



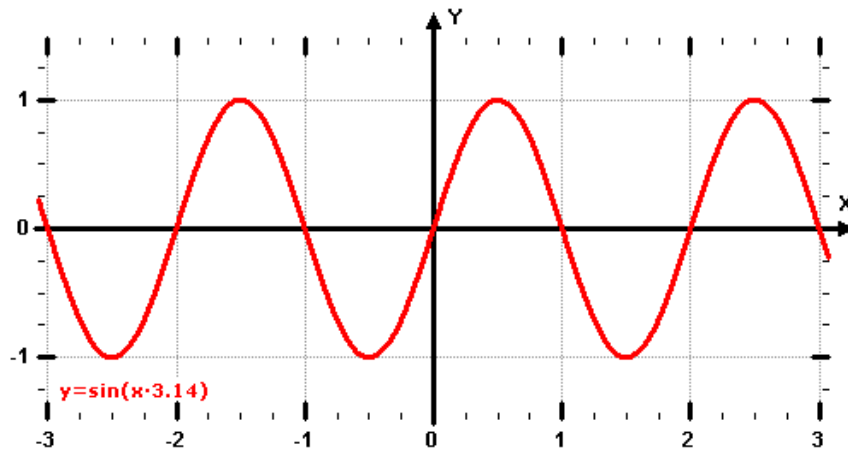
ISEN CIR ³

Théorie des langages

Le projet

- Le projet sera réalisé par un groupe de 5 personnes.
- Le rendu (sous format zip) contiendra:
 - Le code source
 - Manuel du langage (en markdown) avec exemples de code
 - Mini rapport (déroulement, rôles, difficultés, gestion du projet, ...)
- Il est à rendre avant la fin de la dernière séance de projet, le **9 décembre 2021** à 12h00.
- Cette dernière séance sera dédiée aux soutenances et démonstrations de l'outil et du langage, groupe par groupe, pour l'évaluation du projet.

1. Développement d'un outil permettant, au travers d'un mini-langage de **tracer des graphes** de fonctions mathématiques.



2. Développement d'un mini-langage pour apprendre la programmation de façon ludique par la **manipulation d'agents** ayant des missions qui augmentent en difficultés

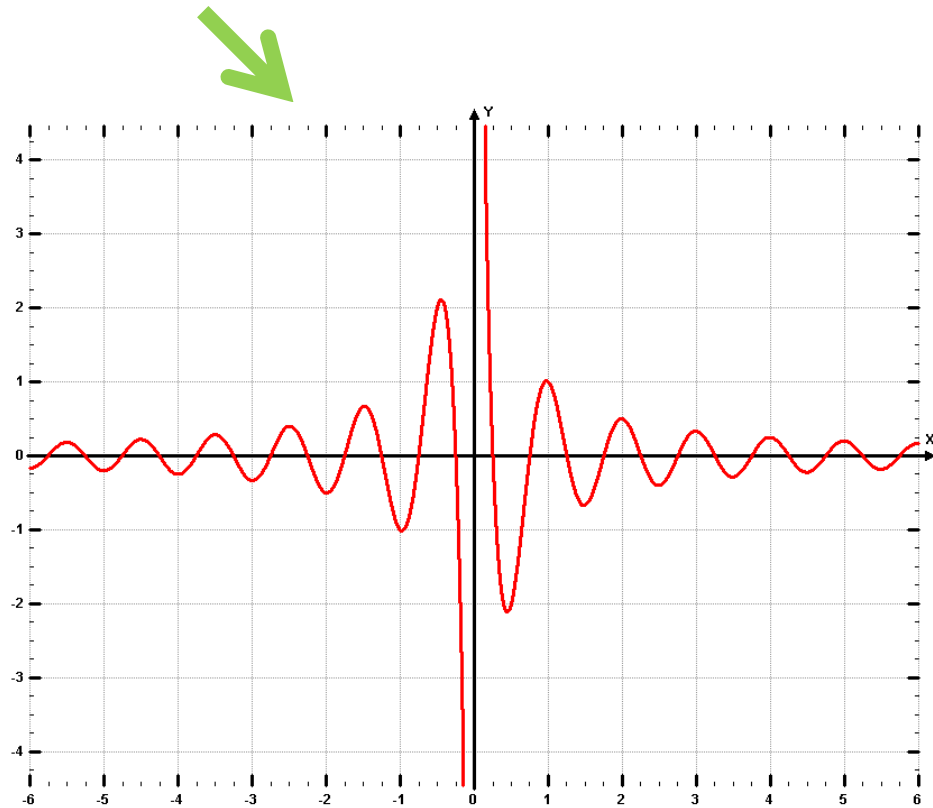
3. Développement d'un langage de **programmation en français**

