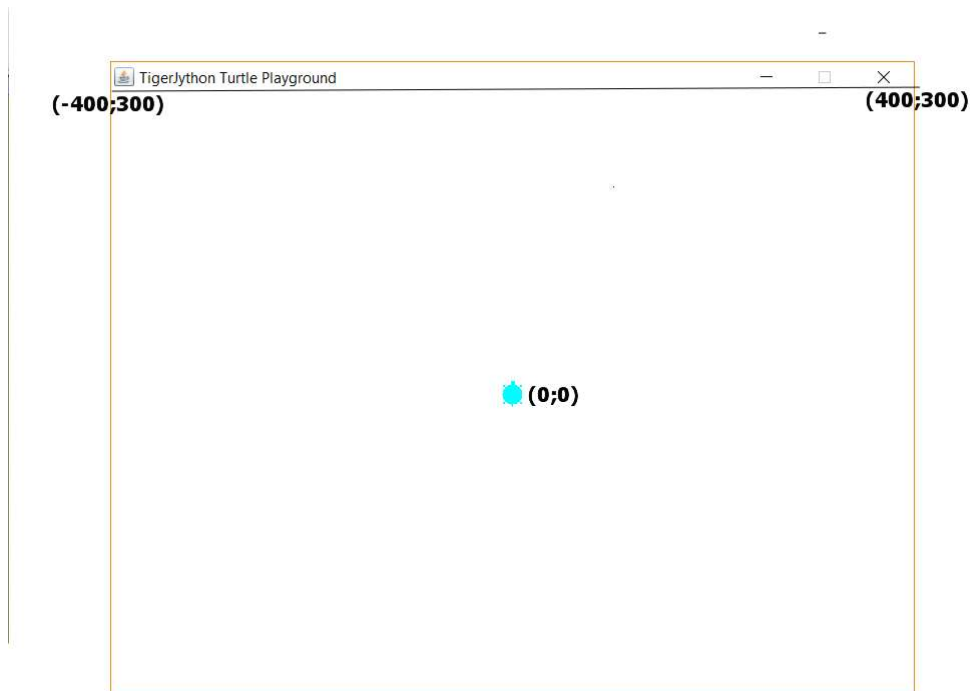


TURTLE GRAPHIC



La taille par défaut de la fenêtre graphique Turtle est 801x601.

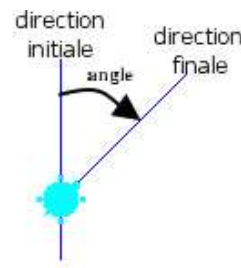
La tortue peut sortir de la fenêtre.

Le repère de la fenêtre est un repère classique avec le centre de la fenêtre comme origine du repère.

Par défaut,

- la couleur de la tortue est bleu clair,
- sa position est l'origine du repère de la fenêtre et
- elle est dirigée vers le nord.

Lecture de l'angle pour les commandes **left** et **right**.



Coordonnées de référence:

- point en haut à gauche: (-400;300),
- point en haut à droite (400;300),
- point en bas à gauche (-400;-300),
- point en bas à droite (400;-300),
- point central (0;0).

Résumé des principales commandes Python spécifiquement liées à la librairie Turtle Graphic

Les commandes s'utilisent de la manière suivante:

nom de la tortue.*nom de la commande*(*paramètres*)

| | |
|---|--|
| <code>makeTurtle()</code> | Crée une tortue par défaut sans nom |
| <code>joe=makeTurtle()</code> | Crée une tortue par défaut de nom <i>joe</i> |
| | |
| <code>makeTurtle(color)</code> | crée une tortue par défaut avec la couleur <i>color</i> |
| <code>makeTurtle("sprites/turtle.gif")</code> | crée une tortue par défaut avec l'image <i>turtle.gif</i> |
| <code>luka = Turtle()</code> | création de la tortue nommée <i>luka</i> dans sa propre fenêtre |
| <code>luka = Turtle(joe)</code> | création de la tortue nommée <i>luka</i> dans la fenêtre de la tortue <i>joe</i> |
| | |
| <code>tf = TurtleFrame ()</code> | crée une fenêtre <i>tf</i> pouvant contenir plusieurs tortues |
| <code>joe = Turtle(tf)</code> | crée un objet tortue <i>joe</i> dans la TurtleFrame <i>tf</i> |
| <code>joe.clone ()</code> | crée un clone de <i>joe</i> (même couleur, position, même direction) |

