire dans la machine.
- Les instructions travaillent sur des informations codées en binaire.
- Les instructions sont essentiellement des opérations de calcul en arithmétique binaire et des commandes pour les périphériques de l'ordinateur (clavier, mémoire, carte vidéo, ...).

Quiz : Qu'est-ce qui est facile ou di□cile pour un ordinateur ?

Quiz : Savoir utiliser un ordinateur, est-ce savoir programmer ?

“Non”

Avec la souris, en cliquant par exemple sur les menus des interfaces utilisateurs des di□érents logiciels, on **commande** l’ordinateur en lui donnant des instructions à exécuter les unes après les autres.

Programmer suppose la description d’un processus d’**automatisation**, reproductible par une simple **invocation** un grand nombre de fois.

Quiz : Pour programmer, faut-il être un *geek* ?


Anyone can ~~cook~~.
code

Savoir programmer est une **compétence essentielle** du scientifique
(et même du non-scientifique).

Savoir programmer est une **compétence essentielle** du scientifique
(et même du non-scientifique).

Savoir programmer est une **façon de penser**.
(et non pas un savoir-faire technologique)

Savoir programmer est une **compétence essentielle** du scientifique
(et même du non-scientifique).

Savoir programmer est une **façon de penser.**
(et non pas un savoir-faire technologique)

Savoir programmer, vous verrez, c'est facile !
(en commençant progressivement)

Objectifs du cours

- **Savoir écrire un programme simple** formé de séquence d'instructions, d'instructions conditionnelles, de boucles, de fonctions et de procédures.

Objectifs du cours

- **Savoir écrire un programme simple** formé de séquence d'instructions, d'instructions conditionnelles, de boucles, de fonctions et de procédures.
- **Manipuler des données** de nature diverse comme les entiers, les chaînes de caractères, les tableaux, les images, ...

Objectifs du cours

- **Savoir écrire un programme simple** formé de séquence d'instructions, d'instructions conditionnelles, de boucles, de fonctions et de procédures.
- **Manipuler des données** de nature diverse comme les entiers, les chaînes de caractères, les tableaux, les images, ...
- **Comprendre ce que produit l'exécution** d'un programme (simple).

Objectifs du cours

- **Savoir écrire un programme simple** formé de séquence d'instructions, d'instructions conditionnelles, de boucles, de fonctions et de procédures.
- **Manipuler des données** de nature diverse comme les entiers, les chaînes de caractères, les tableaux, les images, ...
- **Comprendre ce que produit l'exécution** d'un programme (simple).
- **Savoir trouver une erreur** dans un programme (simple), et la corriger.

Une équipe pédagogique

et il en manque beaucoup...

Fonctionnement d'IP1

http://www.ip101.fr

Séances d'interaction pédagogique

- Un cours d'amphi toutes les deux semaines.
- Un cours/travaux-dirigés par semaine.
- Deux séances de travaux pratiques par semaine.

Travail personnel

- Allez à tous les cours, soyez attentifs et actifs !
- Programmez le plus souvent possible grâce au site :

http://www.hackojo.org

L'amphi

L'amphi *présente les concepts du cours* et
revient sur vos di□cultés.

Comment ?

- Par une interaction entre l'enseignant et les étudiants sur les sujets suivants de la programmation :
  1. Qu'est-ce que la programmation ?
  2. Qu'est-ce que la programmation procédurale (en Java) ?
  3. Comment prédire le comportement d'un programme ?
  4. Qu'est-ce qu'un programme qui calcule bien ?
  5. Qu'est-ce qu'un programme qui calcule vite ?
  6. A quoi s'intéresse l'Informatique ?
- Par la **présentation** et la **correction** (l'amphi suivant) de **défis** que vous devez faire d'un amphi à l'autre en utilisant le Dojo de Programmation.
- Un partiel, un examen de première session, un examen de seconde session.

Le cours/td

Le cours/td *précise les concepts du cours* et *vous aide à les maîtriser*.

Comment ?

- Une partie “cours” avec des exercices corrigés par l’enseignant.
- Une partie “*Do It Yourself* ” qui sert à vous préparer de façon autonome à l’examen. (Mais vous pouvez poser des questions sur ces exercices à l’enseignant.)
- Un contrôle continu.

Les travaux pratiques

Les travaux pratiques *vous aide à devenir des programmeurs*

Comment ?

