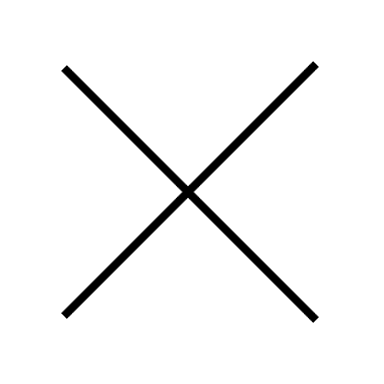
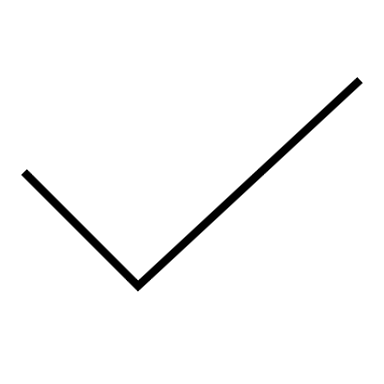
# Python

ctrl + c = copier

ctrl + v= collé

pour utiliser " " ou () ou ’ dans une variable il faut entourer la variable 3 apostrophes ex :

*« aujourd’hui il fait beau »*

*’’’aujourd’hui il fait beau ’’’*

alors pour ne pas mettre les 3 apostrophes il faut mettre un slash avant la parenthèse , apostrophe, guillemet.

*« aujourd/’hui il fait beau »* Ou«/ aujourd’hui il fait beau / »

## Calcul

pour la division il faut utiliser «/»

pour la somme «+»

pour la multiplication «\*»

pour la soustraction «-»

et pour l’égale «=»

## =

avec l’égale ou peux faire devenir des mots en autre mot :

*gâteau = bon print ( gâteau) >>>bon*

## #

### Les « # » servent à faire des commentaires qui ne se voient pas et qui ne changent rien

## %

Le % remplace le « \* » Ex :

*monscore = 100 message = « tu as obtenu %s pts » print ( message % monscore )*

*>>> tu as obtenu 100 pts*

quand il y a « %s », ça veut dire qu’il y a quelque chose qui va le remplacer, et quand il y a  « % », c’est un texte comme là avec monscore, c’est ce qu’il est égal au monscore qui va elle ici.

Ou sinon autre explication : dans le message, le %s est égal à monscore

## les différentes listes

### Il y a deux sortes de listes

### la première sorte de liste est comme une sorte de bloc et qui s’écrit qu’avec juste deux guillemets (un au début et un à la fin )

### la deuxième sorte liste où il y a plusieurs éléments distincts et chaque élément est entouré de crochets, guillemets ou apostrophes pour qu’il soit reconnaissable.

### Exemple d’une liste avec plusieurs éléments distincts :

### *Jour\_de\_la\_semaine = [’demain’, "c’est ",’samedi’, ’donc’, "c’est", ’le’,’ week-end’]*

### *Print =( Jour\_de\_la\_semaine[2])*

### *>>> ’samedi’*

### Le « c’est » est entourer de guillemets au lieu d’apostrophes, car il y a une apostrophe dans l’élément. Aussi, dans une liste, on commence a compté à partir de 0.

### Pour ne prendre que certain élément, il faut faire :

### *Jour\_de\_la\_semaine = [’demain’, "c’est ",’samedi’, ’donc’, "c’est", ’le’,’ week-end’]*

### *Print = (Jour\_de\_la\_semaine[2 :5])*

### *>>> ’samedi’, ’donc’, "c’est", ’le’*

### ou peut aussi mettre deux chaines ensembles :

### *liste\_1 =(‘a’, ‘b’ , ‘c’, ‘d’)*

### *liste\_2 = (‘1’, ‘2’, ‘3’, ‘4’)*

### *liste\_1&2= [1\_2]*

### *print (1\_2)*

### *>>> [[a,b,c,d],[1,2,3,4,]]*

### Dans une liste, on peut aussi ajouter des choses grâce à « append » :

### *liste\_1 =[‘a’, ‘b’ , ‘c’, ‘d’]*

### *liste\_1.append (‘e’)*

### *print (liste\_1)*

### *>>>[‘a’, ‘b’ , ‘c’, ‘d’,’e’]*

### Et c’est la même chose pour « del »

### *liste\_1 =**[‘a’, ‘b’ , ‘c’, ‘d’]*

### *del liste\_1 [2]*

### *print (liste\_1)*

### *>>>[‘a’, ‘b’ , ‘c’]*

### On peut aussi fusionner deux listes :

### *liste\_soleil = ['1','2','3']*

### *liste\_soleil2 = ['soleil']*

### *liste\_marrelle =liste\_soleil+liste\_soleil2*

### *print (liste\_marrelle)*

### ou les multiplier

### *liste\_chiffre= ['1','2','3']*

### *print (liste\_chiffre\*5)*

### grâce «dict» (dictionnaire) on peut retrouver des éléments (valeur)grâce à un autre élément (la clé) :

### *sport\_pref={'agathe':'boxe','melissa':'hand','maia':'volley','semiha':'basket','lubin':'escalade'}*

### *print (sport\_pref ['melissa'])*

### *>>>hand*

## Turtle

Pour faire avancer la tortue il faut marquer «forwad (la longueur que on veut)» puis pour tourner il faut marquer «left ou right (le degré qu’on veut)» :

0

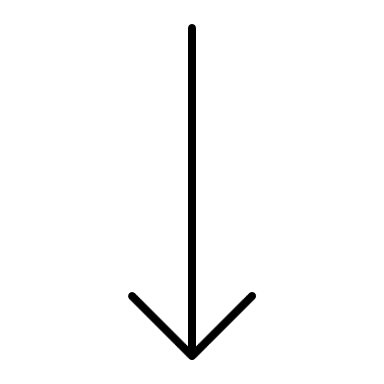
*turtle.forward(100) turtle.right(90)*

375

270

45

90



225

135

180

### Et pour faire un carré il faut faire cette manip. 4 fois :

### *turtle.forward(100)*

### *turtle.right(90)*

### *turtle.forward(100)*

### *turtle.right(90)*

### *turtle.forward(100)*

### *turtle.right(90)*

### *turtle.forward(100)*

### pour que la commande reste afficher il faut mettre « *turtle.exitonclick()»* à la fin

### pour qu’à la fin du dessin le dessin s’enlève et que la flèche revienne au point de départ il faut mettre ça *«turtle.reset()»*

### et si on met *«turtle.clear()»* ça enlève aussi le dessin mais ça laisse la flèche où elle est.

### Avec la commande *«turtle.backward(...)»*on peut faire un trait en arrière.

### Pour lever la flèche (pour mettre un espace) il faut mettre « *turtle.up() »* au début et « *turtle.down() » a la fin et au milieu turtle.forward(100)*

Pour remplir une forme il faut marquer au début *« turtle.begin\_fill() »* et à la fin *« turtule.end\_fill() »* et pour y mettre une couleur il faut mettre *« turtle.color() »*

*Pour faire un cercle il faut mettre juste « turtle.circle(50) »*