| | |
|---|--|
| <i>joe.forward</i> (100) | <i>joe</i> avance de 100 pixels. |
| <i>joe.back</i> (100) | <i>joe</i> recule de 100 pixels. |
| <i>joe.right</i> (90) | <i>joe</i> tourne à droite de 90 degrés. |
| <i>joe.left</i> (90) | <i>joe</i> tourne à gauche de 90 degrés. |
| <i>rightArc</i> (<i>radius</i> , <i>angle</i>) | Trace un arc de cercle de rayon <i>radius</i> et d'angle <i>angle</i> vers la droite |
| <i>rightCircle</i> (<i>radius</i>) | Trace un cercle vers la droite |
| <i>leftArc</i> (<i>radius</i> , <i>angle</i>) | Trace un arc de cercle de rayon <i>radius</i> et d'angle <i>angle</i> vers la gauche |
| <i>leftCircle</i> (<i>radius</i>) | Trace un cercle vers la gauche |
| <i>joe.speed</i> (10) | régler la vitesse de déplacement de la tortue <i>joe</i> à 10 pixels par seconde |
| <i>joe.getSpeed</i> () | Donne la vitesse de la tortue <i>joe</i> . |
| <i>joe.setPos</i> (0, 160) | La tortue <i>joe</i> se place à la position (0;160). |
| <i>joe.getPos</i> () | obtenir la position de la tortue <i>joe</i> |
| <i>joe.setX</i> (40) | régler la position X de la tortue <i>joe</i> |
| <i>joe.setY</i> (40) | régler la position Y de la tortue <i>joe</i> |
| <i>joe.getX</i> () | obtenir la position X de la tortue <i>joe</i> |
| <i>joe.getY</i> () | obtenir la position Y de la tortue <i>joe</i> |
| <i>joe.home</i> () | <i>joe</i> retourne au centre de l'aire de jeu. |
| <i>joe.distance</i> (x, y) | obtenir la distance entre le point (x;y) et la tortue <i>joe</i> |
| <i>joe.setColor</i> ('red') | régler la couleur de la tortue <i>joe</i> |
| <i>joe.setPenColor</i> ('red') | régler la couleur du trait dessiné par la tortue <i>joe</i> |
| <i>joe.penErase</i> () | efface |
| <i>joe.setLineWidth</i> (12) | Le pinceau de <i>joe</i> a une largeur de 12 pixels. |
| <i>joe.setFillColor</i> ('red') | règle la couleur de remplissage |
| <i>joe.fill</i> (1,1) | verser la couleur de remplissage au point (1;1) |
| <i>joe.label</i> (text) | écrire le texte de <i>joe</i> |
| <i>joe.setFont</i> (font) | régler la police de caractère de <i>joe</i> |
| <i>joe.setFontSize</i> (size) | régler la taille de la police de <i>joe</i> |
| <i>joe.penUp</i> () | soulever le crayon de <i>joe</i> |
| <i>joe.penDown</i> () | déposer le crayon de <i>joe</i> |
| <i>joe.hideTurtle</i> () | masquer la tortue <i>joe</i> |
| <i>joe.showTurtle</i> () | afficher la tortue <i>joe</i> |
| <i>clear</i> () | Efface la fenêtre |
| <i>joe.addStatusBar</i> (40) | ajout d'une zone de texte en bas de la fenêtre de 40 pixels de hauteur |
| <i>joe.setStatusText</i> ("Ceci est unmessage") | Affichage dans la zone de texte |
| <i>joe.setStatusText</i> (variable) <i>joe.setStatusText</i> (str (variable)) | Affichage de la variable dans la zone de texte si variable est un texte si variable n'est pas un texte |
| <i>joe.setStatusText</i> ("la variable vaut " + variable) <i>joe.setStatusText</i> ("la variable vaut " + str (variable)) | Affichage d'un message qui contient une variable texte dans la zone de texte Affichage d'un message qui contient une variable non texte dans la zone de texte |
| <i>joe.getPixelColor</i> () | Récupère la couleur du pixel où se trouve <i>joe</i> |
| <i>joe.getPixelColor</i> ().equals(Color.red) | Compare la couleur du fond avec la couleur rouge |
| <i>joe.getPixelColorStr</i> ()=='blue' | Compare la couleur du fond avec la couleur bleue |
| <i>pg= joe.getPlayground</i> () | Déclaration d'un fond pour la fenêtre de <i>joe</i> |
| <i>pg.drawImage</i> ("etoile.png", -150, 150) | On pose l'image etoile.png sur le fond le centre de l'image est en (-150;150). |