- Des exercices corrigés automatiquement en ligne sur le Dojo.
- Des enseignants qui répondent à vos questions de programmation et d'usage des logiciels utilisés dans le cours.
- Un contrôle continu, des examens sur machine.

Evaluation

$$
\begin{aligned}
\text{CC} &= \frac{\text{TD}+\text{TP}}{2} \\
\text{Examen} &= max(\text{Examen}_1, \frac{\text{Partiel}+\text{Examen}_1}{2}) \\
\text{Session}_1 &= \frac{3\square\text{Examen}+\text{CC}}{4} \\
\text{Session}_2 &= max(\text{Examen}_2, \frac{3\square\text{Examen}_2+CC}{4})
\end{aligned}
$$

Le prof a dit :

« Le programme est lui-même représenté en binaire dans la machine. »

Le prof a dit :

« Le programme est lui-même représenté en binaire dans la machine. »

Ce langage de représentation est appelé le **langage machine**.

Le prof a dit :

« Le programme est lui-même représenté en binaire dans la machine. »

Ce langage de représentation est appelé le **langage machine**.

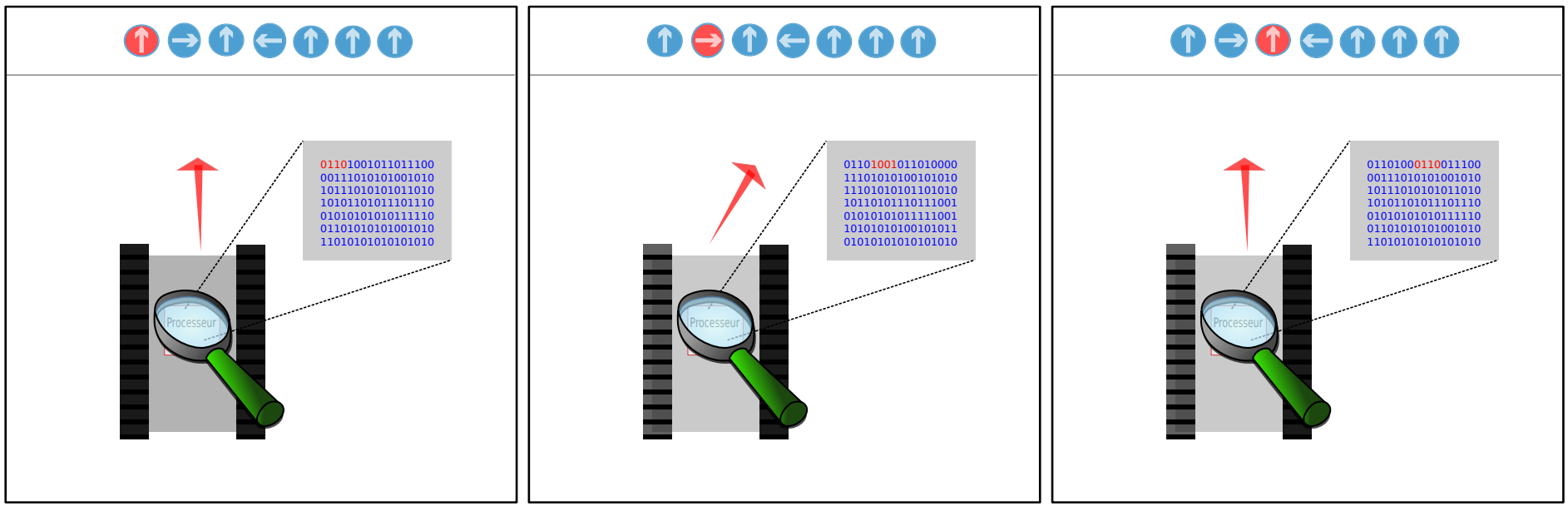
Comment écrire des programmes en langage machine ?

La compilation

Une question étrange ?

Peut-on écrire un compilateur dans le langage qu'il compile ?

Que se passe-t-il pendant l'exécution d'un programme ?



Qu'est-ce qu'un langage de programmation ?

Définition

Un langage de programmation est un **langage artificiel** dont la syntaxe est spécifiée par une **grammaire formelle** et par une **sémantique** telle que chaque programme est **univoque**.

Remarques

Généralement, les langages de programmation sont écrits et lus par des humains. Ils sont donc tout autant des langages support du raisonnement sur un processus de calcul que des langages pour programmer une machine.