<http://langagelinotte.free.fr/wordpress/>

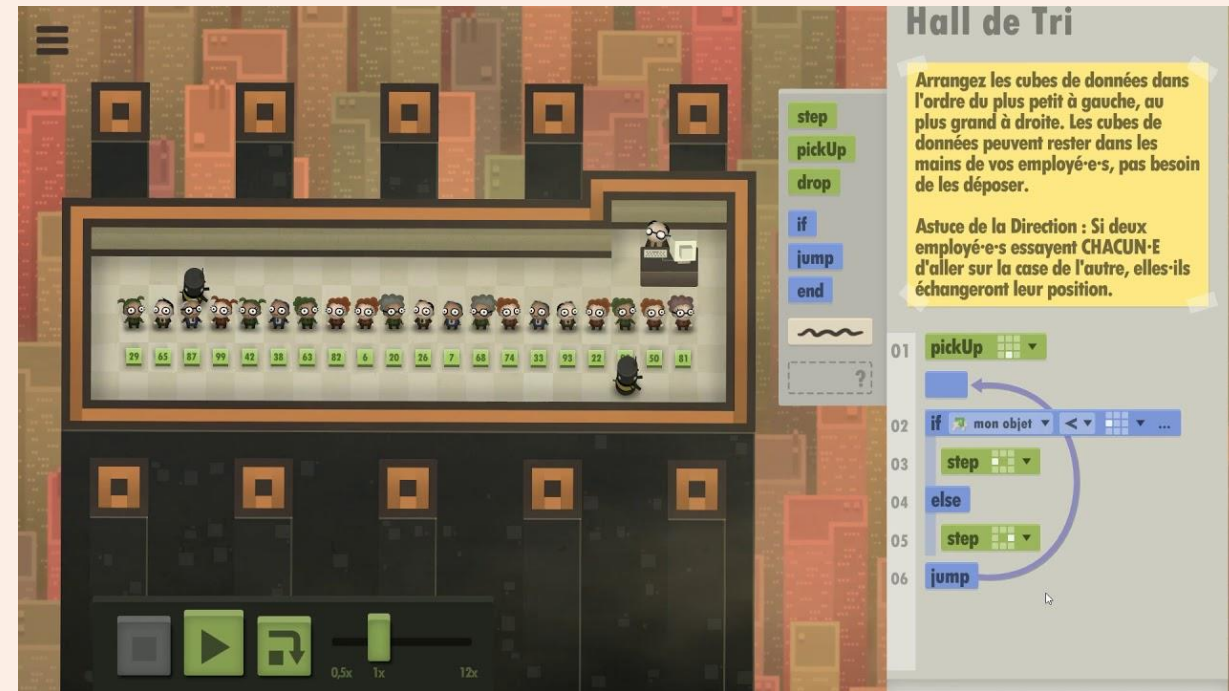
```
principale :  
  s valant hasard(500)  
  b & r est un nombre  
  "Nombre caché entre 0 et 500" !  
  tant que r != s  
    incrémente b  
  questionne r sur "Une idée ?"  
  si r > s, "Trop grand !" !  
  sinon si r < s, "Trop petit !" !  
ferme  
"Trouvé en {b} coups !" !
```

4. ...

$$y = \cos(2 \cdot 3.14 \cdot x) / x$$



Un code pour trier



L'expression minimale reconnue par le langage est :

$$y = f(x)$$

Avec $f(x)$: une expression mathématique contenant :

- La variable : x
- Les opérateurs de bases : $+$, $-$, $/$, $*$, ...
- Les fonctions mathématiques : \sin , \cos , \tan , \arccos , \arcsin , \arctan , \sinh , \cosh , \tanh , \log , \exp , pow , sqrt , abs , ...
- Les parenthèses : ()

- Des instructions pour contrôler l'agent et lui donner des informations sur son environnement :

```
Take box
If box > table_box
    Drop box
Else
    Put box on table
```

- $y = \sin(x)$
- $y = X^2 + 2 * x + 1$
- $y = (x^3)/3 - 1.5 * x^2 + x$
- $y = \text{atan}(\text{abs}(3.14 * x))$

- <https://robocode.sourceforge.io/>
- <https://codecombat.com/>
- <https://www.codingame.com/start>
- <https://vim-adventures.com/>

- Le langage peut être plus évolué pour exprimer : des constantes, des fonctions intermédiaires, des paramètres d'affichage, etc.

```
Pi = 3.14
```

```
F(x) = sin(Pi*x)
```

```
G(x) = cos(Pi*x)
```

```
y = F(x) + G(x)
```

