

Géométrie en folie avec



lynxcoding.org

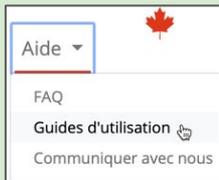
Avec un financement du





Lis le guide **Premiers pas** avant d'utiliser ces fiches... ou commence avec les fiches si tu préfères.

Le guide **Premiers pas** se trouve dans le menu **Aide** sur la page lynxcoding.org



Utilise ces fiches pour explorer des concepts de géométrie et apprendre les bases du codage avec Lynx!

Tu apprendras :

- Théorème du *tour complet de la tortue*
- Polygones
- Superprocédure et sous-procédure
- Hasard
- Variables
- Instructions conditionnelles



- Exécute ces fiches dans l'ordre, chaque nouvelle fiche fait suite à la précédente.
- N'oublie pas de partager tes idées, tes difficultés et tes défis avec tes amis.
- Mets tes amis au défi lorsque tu as de bonnes idées à imiter.
- Affiche tes projets dans ta classe (réelle ou en ligne)!
- **IMPORTANT: Donne un nom à ton projet et enregistre-le souvent!**

Partager

Enregistrer

Menu Objets

Menu Fichiers

Procédures

Clipart

Mes projets

Menu Réglages

Guide

Aide

Mon projet - page1

Procédures

```

1
2 ; Ceci est un exemple de procédure. Tape la
   commande DessineCarré dans le centre de comma
   (l'espace en-dessous de la zone de travail)
3
4 pour DessineCarré
5   baissecrayon
6   répète 4 [
7     avance 100
8     droite 90
9   ]
10 fin
11

```

Panneau de procédures

Panneau des cliparts

HELP!

Nomme ton projet

Tortue

Zone de travail (page)

Centre de commande



bc (baissecrayon)
lc (lèvecrayon)
gc (gommecrayon)

av (avance) **re** (recule)
dr (droite) **ga** (gauche)

attends

Ex:

av 100 attends 2 dr
50

vg (videgraphiques et
ramène la tortue au centre
de la page)



Tape ceci dans le centre de commande :

bc Appuie sur **Entrée**
av 100 Appuie sur **Entrée**

bc abaisse le crayon de la tortue. **Av 100** est une instruction qui dit à la tortue d'avancer de 100 pas (pixels). Le nombre **100** est une donnée.

Essaie aussi ceci :

dr 60 (tourne à droite de 60 degrés)
re 150 (recule de 150 pas)
ga 145 (tourne à gauche de 145 degrés)

Tape plusieurs commandes sur une seule ligne et appuie sur **Entrée**. Revient sur la ligne d'instructions (flèche vers le haut ou clic de souris) et appuie sur **Entrée** de nouveau.

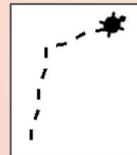


Essaie de placer des commandes **lc** et **bc** ici et là dans tes instructions.

lc (lèvecrayon)
bc (baissecrayon)

Peux-tu:

- Dessiner un trait pointillé?
- Écrire ton nom ou tes initiales?
- Déterminer la largeur ou la hauteur de l'espace de travail en pas de tortue?
- **Amuse toi!** Essaie de grandes et de petites valeurs. Essaie des valeurs inférieures à un ou des valeurs négatives!





La commande **répète** peut te sauver beaucoup de temps!

Les **crochets []** servent à contenir une liste - dans le cas de **répète**, il s'agit d'une liste d'instructions à répéter.

Les **crochets** sont près de la touche **Entrée**, ou **Option 8 et 9**, selon ton clavier.

Note: tu peux utiliser les primitives Lynx avec ou sans caractère accentué.



Tape ceci dans le centre de commande et appuie sur Entrée :

bc

répète 10 [av 100 re 90 dr 3]

Cela signifie « fais ceci 10 fois : avance de 100 pas, recule de 90 pas et tourne à droite de 3 degrés ».

Essaie ceci et utilise **vg** pour vider les graphiques quand tu veux.

répète 20 [av 100 dr 165]

répète 8 [av 70 re 60 dr 45]

répète 10 [av 100 dr 140 re 100 dr 45]

répète 6 [av 80 rt 60 re 80 ga 120 attends 2]

répète 20 [av 80 dr 18 attends 2 re 80 av 10 attends 2]

Regarde! Un répète qui se répète!

répète 10 [répète 15 [av 4 dr 15] dr 120]

répète 9 [répète 10 [av 4 dr 20] dr 120]



Ajoute une instruction **attends** pour ralentir la tortue au besoin.