Il existe des dizaines de milliers de langages de programmation.

Quiz : Avez-vous déjà entendu parler des langages suivants ?

Le langage d'IP1

Nous allons étudier un **sous-ensemble de Java**.

Le langage d'IP1

Nous allons étudier un **sous-ensemble de Java**.

Remarques

Java est le langage **orienté objet** le plus utilisé. Il a été créé dans les années 90 par l'entreprise américaine SUN.

Le langage d'IP1

Nous allons étudier un **sous-ensemble de Java**.

Remarques

Java est le langage **orienté objet** le plus utilisé. Il a été créé dans les années 90 par l'entreprise américaine SUN.

Nous n'utiliserons qu'une minuscule partie du langage appelé **sous-langage procédural**. C'est un langage que vous retrouverez dans d'autres langages de programmation.

Le langage d'IP1

Nous allons étudier un **sous-ensemble de Java**.

Remarques

Java est le langage **orienté objet** le plus utilisé. Il a été créé dans les années 90 par l'entreprise américaine SUN.

Nous n'utiliserons qu'une minuscule partie du langage appelé **sous-langage procédural**. C'est un langage que vous retrouverez dans d'autres langages de programmation.

On peut utiliser ce sous-langage pour :

- décrire des calculs arithmétiques ou sur des tableaux de données.
- écrire et lire dans des variables,
- boucler sur certaines instructions,
- exécuter des instructions plutôt que d'autres en fonction des entrées.
- créer des procédures et les appeler.

Un premier programme pour notre robot

```
avance ();
```

Les séquences d'instructions

On peut enchaîner plusieurs instructions en les écrivant les unes à la suite des autres.

Un second programme pour notre robot

```
avance ();
tourne (90);
avance ();
tourne (-90);
```

Comme pour une calculatrice

En fait, vous avez tous déjà utilisé un langage de programmation : celui des expressions arithmétiques de vos calculatrices.

JAVA permet aussi de faire des calculs arithmétiques. Seulement, la plupart des nombres qu'on manipule en JAVA sont bornés. Par exemple, l'ensemble de tous les entiers compris entre -2147483648 ($= -2^{31}$) et 2147483647 ($= 2^{31} - 1$) est appelé le **type int**.

Quiz : D'après vous, qu'arrive-t-il quand on dépasse les bornes pendant un calcul sur un entier borné ?

Un troisième programme pour notre robot

```
tourne ((90 + 90) * 2 / 4);
```

Utilisation de la mémoire

Les ordinateurs ont des mémoires qui leur permettent de stocker des informations, plus ou moins temporairement. Le langage JAVA permet de déclarer des zones mémoires, appelées variables, en leur donnant un type, un nom et une valeur initiale.

Un quatrième programme pour notre robot

```
int gauche = 45;
tourne (gauche * 2);
```

Les instructions conditionnelles

Les programmes doivent agir **en fonction de leur entrée**. Les instructions conditionnelles sont formées d'une condition, d'instructions à exécuter si la condition est vraie et d'instructions à exécuter si la condition fausse.

Un cinquième programme pour notre robot

```
if (couleur_est (rouge)) {
  tourne (gauche);
} else {
  tourne (droite);
}
```

Les boucles

Les ordinateurs ne s'ennuient pas. On peut les faire boucler des milliards de fois sur seulement quelques instructions sans que cela pose problème.

Il existe deux types de boucles :

1. Les boucles de répétitions bornées
2. Les boucles non bornées.

Un sixième programme pour notre robot

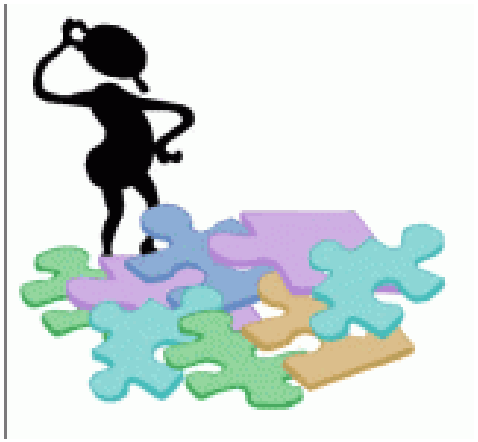
```
for (int i = 0; i < 10; i++) {
  avance ();
}
```