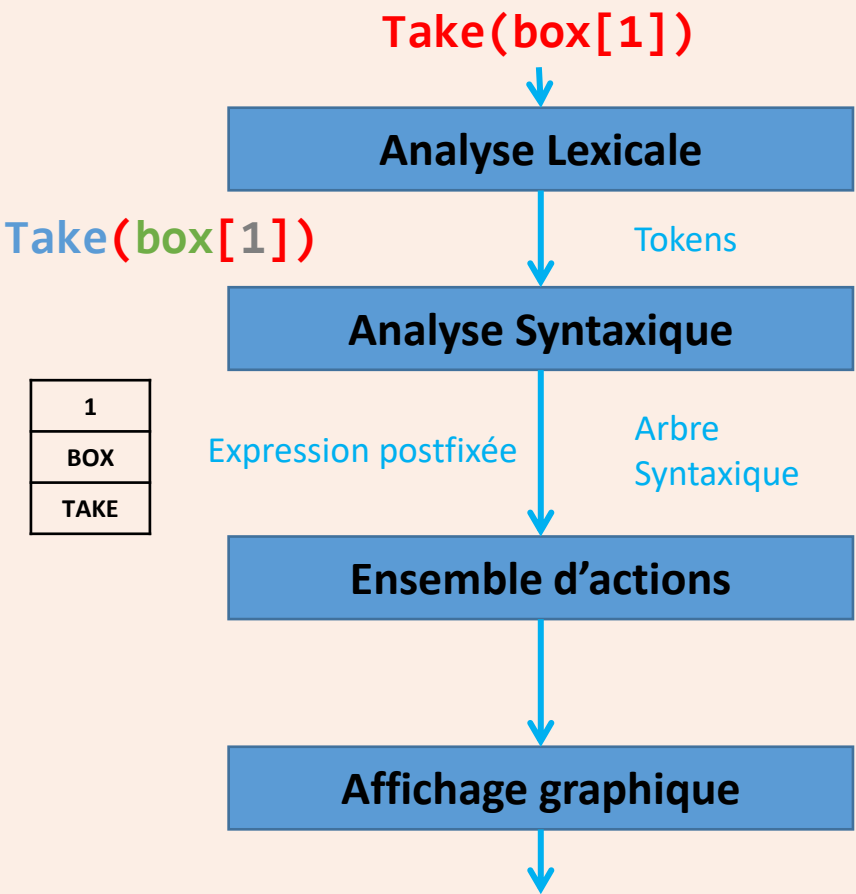
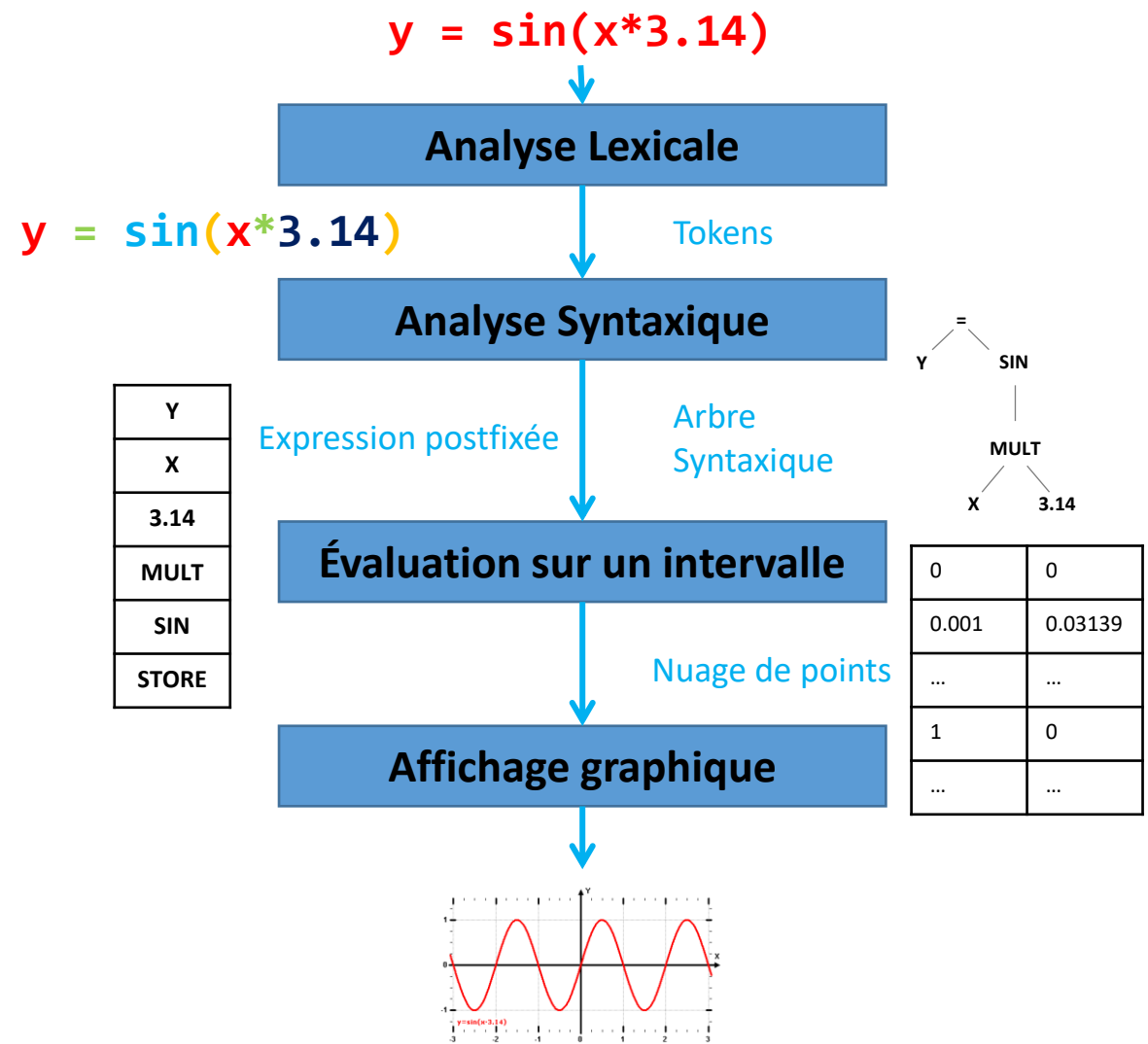
```
Draw F color red
```

```
Draw G color blue
```

```
Draw y color green in [-5 .. 5]
```