Essaie de changer une donnée à la fois dans ces exemples.

Essaie d'ajouter des instructions **lc** et **bc** à certains endroits dans ces exemples.

Exécute la même instruction plusieurs fois. Super, non? Peux-tu créer une instruction **répète** qui fait exactement ceci pour toi?



Utilise le panneau de procédures pour créer des procédures.

Assure-toi de toujours terminer une procédure avec le mot **fin** seul sur sa ligne.

Si tu oublies, les autres procédures ne fonctionnent pas et tu verras le message

Je ne sais pas comment ceci-ou-cela



Une procédure est un groupe d'instructions auquel tu donnes un nom. Le nom devient un nouveau mot du vocabulaire de Lynx. Ce nouveau mot fonctionne seulement dans le projet sur lequel tu travailles présentement.

Une procédure comporte TROIS parties :

pour trait *La ligne titre : **pour** suivi d'une espace et du nom de la procédure.*

av 80 dr 90 *Le corps : des instructions pour les tortues ou d'autres objets.*

fin *Dernière ligne : **fin**, toujours seul sur sa ligne.*



Clique dans le panneau de procédures.

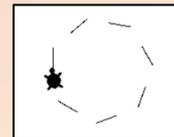
Tape quelque chose comme :

pour pointe
av 80 dr 60 re 80 ga 120
fin

Tape **pointe** dans le centre de commande.

Change les données (les valeurs) dans le procédure et essaie-la de nouveau.

Peux-tu faire une procédure qui dessine ceci??





Tour complet de la tortue

Un tour complet, c'est lorsque la tortue termine sa course avec exactement le même cap qu'à l'origine.

Pour faire un carré, la tortue tourne 4 fois de 90 degrés.

$$4 \times 90 = 360$$

360 degrés, c'est un tour complet!



pour carré
répète 4 [av 100 dr 90 attends 2]
fin

Plus grand ou plus petit?

Quelle valeur changerais-tu pour faire un carré plus grand ou plus petit?

Oui! La valeur de la commande **av**.

Quelle commande changerais-tu pour dessiner un carré « de l'autre côté »?

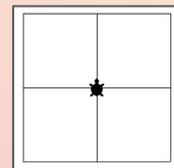
Oui! Change **dr** pour **ga**, ou change **av** pour **re**.

Note : Si tu utilises un caractère accentué dans le nom d'une procédure, tu dois toujours l'utiliser avec le caractère accentué.



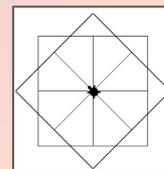
Dessine le plus petit carré possible. Ou le plus grand possible

Peux-tu recréer ce motif?



Indice: dessine un carré, tape une instruction dans le centre de commande pour faire pivoter la tortue, dessine un autre carré, et ainsi de suite.

Ou celui-ci?



Essaie de créer d'autres motifs intéressants!



fixecouleur
fixefond
fixetaillecrayon

fixecouleur 'rouge'
rend la tortue rouge, et elle dessinera en rouge.

fixefond 49 rend l'arrière-plan de la page olive. Pas besoin d'apostrophe (')

Il y a 140 couleurs, de 0 à 139

fixetaillecrayon 10 pour dessiner des traits plus larges.

La plus petite taille est 1, la plus grande est 30.



```
pour carrébleu
fixecouleur 'bleu'
fixetaillecrayon 8
répète 4 [av 100 dr 90 attends 2]
;fixecouleur change la couleur de la
tortue et de son crayon
fin
```

```
pour carrédroite
fixecouleur 'violettes'
fixetaillecrayon 5
répète 4 [av 100 dr 90 attends 2]
fin
```

```
pour carrégauche
fixecouleur 'orange'
fixetaillecrayon 5
répète 4 [av 100 ga 90 attends 2]
fin
```



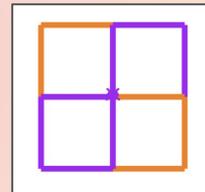
Modifie la procédure **pointe** en ajoutant des commandes **fixecouleur** et **fixetaillecrayon**.

```
pour pointe
av 80 dr 60 re 80 ga 120
fin
```

Peux-tu recréer ce motif?