Un septième programme pour notre robot

```
while (couleur_est(blanc)) {
  tourne (1);
}
```

Les ordinateurs servent à résoudre des problèmes complexes

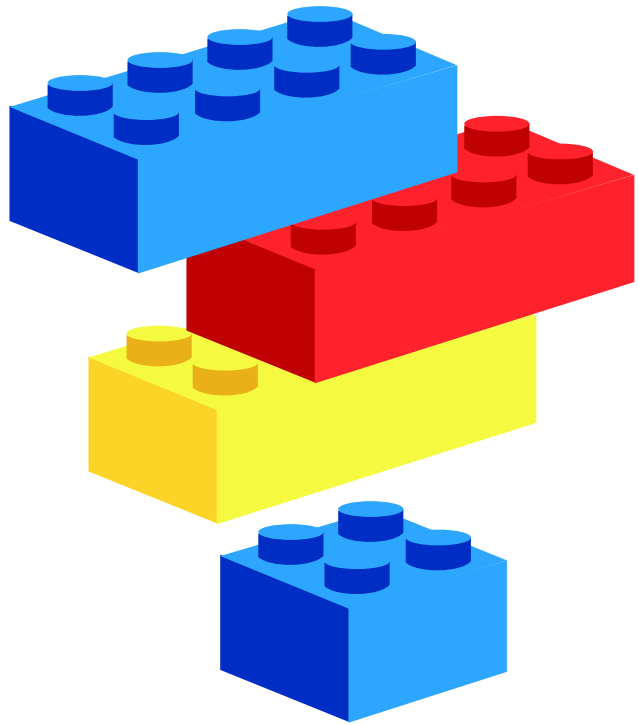
- Aller sur la Lune.
- Faire fonctionner Internet.
- Calculer des images de synthèse.
- ...



Pour résoudre ces problèmes, on les **décompose** en problèmes un peu moins complexes, que l'on décompose en problèmes encore un peu moins complexes, et ainsi de suite, jusqu'à obtenir des problèmes très simples.

Comme avec les LEGOs, on construit des briques et les réutilise !

Un logiciel, c'est un ensemble de briques qui résout un problème.



En programmation, ces briques sont appelées des fonctions ou procédures.

Quiz : Combien de lignes de codes dans Android ?

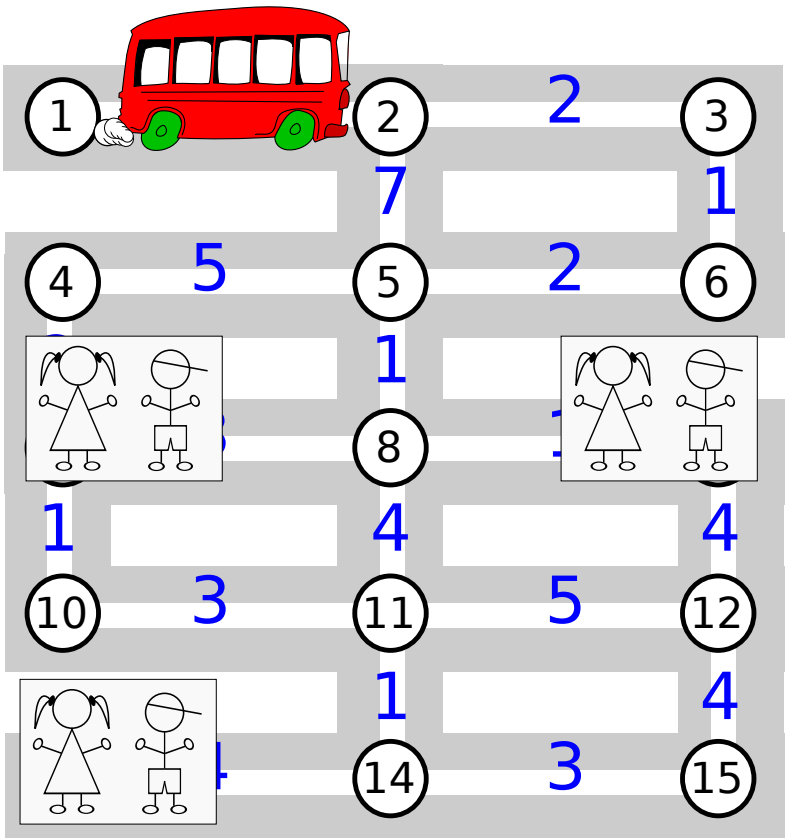
Comment construire des logiciels ?