Module import: **from gturtle import ***

| | |
|----------------------------------|--|
| makeTurtle() | Crée une tortue (globale) dans une nouvelle fenêtre et définit toutes les commandes globales |
| makeTurtle(color) | Idem, mais en spécifiant la couleur de la tortue à créer |
| makeTurtle("sprites/turtle.gif") | Idem, mais en créant une tortue représentées par l'image de sprite spécifiée |
| t = Turtle() | Créer un objet tortue t |
| tf = TurtleFrame() | Crée une fenêtre de graphique (TurtleFrame) pouvant contenir plusieurs tortues |
| t = Turtle(tf) | Crée une tortue t qui se trouve dans le TurtleFrame tf |
| clone() | Crée un clone de la tortue globale possédant les mêmes propriétés (même couleur, position et orientation) |
| isDisposed() | Retourne True si la fenêtre est fermée |
| putSleep() | Met l'exécution du programme en pause jusqu'à l'appel de la fonction wakeUp() |
| wakeUp() | Reprend l'exécution d'un programme mis en pause avec putSleep() |
| enableRepaint(False) | Déclenche le rendu automatique de l'écran |
| repaint() | Restitue l'écran manuellement (après déclenchement du rendu automatique) |
| savePlayground() | Copie le playground dans une memoire d'image interne (restituer avec clear()) |
| savePlayground(fileName, format) | Copie le playground dans un fichier image (format: "png" ou "gif"). Retourne False dans le cas échéant |
| setPlaygroundSize(width, height) | Sélectionne la grandeur de la fenêtre graphique indépendant des options dans TigerJython (doit être appelé avant makeTurtle()) |
| setFramePosition(x, y) | Positionne le coin en haut à gauche de la fenêtre graphique à la position x, y de l'écran |
| setFramePositionCenter() | Positionne la fenêtre graphique au centre de l'écran |