Peux-tu changer la couleur de l'arrière-plan à l'aide de **fixefond**?

Essaie tout ce qui te passe par l'esprit!





Tour complet de la tortue (encore!)

Un tour complet, c'est lorsque la tortue termine sa course avec exactement le même cap qu'à l'origine. **Elle a fait 360 degrés.**

Un triangle **équilatéral** possède trois côtés de même longueur, et trois angles égaux.



À trois on y va!

Tu connais le théorème du tour complet de la tortue. Le nombre de fois que tu répètes (???) multiplié par l'angle que tu utilises (???) doit donner 360 :

répète ??? [av 100 dr ???]

Regarde cette procédure. Quelles valeurs utiliseras-tu pour créer un triangle équilatéral? Vas-y, remplace les ??? par tes valeurs!

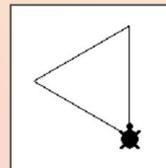
pour triangle

répète ??? [av 100 dr ???]

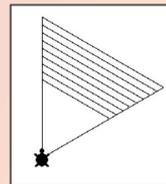
fin



Peux-tu faire un triangle équilatéral qui va vers la gauche? Un un triangle équilatéral plus grand ou plus petit?



Peux-tu recréer un motif semblable à celui-ci :



Invente un nouveau motif à ton goût.

Pense à utiliser **lc**, **bc**, **fixecouleur**, **fixetaillecrayon**



Le mot **polygone** vient de deux racines grecques.

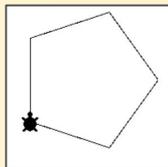
Poly signifie plusieurs et **gone** signifie angle.

Le mot **polygone** signifie **plusieurs angles**.

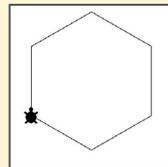


Alors dessinons des polygones!

pour pentagone
répète 5 [av 100 dr ???]
fin



pour hexagone
répète ??? [av 100 dr ???]
fin



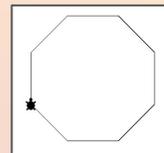
Quelles sont les valeurs d'angle (**dr ???**) pour ces polygones?

N'oublie pas le théorème du tour complet de la tortue!



Dessine un octogone.

Et un décagone.



Crée un polygone avec le plus grand nombre de côtés possible! Tu devras probablement réduire la donnée de **av** pour que le polygone ne déborde pas de la page.

Crée une procédure **polygone** et laisse l'ordi calculer l'angle pour toi!



N'oublie pas le théorème du tour complet de tortue!

En travaillant sur les polygones, tu as peut-être découvert comment faire un cercle...

N'oublie pas **fixecouleur** et **fixetaillecrayon**.



Regarde cette procédure **cercle** - quelles valeurs devrais-tu utiliser pour créer un cercle? Vas-y, essaie!

pour cercle
; n'oublie pas le tour complet
répète ??? [av ??? dr ???]
fin

Est-ce qu'un cercle est une sorte de polygone?

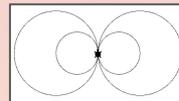
Tu devras probablement utiliser une petite valeur pour **av**, pour éviter que le cercle déborde de la page.

Lorsque tu as découvert la recette magique, tu peux vraiment t'amuser!



Peux-tu faire :

- Un très petit cercle.
- Un très grand cercle.
- Un cercle qui se trace vers la gauche!
- Des yeux de bande dessinée!



- Un bonhomme de neige?

Crée d'autres dessins faits de cercles!
Partage-les avec tes amis!



Même cercle,
différentes recettes!

Examine bien les
exemples suivants.
Porte attention aux
décimales!

Lynx accepte le point
décimal et la virgule
décimale.



Crée et essaie ces procédures :

```
pour cercle1
  répète 360 [av 1 dr 1]
end
```

```
pour cercle2
  répète 180 [av 0,5 dr 2]
end
```

```
pour cercle3
  répète 720 [av 1 dr 0,5]
end
```

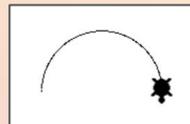
```
pour cercle4
  répète 720 [av 0,5 dr 0,5]
end
```

Pourquoi **cercle1** et **cercle4** donnent-ils le
même résultat? Demande à tes amis!



