

La validation de la commande Tortue[] dans Saisie crée un point nommé tortue1, de coordonnées (0,0) (et de cap, 0°) dans Graphique dont le visuel est .

et aussi vous devez noter l'apparition du bouton Exécuter ▷ (qui, lorsqu'il est cliqué, devient Pause ◻◻) dans le coin inférieur gauche de Graphique, si le bouton Exécuter ▷ est affiché, tous les ordres que vous allez saisir seront enregistrés en différé.

Une tortue est, bien entendu, un point animé, d'où la présence de ces boutons, la sollicitation du bouton Exécuter ▷ pourra être remplacée par la commande DémarrerAnimation[]

La fenêtre de dialogue classique des propriétés d'un point est accessible, mais la modification de beaucoup de ces propriétés est inopérante actuellement, ainsi par exemple, modifier la couleur du point ne changera pas la couleur de la tortue, simplement, lorsque celle-ci est sélectionnée, le tracé qu'elle aura construit apparaîtra dans la couleur choisie.

La seule propriété intéressante et opérationnelle à ce jour est l'association d'image au point, ce qui vous permet de pouvoir quand même différencier vos tortues si vous en utilisez plusieurs.

La commande SoitCoordonnées[<Objet>, <x>, <y>] est opérationnelle pour modifier la position de la tortue. Les fonctions x() et y() agissent sur la tortue, mais ATTENTION, elles retournent les coordonnées finales de la tortue que les ordres saisis aient été exécutés ou non.

Les commandes spécifiques à la tortue :

Translations :

TortueAvance et TortueReculer,

la tortue nommée va avancer ou reculer, en ligne droite, de la distance donnée

Rotations :

TortueADroite et TortueAGauche,

la tortue nommée va pivoter, sur place, vers la droite ou vers la gauche, de

l'angle donné.

Tracés :

TortueBC (BC pour BaisseCrayon),

la tortue nommée va laisser une trace de son déplacement, et

TortueLC (LC pour LèveCrayon),

la tortue nommée ne va pas laisser de trace de son déplacement.

Certaines commandes comme Exécute[], Répéter[] et LanceScriptActu|Clic[] seront souvent sollicitées.

Construire un carré :

Valider successivement chacune des commandes suivantes dans Saisie :

```
t=Tortue[]  
TortueAvance[t,1]  
TortueAGauche[t,90°]  
TortueAvance[t,1]  
TortueAGauche[t,90°]  
TortueAvance[t,1]  
TortueAGauche[t,90°]  
TortueAvance[t,1]
```

Bien sûr, comme annoncé dès le départ, rien ne se passe dans Graphique, après l'apparition de la tortue et du bouton Exécuter ►, simplement si Algèbre est ouverte, vous pouvez remarquer les variations des coordonnées du point t .

(rappel : ne pas se priver de la fonctionnalité "Refaire" du champ de saisie !!!)
et presser enfin le bouton Exécuter ►

Une tortue, nommée t , va construire un carré direct, de côté 1, à partir de l'origine.

Il est évident que la saisie de ces 8 lignes de commandes est fastidieuse, surtout lorsque l'on sait qu'elle peut être ramenée à une saisie de 2 lignes :

```
t=Tortue[]  
Répéter[4, TortueAvance[t, 1], TortueAGauche[t, 90° ]]
```

Tout de suite, une remarque, le passage d'arguments globaux fonctionne, ainsi, si est définie une variable numérique c :

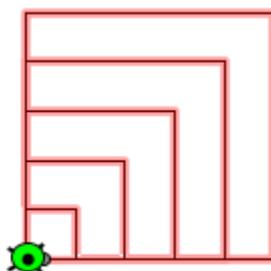
```
Répéter[4, TortueAvance[t, c], TortueAGauche[t, 90° ]]
```

La tortue t , va construire un carré direct, de côté c , à partir de l'origine, à reculons, si c est négatif. Mais le tracé ne sera pas impacté par une modification ultérieure de la valeur de c .

Et sans plus attendre, valider successivement chacune des commandes suivantes dans Saisie :

```
c=1  
n=5  
Répéter[n,Répéter[4, TortueAvance[t, c], TortueAGauche[t, 90° ]],SoitValeur[c,c+1]]
```

L'exécution vous tracera ces 5 classiques carrés emboîtés de côtés respectifs 1, 2, 3, 4 et 5.



