



## Bienvenue dans la PLM, l'exerciseur du programmeur

### Liste des leçons

La PLM vous apprendra l'art de la programmation par le biais d'exercices interactifs réalisables à votre propre rythme.

Choisissez une leçon sur la gauche, et commencez à coder! Si vous débutez, sélectionnez celle intitulée "welcome": vous y apprendrez les bases de la programmation

## Select Level :



welcome



maze



turmites



turtleart



sort.basic



sort.dutchflag



sort.baseball



sort.pancake



recursion.cons



recursion.logo



recursion.hanoi



bat.string1



### Back to the Game ?

lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Continue Quest



Profile

Continue JAVA

explore

FR ▼

LVL 3

## Bienvenue dans la PLM, l'exerciceur du programmeur

### Liste des leçons

La PLM vous apprendra l'art de la programmation par le biais d'exercices interactifs réalisables à votre propre rythme.

Choisissez une leçon sur la gauche, et commencez à coder! Si vous débutez sélectionnez celle intitulée "welcome": vous y apprendrez les bases de la programmation.

### Select Level :



welcome



maze



sort.basic



sort.dutchflag



recursion.cons



recursion.log



### Back to the Game ?

lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Continue



Username

Expert Mode

## USER INFOS

JAVA

LVL 3: 75%



Continue

Scala

LVL 1: 25%



Continue

Blocky

LVL 5: 55%



Continue

Python

LVL 1: 0%



Continue

HELP? / CONTACT US



PLM



Profile

Select exos ▼

Switch langage ▼

FR ▼

LVL 3



Username

← Beginner Mode

## USER INFOS



List of Lessons

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item

Item



### Bienvenue dans le monde des buggles

Vous venez de lancer la PLM, l'exerciseur du programmeur (Programmer's Learning Machine). Il s'agit d'une plate-forme pédagogique destinée à simplifier l'apprentissage de la programmation. Il est constitué d'un ensemble d'exercices groupés par leçons, pour vous permettre de progresser à votre rythme. Pour l'instant, l'outil est configuré pour vous permettre de programmer en Java, mais vous pouvez changer de langage de programmation grâce au menu déroulant se trouvant dans la barre du haut si vous le désirez.

Dans cette première leçon, les buggles guideront vos premiers pas en programmation.

### Les buggles ? Qu'est ce que c'est ??

Les buggles sont de petites bêtes qui obéissent aux ordres que vous leur donnez. Dans chaque exercice, vous devez donner des ordres à vos buggles pour faire en sorte que le monde ressemble à l'objectif de l'exercice. Par exemple dans cet exercice, vous devez instruire votre buggle pour qu'elle avance d'un pas. Observez la différence entre l'état initial et l'objectif en utilisant les onglets ci-dessous. En fonction de l'exercice (et vos réglages), vous devez écrire vos programmes dans l'un des langages de programmation suivant : Java, Python, Scala ou C (en fonction de l'exercice).

### L'environnement de travail

Avant d'aller plus loin, familiarisez vous avec l'environnement. Observez les différents éléments composant la fenêtre, placez la souris dessus pour voir les bulles d'aide, et essayez les pour voir ce qu'ils font. La section Résultats de l'exécution en bas est la console : c'est là que les erreurs et messages sont affichés.

Si votre code contient des erreurs (et les programmes finissent toujours par en contenir), l'ordinateur affichera les messages d'erreur sur la console. Vous devez bien entendu corriger ces erreurs pour réussir les exercices. Les messages affichés peuvent paraître inquiétant au premier abord, mais pas de panique. Le compilateur est seulement pas très doué pour communiquer, il n'est pas fondamentalement méchant. En regardant de plus près, la solution pour corriger les problèmes est inscrite dans ces messages cryptiques. Vous verrez, avec un peu d'entraînement, vous vous y habituerez.

### Que dois-je faire ?

Il est temps d'écrire votre premier programme. Demandez simplement de faire un pas en avant dans le panneau Éditeur de code. Pour cela, écrivez simplement le code suivant :

```
avance();
```

N'oubliez pas le ; final, car cela indique au compilateur que l'instruction est terminée (oui, les ordinateurs sont si stupides qu'il faut leur préciser des choses aussi simples).

Une fois terminé, cliquez sur le bouton se trouvant dans la section Cas de tests pour lancer votre programme. Quand cela fonctionne, vous pouvez passer à l'exercice suivant. Pour changer d'exercice, cliquez sur le bouton se trouvant en haut à gauche de votre écran pour faire apparaître le menu latéral. Sélectionnez ensuite le prochain élément de la liste d'exercices.