Programmons un système de navigation automatique !

Un problème complexe : un système de navigation

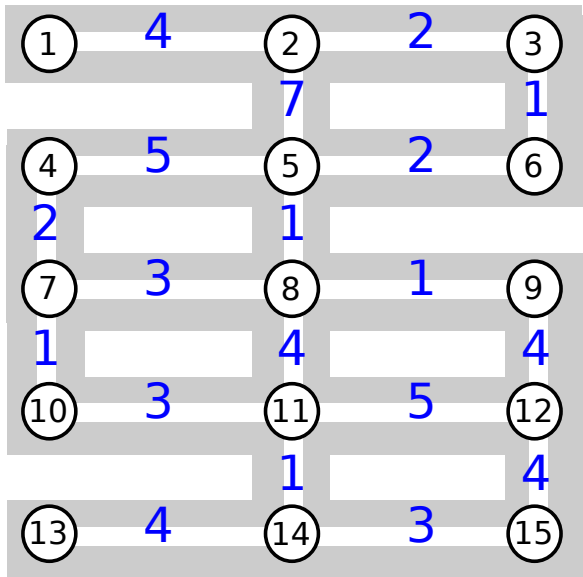
- Le bus doit aller chercher les enfants à leur arrêt.
- Il doit suivre les lignes blanches.
- En bleu est indiqué le temps nécessaire pour se rendre d'une intersection à une autre.

$\Rightarrow$ Comment mettre le moins de temps possible pour e□ectuer ce trajet ?



Représentation des **données** du problème

La carte



L'itinéraire

Un itinéraire de 1 à 9 :
[1; 2; 5; 8; 9]

Un autre itinéraire de 1 à 9 :
[1; 2; 3; 6; 5; 8; 9]

Un itinéraire de 5 à 15 :
[5; 8; 9; 12; 15]

Un autre itinéraire de 5 à 15 :
[5; 4; 7; 10; 11; 14; 15]

Premier problème : Comment suivre un itinéraire ?

- Notre robot sait aller à gauche, à droite ou tout droit.
- Comment suivre un itinéraire qui est une liste de numéros d'intersection ?

Une idée ?

Deux sous-problèmes

1. Aller d'un point d'intersection à un autre en suivant une direction donnée.
2. Choisir la bonne direction pour aller d'un point A à un point (voisin) B.

Aller d'un point à un autre en suivant une direction donnée

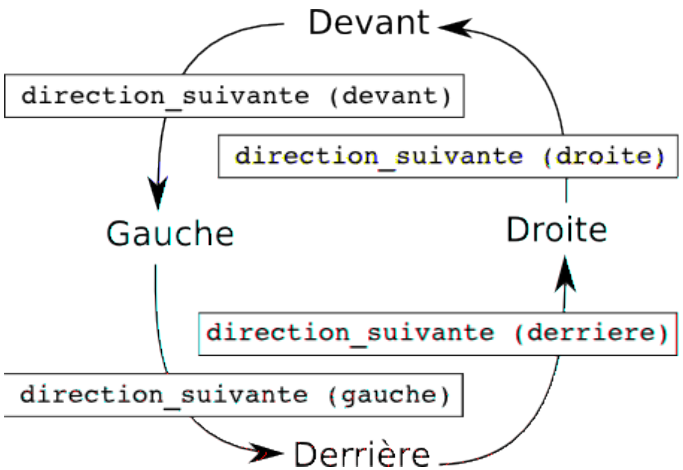
```
while (couleur_est (blanc) || couleur_est(noir)) {
  while (couleur_est (blanc)) {
    avance ();
  }
  cherche_blanc ();
}
```

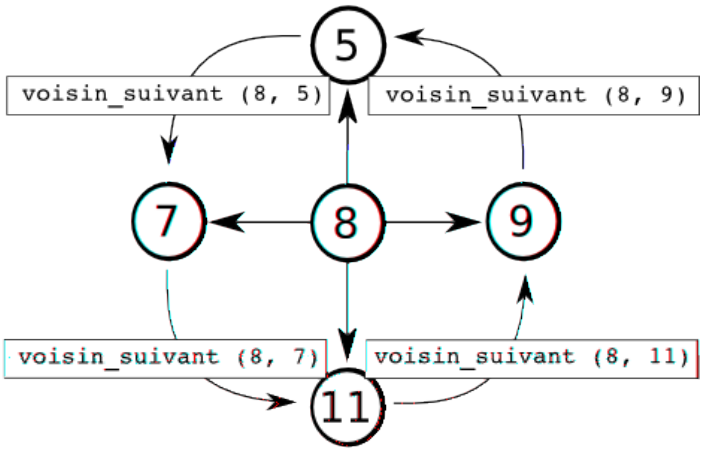
Quiz : Pensez-vous que le programme précédent a une erreur ?