Mouvements

| | |
|--|--|
| <code>back(distance), bk(distance)</code> | Déplace la tortue en arrière sur la distance indiquée (coordonnées tortue) |
| <code>forward(distance), fd(distance)</code> | Déplace la tortue en avant sur la distance indiquée (coordonnées tortue) |
| <code>hideTurtle(), ht()</code> | Cache la tortue, ce qui a pour effet d'accélérer le traçage du dessin |
| <code>home()</code> | Remplace la tortue dans sa position d'origine au centre de la fenêtre et orientée vers le haut |
| <code>left(angle), lt(angle)</code> | Tourne la tortue de angle degrés vers la gauche |
| <code>penDown(), pd()</code> | Active le crayon (la tortue trace son chemin) |
| <code>penErase(), pe()</code> | Assigne au crayon la même couleur que le fond d'écran (trace invisible) |
| <code>leftArc(radius, angle)</code> | Déplace la tortue sur un arc de cercle orienté vers la gauche, d'angle angle (en degrés) et de rayon radius |
| <code>leftCircle(radius)</code> | Déplace la tortue sur un cercle de rayon radius en partant vers la gauche. |
| <code>penUp(), pu()</code> | Désactive le crayon (la trace de la tortue devient invisible) |
| <code>penWidth(width)</code> | Règle la largeur de la trace (en pixels) |
| <code>right(angle), rt(angle)</code> | Tourne la tortue de angle degrés vers la droite |
| <code>rightArc(radius, angle)</code> | Déplace la tortue sur un arc de cercle orienté vers la droite, d'angle angle (en degrés) et de rayon radius |
| <code>rightCircle(radius)</code> | Déplace la tortue sur un cercle de rayon radius en partant vers la droite. |
| <code>setCustomCursor(cursorImage)</code> | Règle le fichier image utilisé en guise de curseur de la souris |
| <code>setCustomCursor(cursorImage, Point(x, y))</code> | Idem, en indiquant les coordonnées relatives de l'image par rapport au point d'action de la souris. |
| <code>setLineWidth(width)</code> | Règle la largeur du crayon (en pixels) |
| <code>showTurtle(), st()</code> | Affiche la tortue globale |
| <code>speed(speed)</code> | Règle la vitesse du mouvement de la tortue |
| <code>delay(time)</code> | Arrête l'exécution du programme durant l'intervalle de temps time (en millisecondes) |
| <code>wrap()</code> | Les positions des tortues se trouvant en dehors de l'écran sont envoyées à l'intérieur de la fenêtre par une symétrie sur un tore. Une tortue qui dépasse à gauche réapparaît donc à droite et une tortue qui dépasse vers le haut réapparaît en bas de l'écran. |
| <code>clip()</code> | Contraire de <code>wrap()</code> : les tortues qui sortent de la fenêtre sont invisibles |
| <code>getPlaygroundWidth()</code> | Retourne la largeur m du territoire de la tortue (coordonnées -m/2...m/2) |
| <code>getPlaygroundHeight()</code> | Retourne la hauteur m du territoire de la tortue (coordonnées -m/2...m/2) |

Localisation

| | |
|---|---|
| <code>direction(x, y)</code> | Retourne l'angle pour tourner dans la direction de la position (x, y) |
| <code>direction(coords)</code> | Idem, en spécifiant les coordonnées coords sous forme de tuple (x,y), de liste [x,y], ou de nombre complexe $x + yj$ |
| <code>direction(turtle)</code> | Retourne l'angle pour tourner dans la direction d'une autre tortue |
| <code>distance(x, y)</code> | Retourne la distance séparant la position de la tortue et le point de coordonnées (x, y) |