Accept Quest

### The JAVA Boss





## Quest

## Help / API

## Objective

## L'environnement de travail

Avant d'aller plus loin, familiarisez vous avec l'environnement. Observez les différents éléments composant la fenêtre, placez la souris dessus pour voir les bulles d'aide, et essayez les pour voir ce qu'ils font. La section Résultats de l'exécution en bas est la console : c'est là que les erreurs et messages sont affichés.

Si votre code contient des erreurs (et les programmes finissent toujours par en contenir), l'ordinateur affichera les messages d'erreur sur la console. Vous devez bien entendu corriger ces erreurs pour réussir les exercices. Les messages affichés peuvent paraître inquiétant au premier abord, mais pas de panique. Le compilateur est seulement pas très doué pour communiquer, il n'est pas fondamentalement méchant. En regardant de plus près, la solution pour corriger les problèmes est inscrite dans ces messages cryptiques. Vous verrez, avec un peu d'entraînement, vous y habituerez.

## Que dois-je faire ?

Il est temps d'écrire votre premier programme. Demandez simplement de faire un pas en avant dans le panneau Éditeur de code. Pour cela, écrivez simplement le code suivant :

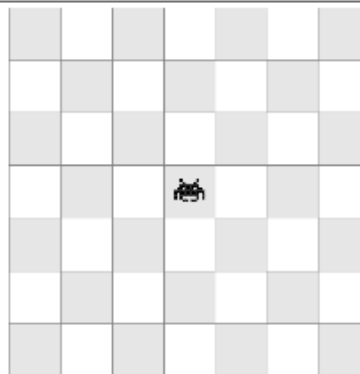
```
avance();
```

Try Again ?

```
1 <>
2
3
4
5
6
7
8
9
```



RUN



Vitesse d'animation:

## Quest Result



## Quest

## Help / API

## Objective

## Le monde des Buggles

Ce monde a été inventé par Lyn Turbak, du Wellesley College. Il est peuplé de Buggles, petites bêtes qui comprennent des ordres simples, et offre de nombreuses possibilités d'interaction avec le monde : prendre ou poser des objets, colorier le sol, se cogner à des murs, etc.

Méthodes comprises par les buggles

**Bouger**

(voir aussi la note sur les exceptions, plus bas)

**Tourner à gauche****Tourner à droite****Se retourner****Avancer****Reculer**

void gauche()

void droite()

void retourne()

void avance() or void avance(int nbPas)

void recule() or void recule(int nbPas)

**Obtenir l'abscisse****Obtenir l'ordonnée****Changer l'abscisse****Changer l'ordonnée****Changer la position**

int getX()

int getY()

void setX(int x)

void setY(int y)

void setPos(int x, int y)

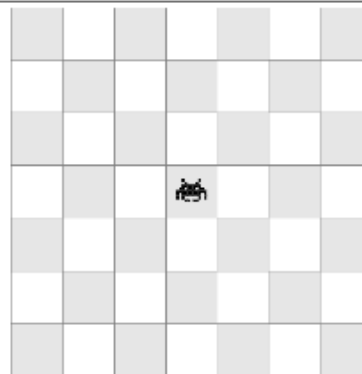
L'origine (0, 0) est dans le coin en haut à gauche, comme c'est souvent le cas en informatique.

Try Again ?

```
1 <>
2
3
4
5
6
7
8
9
```



RUN



Vitesse d'animation:



## Quest Result



PLM



Profile

Continue JAVA

explore

FR ▼

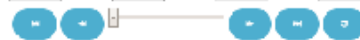
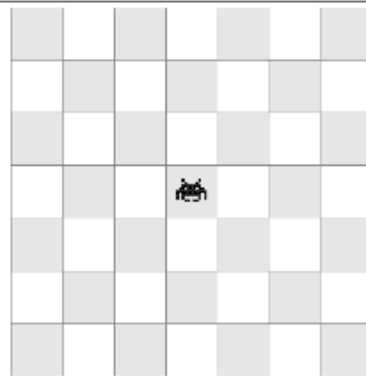
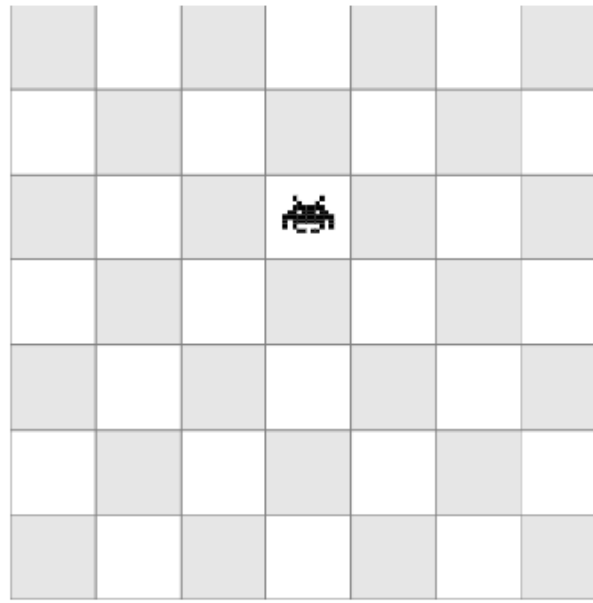
LVL 3