- <https://store.steampowered.com/app/370360/TIS100/>

- *flex* (générateur d'analyseur lexicaux)
- *bison* (générateurs d'analyseurs syntaxiques)
- Le compilateur *C/C++* de votre choix
- Autres outils et langages de votre choix permettant d'interagir avec les outils précédents. (*JavaScript, SVG, OpenGL, SDL, SFML, librairies, ...*).



- **Analyse lexicale**: À réaliser en *flex*
- **Analyse syntaxique**: À réaliser sous *bison*
- **Module d'évaluation**: À développer en *C* ou *C++*
Sur un intervalle d'affichage découpé en fonction de la résolution de l'écran, évaluer la valeur de la fonction pour sortir un nuage de points (x, y)
- **Module d'affichage**: Afficher la fonction en liant les points du nuage obtenu, mais aussi : les axes, la grille, les graduations, les labels, ...

- **OpenGL / SDL** (C/C++)

Rendu de l'affichage graphique dans une fenêtre avec les librairies OpenGL ou SDL

- **WEB** (SVG/JavaScript)

Rendu au format SVG (visualisable sur browser) avec code en JavaScript pour la partie interactive.

- **QT** (C++)

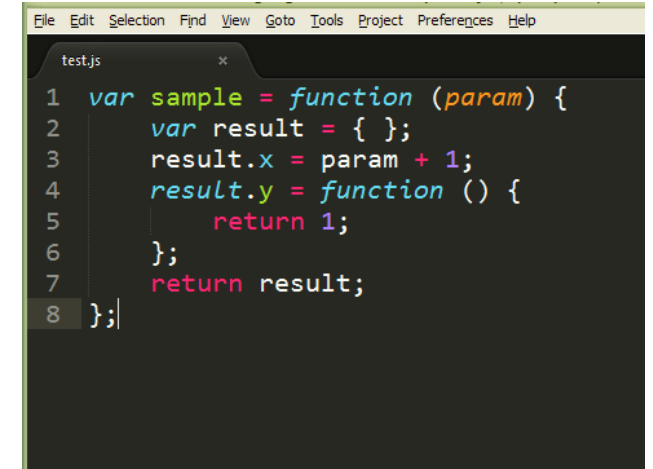
Interface interactive (IDE) à base de contrôles pour la saisie, le paramétrage, le zoom, le pan, ...

- **Autres propositions**

- Possibilité de dessiner plusieurs graphes à la fois.
- Gestion des axes et des échelles.
- Possibilité de choisir un intervalle à afficher (ou développer un zoom/pan interactif)
- Aire sous la courbe.
- Obtenir interactivement les coordonnées d'un point sur la courbe.
- Gestion des intervalles de définitions.
- Dessin dans des espaces cartésiens ou polaires.
- ...

- Affichage graphique
- Pouvoir agir sur plusieurs agents
- Niveaux de difficultés progressifs

- Produire le fichier de configuration qui permet de colorer syntaxiquement votre langage dans « sublime text » lors de l'édition du code



```
File Edit Selection Find View Goto Tools Project Preferences Help
test.js
1 var sample = function (param) {
2   var result = { };
3   result.x = param + 1;
4   result.y = function () {
5     return 1;
6   };
7   return result;
8 };
```

- <https://www.sublimetext.com/docs/syntax.html>
- <http://ilkinulas.github.io/programming/2016/02/05/sublime-text-syntax-highlighting.html>