Aller d'un point à un autre en suivant une direction donnée (corrigé)

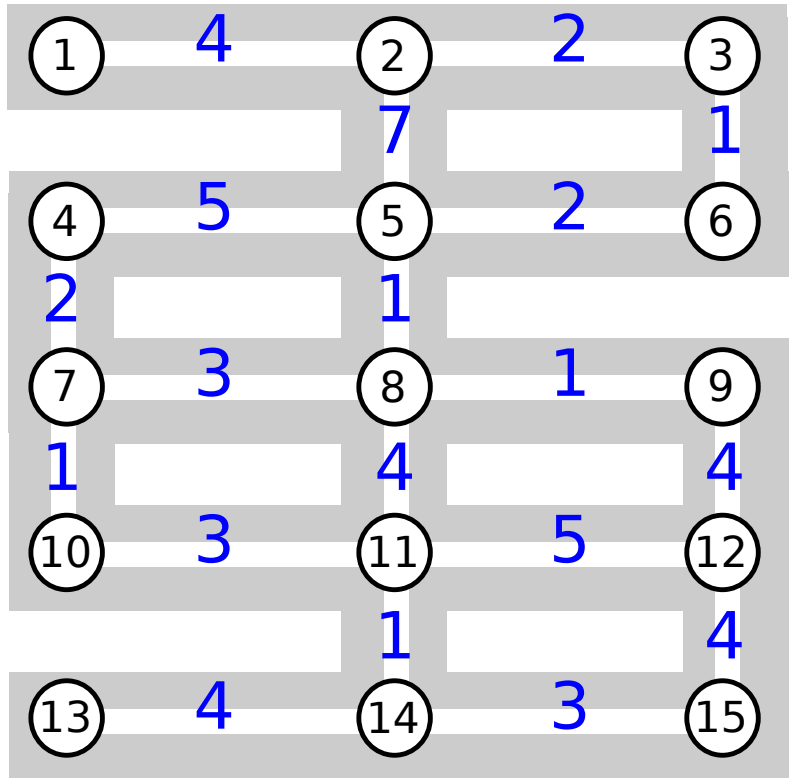
```
while (couleur_est (blanc) || couleur_est(noir)) {
  while (couleur_est (blanc)) {
    avance ();
  }
  cherche_non_noir ();
}
```

Choisir la bonne direction





Trouver l'itinéraire le plus rapide



En essayant toutes les solutions ...

[1; 2; 3; 6; 5; 8; 9; 12; 15; 14; 11; 10; 7]	27
[1; 2; 5; 8; 9; 12; 15; 14; 11; 10; 7]	29
[1; 2; 3; 6; 5; 8; 9; 12; 11; 10; 7]	24
[1; 2; 5; 8; 9; 12; 11; 10; 7]	26
[1; 2; 3; 6; 5; 8; 11; 10; 7]	18
[1; 2; 5; 8; 11; 10; 7]	20
[1; 2; 3; 6; 5; 8; 7]	13
[1; 2; 5; 8; 7]	15
[1; 2; 3; 6; 5; 4; 7]	16
[1; 2; 5; 4; 7]	18

10 chemins possibles du point 1 au point 7

Aïe !

J'ai fait quelques calculs . . .

Nombre N d'emplacements	Nombre de chemin de la position 1 à N
15	~ 26

Aïe !

J'ai fait quelques calculs . . .

Nombre N d'emplacements	Nombre de chemin de la position 1 à N
15	~ 26
25	~ 5000

Aïe !

J'ai fait quelques calculs . . .

Nombre N d'emplacements	Nombre de chemin de la position 1 à N
15	~ 26
25	~ 5000
36	~ 1000000

Aïe !

J'ai fait quelques calculs . . .