Quest

Help / API

Objective



Vitesse d'animation:



Try Again ?

```
1 </>  
2  
3  
4  
5  
6  
7  
8  
9
```



RUN

Quest Result





PLM



Profile

Continue JAVA

explore

FR ▼

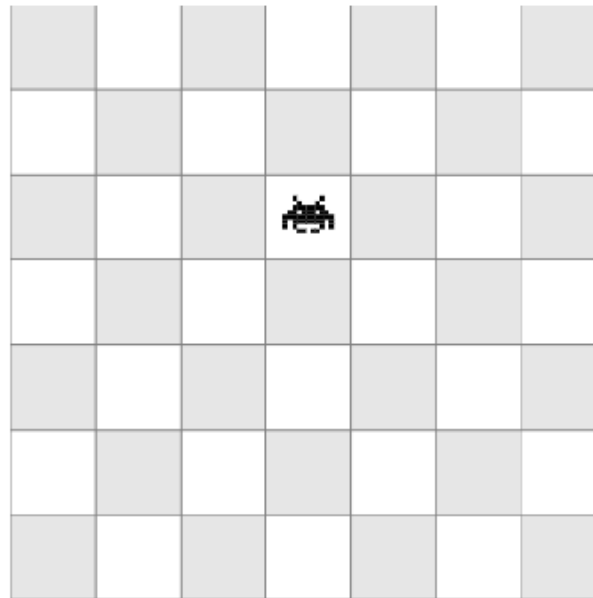
LVL 3



Quest

Help / API

Objective



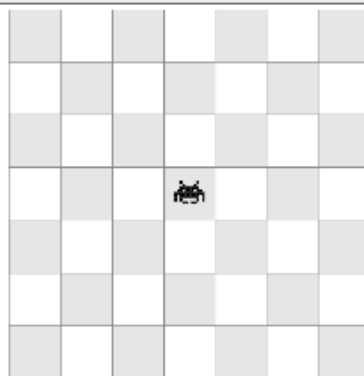
Demo

Try Again ?

```
1 </>
2
3
4
5
6
7
8
9
```



RUN



Vitesse d'animation:



Quest Result

A paragraph of **text** with an [unassigned link](#).  
A second row of text with a [web link](#)

SUCCESS





Quest

Help / API

Objective


```
1 </>
2
3
4
```

Try Again ?

## Exercice réussi \o/

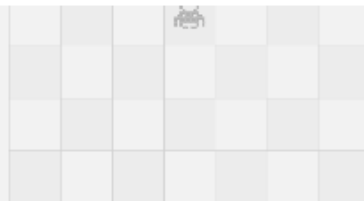


Congratulations, you pass this exercise!

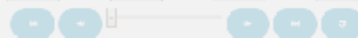
Voulez-vous tenter l'exercice suivant?

Pas encore

Bien sûr



Vitesse d'animation:





## Perdu dans la Moria

Vos buggles sont coincées dans la mine de la Moria ! Des rochers bloquent la sortie, et il va falloir dégager le chemin pour passer. Bon, ok, ce ne sont pas vraiment des rochers mais juste des biscuits, et vos buggles pourraient facilement passer au dessus sans se fatiguer. Mais il est probablement plus simple de programmer vos buggles pour qu'elles déplacent ces «rochers» plutôt que de tenter de les convaincre de passer à la suite sans avoir résolu le problème. Les nains sont souvent un peu têtus...

Donc, il vous faut trouver le premier biscuit en travers de votre chemin (marchez vers l'est jusqu'à vous trouver au dessus d'un biscuit), le ramasser, et retourner à l'autre extrémité du couloir pour le déposer (marchez vers l'ouest jusqu'à vous trouver au dessus d'un biscuit puis reculez d'un pas). Il faut ensuite faire de même pour tous les biscuits jusqu'à trouver la sortie. Une fois ceci fait, marchez vers l'air frais comme dans le monde objectif, cela ne sera plus qu'à quelques pas de là.

Pour compliquer encore les choses, vous devez écrire un programme qui fonctionne pour toutes les buggles, même si écrire un programme spécifique à chacune serait peut-être un peu plus simple. Il semblerait qu'utiliser une boucle while soit la seule solution pour déplacer vos buggles correctement.

Quand vous serez parvenu à sortir de ce piège, vous pouvez passer à l'exercice suivant. Cet exercice est un peu plus difficile que les autres, et vous pouvez tout à fait le laisser momentanément de côté pour y revenir plus tard si vous n'y parvenez pas pour l'instant



Kill the Boss!