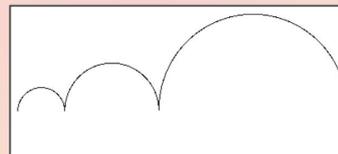
Dessine le même cercle de différentes façons!

Dessine un demi-cercle.



Dessine le même demi-cercle
façons.

Essaie de reproduire ce motif :



Indice:

Tu dois modifier le cap de la tortue entre
chaque demi-cercle... ou trouver une autre
façon...



Une **sous-procédure** est une procédure qui est utilisée dans une autre procédure.

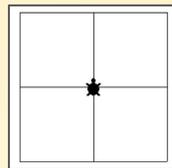
Une **superprocédure** est une procédure qui utilise une **sous-procédure**.

Ceci te permet de faire des programmes complexes composés de plus petits morceaux.



Tu te rappelles le défi de la fiche
Un simple carré?

C'est plus facile à réaliser à l'aide d'une **superprocédure** et d'une **sous-procédure**!



pour carré
bc

répète 4 [av 100 dr 90 attends 2]

lc

fin

pour 4carrés

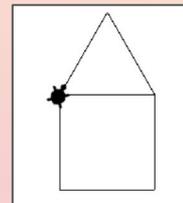
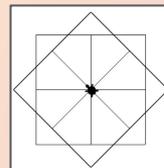
répète 4 [carré dr 90]

end

Carré est maintenant une **sous-procédure** à l'intérieur de la procédure **4carrés**.



Crée une **superprocédure** pour reproduire ces illustrations (ou d'autres illustrations).



Compare tes solutions à celles de tes amis!

Est-ce qu'il y a des solutions plus efficaces que d'autres? Pourquoi?



Carré est une sous-procédure utilisée dans les autres superprocédures.

Tour complet de la tortue

Lorsque la tortue termine sa course avec le même cap (la même orientation) qu'au début, elle a fait un tour complet : **360 degrés**.



pour carré
bc

répète 4 [av 100 dr 90 attends 1]

lc

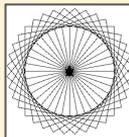
fin

to carréplus1

répète 36 [carré dr 10 attends 2]

fin

$$36 \times 10 = 360$$

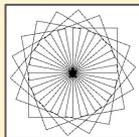


to carréplus2

répète 18 [carré dr 20 attends 2]

fin

$$18 \times 20 = 360$$



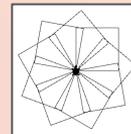
Peux-tu créer une procédure **carréplus3** pour créer une illustration comme celle-ci??

pour carréplus3

répète 9 [carré dr ??? attends 2]

fin

$$9 \times ??? = 360$$



Quel serait l'angle (**dr**) pour les cas suivants?

- Change la donnée de **répète** pour **12**.
- Change la donnée de **répète** pour **360**.
- Change la donnée de **répète** pour **180**.
- Expérimente avec d'autres données et d'autres angles!

Peux-tu demander à l'ordi de calculer l'angle pour toi?



Fixecouleur Hasard

fixecouleur hasard
140 choisit une couleur au hasard entre 0 et 139.

**Fixetaillecrayon 1 +
hasard 30** choisit une taille de crayon au hasard entre 1 et 30.

Note: Tu dois utiliser +1, car **fixetaillecrayon** ne peut pas utiliser 0 comme valeur.

Hasard est utile dans les jeux, les simulations, les math, etc.



Expérimente avec les primitives **fixecouleur**, **fixetaillecrayon** et **hasard**.

Reviens à la procédure **carréplus1** et ajoute un instruction de **hasard** pour changer la couleur. Une instruction peut déborder sur plusieurs lignes.

```
pour carré1
  répète 36 [carré dr 10 attends 2
    fixecouleur hasard 140]
end
```

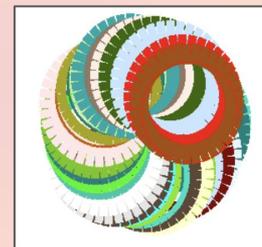
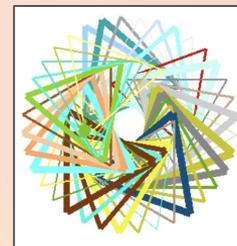
Tu peux aussi modifier la taille de crayon au hasard!

```
pour carré1
  répète 36 [carré dr 10 attends 2
    fixecouleur hasard 140 fixetaillecrayon
    1 + hasard 30]
fin
```



Utilise tes procédures de polygone pour créer des motifs colorés!