| <code>distance(coords)</code> | Idem, en spécifiant les coordonnées coords sous forme de tuple (x,y), de liste [x,y], ou de nombre complexe $x + yj$ |
| <code>distance(turtle)</code> | Retourne la distance séparant la position de la tortue de la position d'une autre tortue |
| <code>getPos()</code> | Retourne la position courante de la tortue sous forme de liste |
| <code>getX()</code> | Retourne la coordonnée x de la tortue |
| <code>getY()</code> | Retourne la coordonnée y de la tortue |
| <code>heading()</code> | Retourne l'orientation actuelle de la tortue en degrés. L'orientation vers le nord correspond à un angle de 0°. Les angles sont orientés dans le sens des aiguilles de la montre. |
| <code>heading(degrees)</code> | Règle l'orientation de la tortue en degrés. L'orientation vers le nord correspond à un angle de 0°. Les angles sont orientés dans le sens des aiguilles de la montre. |
| <code>moveTo(x, y)</code> | Déplace la tortue à la position (x,y) en dessinant la trace |
| <code>moveTo(coords)</code> | Idem, en spécifiant les coordonnées coords sous forme de tuple (x,y), de liste [x,y], ou de nombre complexe $x + yj$ |
| <code>setHeading(degrees), setH(degrees)</code> | Règle l'orientation de la tortue (en degrés, 0° correspond au Nord, angles orientés dans le sens des aiguilles de la montre) |
| <code>setRandomHeading()</code> | Règle l'orientation de manière aléatoire avec un angle de vue compris entre 0° et 360° |
| <code>setPos(x, y)</code> | Positionne la tortue au point de coordonnées (x,y) sans dessiner la trace |
| <code>setPos(coords)</code> | Idem, en spécifiant les coordonnées coords sous forme de tuple (x,y), de liste [x,y], ou de nombre complexe $x + yj$ |
| <code>setX(x)</code> | Règle la coordonnée x de la tortue |
| <code>setY(y)</code> | Règle la coordonnée y de la tortue |
| <code>setRandomPos(width, height)</code> | Place la tortue sur une position aléatoire située à l'intérieur du rectangle centré à la position actuelle, de largeur width et de hauteur height |
| <code>setScreenPos(x, y)</code> | Positionne la tortue globale aux coordonnées d'écran x et y |
| <code>setScreenPos(Point(x, y))</code> | Idem en spécifiant les coordonnées par un objet Point(x,y) |
| <code>towards(x, y)</code> | Retourne l'angle d'orientation (en degrés) qu'il faut donner à la tortue pour qu'elle vise le point de coordonnées (x, y). |
| <code>towards(coords)</code> | Idem, en spécifiant les coordonnées coords sous forme de tuple (x,y), de liste [x,y], ou de nombre complexe $x + yj$ |
| <code>towards(turtle)</code> | Retourne l'angle d'orientation (en degrés) qu'il faut donner à la tortue pour qu'elle vise la tortue turtle |
| <code>toTurtlePos(x, y)</code> | Retourne une liste des coordonnées de la tortue globale au point x, y de l'écran |
| <code>toTurtlePos(Point(x, y))</code> | Idem en spécifiant les coordonnées par un objet Point(x,y) |
| <code>pushState()</code> | Sauvegarde l'état actuel de la tortue sur une pile (Last In, First Out) |
| <code>popState()</code> | Restaure le dernier état sauvegardé avec pushState(). Supprime ce dernier de la pile des états |
| <code>clearStates()</code> | Vide la pile de sauvegarde des états en supprimant tous les états sauvegardés |