"Procédures"

La philosophie de Logo était plutôt de définir des "procédures", réalisant un travail basique, les procédures étant ensuite sollicitées en groupe pour effectuer une tâche plus complexe.

Je vais utiliser les scripts GeoGebra attachés aux objets pour définir des procédures. J'ai choisi sciemment comme type d'objets les textes, ce qui me permet d'afficher facilement le listing,

Je lance l'outil Insérer un texte, je valide :

Pour Carré :c

```
Répéter[4, TortueAvance[t, c], TortueAGauche[t, 90° ]]
```

En ouvrant les propriétés de ce texte, je le renomme Carré, et je fais un copier/coller de l'instruction dans l'onglet Script/ParClic (ne pas oublier de presser le bouton "Ok")

L'exécution des 2 commandes suivantes :

c=2

t=Tortue[]

et un **clik sur le texte**

construit comme attendu un carré de côté 2

De même maintenant avec un autre texte, dans le même fichier, que je renomme CarrésEmboîtés :

Pour CarrésEmboîtés :n :c

```
Répéter[n,LanceScriptClic[Carré],SoitValeur[c,c+1]]
```

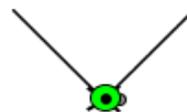
après avoir validé c=1 et n=5, un **clik sur ce nouveau texte** me redonne la construction des carrés emboîtés précédente.

Exemple page 55 de "Logo, des ailes pour l'esprit"

<http://tube.geogebra.org/material/show/id/878345>

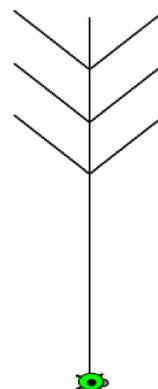
Pour Feuille

```
TortueAGauche[t,45°]  
TortueAvance[t,2]  
TortueRecule[t,2]  
TortueAGauche[t,90°]  
TortueAvance[t,2]  
TortueRecule[t,2]  
TortueADroite[t,135°]
```



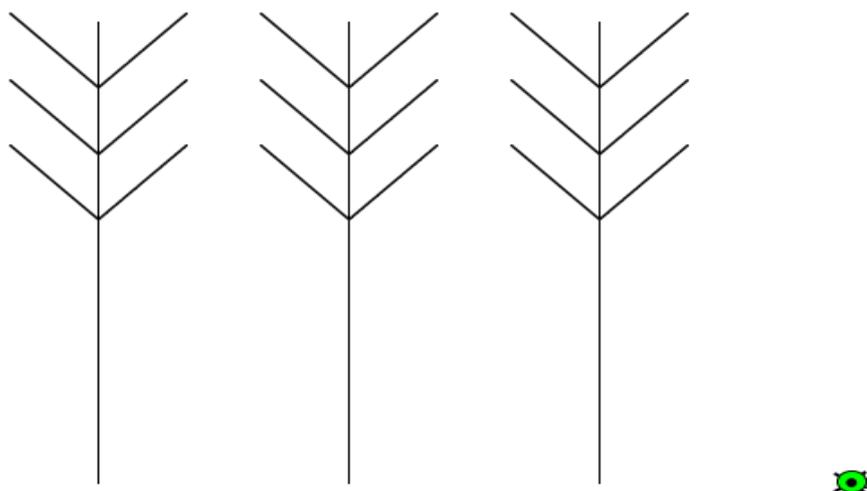
Pour Epi

```
TortueAGauche[t,90°]  
TortueAvance[t,5]  
Répéter[3, TortueADroite[t,90°],  
  LanceScriptClic[Feuille],  
  TortueAGauche[t,90°], TortueAvance[t,1.25]]  
TortueRecule[t,8.75]  
TortueADroite[t,90°]
```



Pour Champ

```
Répéter[3, LanceScriptClic[Epi], TortueLC[t], TortueAvance[t1,4], TortueBC[t]]
```



Avec un bouton de gestion ayant pour script :

```
Effacer[t]  
t=Tortue[]  
SoitCoordonnées[t1,-8,-1]  
DémarrerAnimation[]
```