Partage tes productions artistiques avec tes amis!





Décompose ton programme!

Crée des sous-procédures faciles à comprendre et à corriger.

La **décomposition** consiste prendre un gros défi et à le briser en petits défis plus facile à relever.

C'est une habileté fondamentale de la pensée algorithmique.



```
pour carré
répète 4 [av 100 dr 90 attends 2]
fin
```

```
pour déplace
lc
dr hasard 360
av hasard 300
bc
fin
```



```
pour dansecarrée
répète 50 [déplace fixecouleur hasard
140 fixetaillecrayon 1 + hasard 30
carré]
fin
```

Déplace est une sous-procédure de la procédure **dansecarrée**.



Modifie la procédure **déplace**.

- Change la donnée de **dr hasard**.
- Change la donnée de **av hasard**.

Fais des changements semblables à la procédure **dansecarrée**.

Conseil : Fais un seul changement à la fois pour bien voir son effet.



Remplis = remplir une surface de couleur!

fixetaillecrayon 4
crée un carré bien
« solide » pour que la
couleur ne coule pas par
les coins!

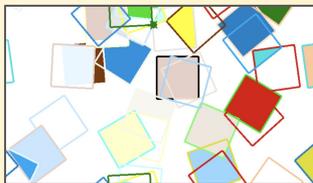
Toujours exécute les
instructions entre
crochets jusqu'à ce qu'on
l'arrête.

Clique sur  pour
arrêter l'exécution.



```
pour square
  répète 4 [av 100 dr 90 attends 2]
fin
```

```
pour peinsdedans
  ;va à l'intérieur du carré
  lc dr 45 av 20 bc
  fixe couleur hasard 140
  remplis
fin
```



```
pour déplace
  lc dr hasard 360
  av hasard 300 bc
fin
```

```
pour motif
  fixetaillecrayon 4
  toujours [carré peinsdedans déplace]
fin
```

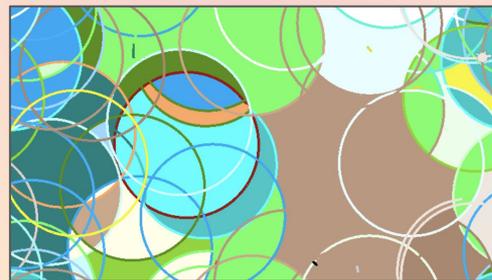


Dans la procédure **déplace**, change :

- La donnée de **dr hasard**
- La donnée de **av hasard**

Fais des changements semblables à la
procédure **peinsdedans**.

Essaie ceci avec d'autres types de
polygones!





En mathématiques, une variable est une lettre qui change de valeur. Avec Lynx, tu peux utiliser des mots pour représenter les variables.

Dans cet exemple, `:taille` est une variable locale.

Tu peux utiliser le mot que tu veux...
`:taille` par exemple. N'oublie pas les deux points (:)

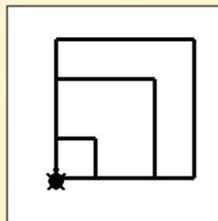


```
pour carré :taille
bc
  répète 4 [av :taille dr 90]
end
```

La variable `:taille` doit se trouver sur la ligne titre et dans le corps de la procédure.

Essaie ceci dans le centre de commande :

```
carré 40
carré 100
carré 140
```



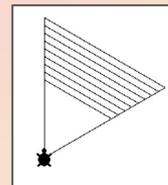
Crée une procédure `cercle` qui utilise une variable locale.

Crée d'autres polygones qui utilisent une variable locale.

Tu te souviens de ce défi sur la fiche *Triangles?*

C'est plus facile avec une variable!

Essaie!





Voici une procédure qui utilise **deux** variables :

:long pour la longueur d'un trait, et **:angle** pour l'angle.