Nombre N d'emplacements	Nombre de chemin de la position 1 à N
15	~ 26
25	~ 5000
36	~ 1000000

- Pour la carte de Paris, il y a plus de chemins que d'atomes dans l'Univers !

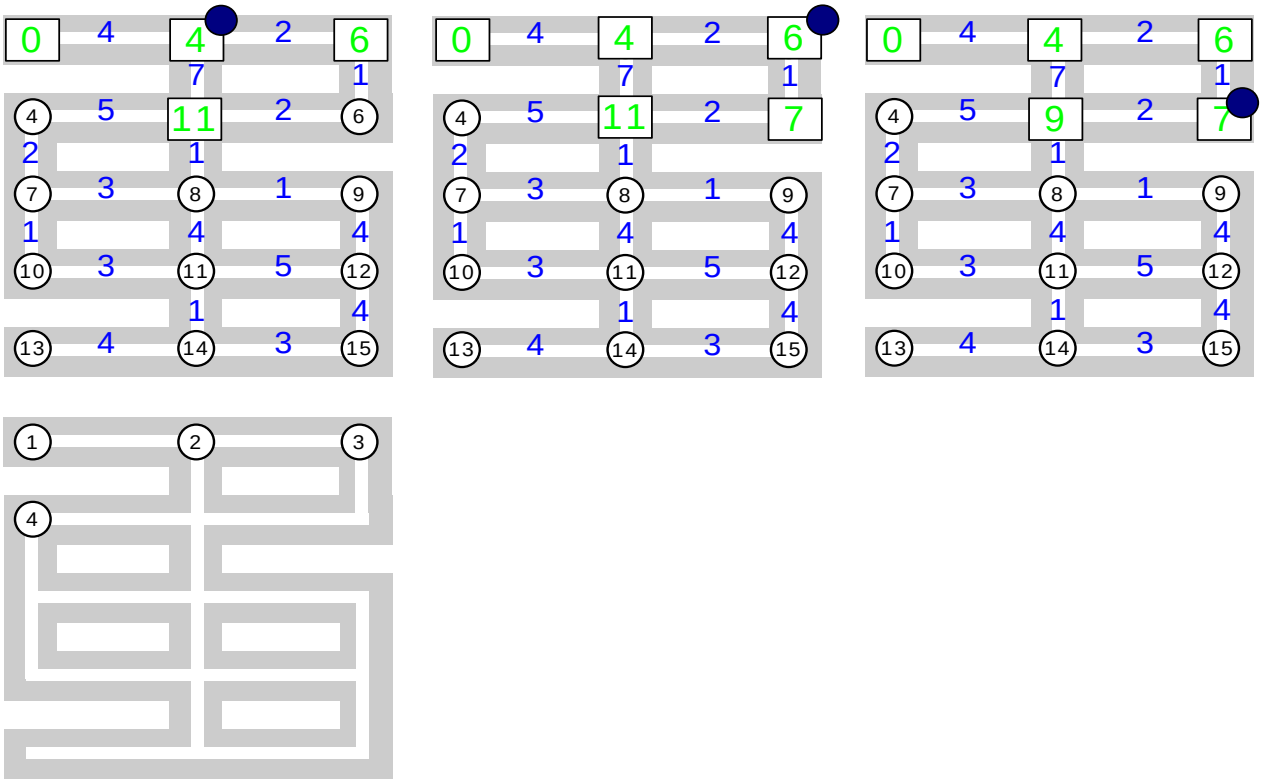
Aïe !

J'ai fait quelques calculs . . .

Nombre N d'emplacements	Nombre de chemin de la position 1 à N
15	~ 26
25	~ 5000
36	~ 1000000

- Pour la carte de Paris, il y a plus de chemins que d'atomes dans l'Univers !

$\Rightarrow$ Peut-être qu'il existe une meilleure façon de résoudre ce problème ?



L'**algorithme** de Dijkstra

```java
class Hello {
  public static void main (String[] argv) {
      System.out.println ("Hello IP1!");
  }
}
```

- On compile grâce à la commande :

  ```
  % javac Hello.java
  ```

- On exécute grâce à la commande :

  ```
  % java Hello
  ```

(Ces opérations n'auront plus de secret pour vous après quelques TPs.)

Le premier défi

Apprendre la syntaxe du langage de programmation
. . . en sauvant une fourmi.