Note : La procédure **spirale** appelle la procédure **spirale**. Elle s'appelle elle-même comme sous-procédure!

Mais à chaque fois qu'elle s'appelle, elle ajoute **2** à **:long**

```
spirale :long + 2 angle
```

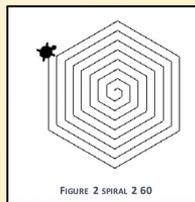
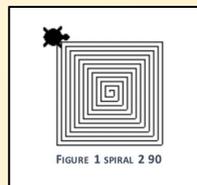


```
pour spirale :long :angle
  fixetaillecrayon 1 ;pour rétablir
  la taille normale de crayon
  si :long = 100 [stop]
  ;Instruction conditionnelle qui
  dit que si :long = 100, la
  procédure doit arrêter.
  av :long
  dr :angle
  spirale :long + 2 :angle
  fin
```

Essaie : **spirale 2 90**

Essaie : **spirale 2 60**

Peux-tu expliquer la différence?



Jouons un peu avec les valeurs.

Change le **2** dans
spirale :long + 2 :angle
pour une autre valeur.

Reviens à :
spirale :long + 2 :angle
et essaie **spirale 3 60**

Oops, qu'est-ce qui se passe?

Indice : la condition d'arrêt est :
si :long = 100 [stop]

Hmmmm... est-ce que « **= 100** » cause un problème?

Y a-t-il un autre symbole mathématique qui permettrait d'éviter ce problème?



Attention :
Ici, nous utilisons « > »
au lieu de « = ».

```
si :long > 100 [stop]
```

Aussi, pour faire une
spirale plus « serrée »
la **:long** augmente
seulement de **0,5** à
chaque cycle.

```
spirale :long + 0,5 :angle
```



pour spirale :long :angle

```
si :long > 100 [stop]
```

;Instruction conditionnelle qui dit
que si la **:long > 100**, la procédure
doit arrêter.

```
av :long
```

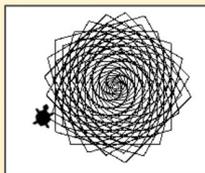
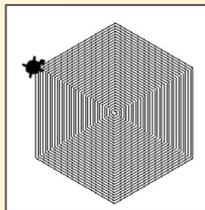
```
dr :angle
```

```
spirale :long + 0,5 :angle
```

```
fin
```

Essaie : **spirale 1 60**

Essaie : **spirale 2 75**



Peux-tu expliquer
la différence?



Change

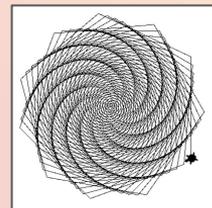
```
si :long > 100 [stop] pour :
```

```
si :long > 200 [stop]
```

Tape ceci dans le centre
de commande:

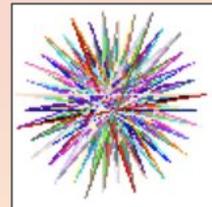
```
spirale 2 65
```

```
spirale 2 170
```



OK!

Ajoute une instruction
fixecouleur hasard 140
dans la procédure.



Vas-y, expérimente!
Et partage!



Qu'as-tu appris?

Beaucoup de choses en fait!

- `lc, bc, av, re, dr, ga`
- `vg, attends, répète, []`
- `fixecouleur, fixetaillecrayon, remplis`
- `fixefond, hasard`
- `toujours, stop`
- `pour, fin,`
procédures,
superprocédures,
sous-procédures
- Tour complet de la tortue
- Instruction
conditionnelle avec `si`



Une touche artistique

Bien qu'il reste beaucoup à apprendre, tu es maintenant prêt(e) à expérimenter avec ce que tu sais déjà.

Tu peux maintenant créer de petites oeuvres d'art!

Relis les fiches pour te donner des idées et allumer ta flamme d'artiste.

Tu peux faire une oeuvre statique (qui ne change pas), ou tu peux créer un oeuvre dynamique, comme une présentation, avec une instruction `attends` entre chaque création.



Tu pourrais créer une procédure `prépare` semblable à celle-ci, ainsi qu'une procédure `art`.

```
pour prépare
fixecouleur hasard 140
lc
dr hasard 360
av hasard 200
bc
fin
```

```
pour art
prépare
;écrit tes instructions ici
fin
```