Couleurs

| | |
|-------------------------------|---|
| askColor(title, defaultColor) | Ouvre une boîte de dialogue de sélection de couleur et retourne la couleur sélectionnée. Si l'utilisateur clique sur le bouton annuler, retourne None. |
| clear() | Efface toutes les traces et cache toutes les tortues en les laissant par contre à leur place. Si la memoire d'image créée par savePlayground() n'est pas vide, son contenu est affiché |
| clear(color) | Efface les traces et cache toutes les tortues puis colorie le fond d'écran avec la couleur color |
| clean() | Efface toutes les traces et toutes les tortues. Réinitialise la position de toutes les tortues au centre de l'écran. La memoire d'image créée par savePlayground() est vidée |
| clean(color) | Efface toutes les traces et toutes les tortues en coloriant le fond de la fenêtre avec la couleur color. Réinitialise la position de toutes les tortues au centre de l'écran. |
| clearScreen(), cs() | Efface toutes les traces et replace la tortue dans sa position d'origine |
| dot(diameter) | Dessiner un disque plein de diamètre diameter en le remplissant de la couleur assignée au crayon de la tortue. |
| openDot(diameter) | Dessiner un cercle vide de diamètre diameter en coloriant son périmètre de la couleur assignée au crayon de la tortue. |
| spray(density, spread, size) | Dessiner un nuage de points aléatoires à la position de la tortue. Paramètres: nombre de points, extension du nuage et largeur des points (sans size: size = 1) |
| fill() | Remplit la surface fermée qui entoure le point auquel se trouve la tortue avec la couleur de remplissage spécifiée au préalable avec setFillColor(color). |
| fill(x, y) | Remplit la surface fermée qui entoure le point de coordonnées (x,y) avec la couleur de remplissage spécifiée au préalable avec setFillColor(color) |
| fill(coords) | Idem, en spécifiant les coordonnées coords sous forme de tuple (x,y), de liste [x,y], ou de nombre complexe x + yj |
| fillToPoint() | Remplissage continu depuis la position actuelle de la tortue |
| fillToPoint(x, y) | Remplissage continu depuis le point de coordonnées (x, y) |
| fillToPoint(coords) | Idem, en spécifiant les coordonnées coords sous forme de tuple (x,y), de liste [x,y], ou de nombre complexe x + yj |
| fillToHorizontal(y) | Remplissage continu de la surface délimitée par la trace actuelle de la tortue et la droite horizontale formée par les points d'ordonnée y |
| fillToVertical(x) | Remplissage continu de la surface délimitée par la trace actuelle de la tortue et la droite verticale formée par les points d'abscisse x |
| fillOff() | Termine le mode remplissage |
| getColor() | Retourne la couleur de la tortue globale |
| getColorStr() | Retourne la couleur de la tortue globale sous la forme d'une chaîne de caractères X11 |
| getFillColor() | Retourne la couleur de remplissage actuelle |
| getFillColorStr() | Retourne la couleur de remplissage actuelle sous la forme d'une chaîne de caractères X11 |
| getPixelColor() | Retourne la couleur du pixel (trace ou fond) situé à la position actuelle de la tortue globale (None, si hors de la fenêtre graphique) |
| getPixelColorStr() | Idem, en retournant la couleur sous forme de chaîne X11 (chaîne vide, si le nom n'existe pas; None, si hors de la fenêtre graphique) |
| getRandomX11Color() | Retourne une couleur aléatoire sous forme de chaîne de caractères X11 |
| makeColor() | Retourne une couleur de référence correspondant à la valeur passée. Exemples de valeurs : ("7FFED4"), ("Aqua-Marine"), (0x7FFED4), (8388564), (0.5, 1.0, 0.83), (128, 255, 212), ("rainbow", n) avec n = 0..1 qui indique une couleur de l'arc-en-ciel : n=0.1 correspond au violet et n=0.9 au rouge |
| setColor(color) | Règle la couleur de la tortue à color. |
| setPenColor(color) | Règle la couleur du crayon utilisé par la tortue à color |
| setFillColor(color) | Règle la couleur de remplissage utilisée par la tortue à color |
| startPath() | Commence à tenir compte du mouvement de la tortue pour les opérations de remplissage subséquentes. |
| fillPath() | Contraire de startPath() : termine l'opération de remplissage à la position courante de la tortue et remplit la zone ainsi délimitée par la couleur de remplissage. |
| stampTurtle() | Crée une image de la tortue à la position courante. |
| stampTurtle(color) | Crée une image de la tortue à la position courante avec la couleur color |

Fonctions de rappel (callbacks)

| | |
|---|---|
| <p>makeTurtle(mouseNNN = onMouseNNN)</p> <p>makeTurtle(mousePressed = onMousePressed , mouseReleased= onMouseReleased))</p> | <p>Enregistre la fonction de rappel (callback) onMouseNNN(x,y) qui est appelée lorsqu'un événement de la souris survient. Les valeurs possibles pour NNN sont : Pressed, Released, Clicked, Dragged, Moved, Entered, Exited, SingleClicked, DoubleClicked, Hit : L'invocation se fait dans un thread (fil d'exécution) séparé, HitX: idem, mais les événements sont ignorés jusqu'à ce que le traitement de l'événement en cours soit terminé</p> <p>Utiliser la virgule pour séparer les différents appels</p> |
| <p>isLeftMouseButton()</p> <p>isRightMouseButton()</p> | <p>Retourne vrai si l'événement est généré par le bouton gauche, respectivement droit</p> |
| <p>makeTurtle(keyNNN = onKeyNNN)</p> | <p>Enregistre la fonction de rappel (callback) onKeyNNN(keyCode) qui est appelée lorsqu'une touche du clavier est enfoncée. Les valeurs possibles pour NNN sont : Pressed, Hit : invocation du gestionnaire d'événement dans un thread séparé et HitX : idem que Hit, mais les événements sont ignorés jusqu'à ce que la dernière exécution de la fonction de rappel ait retourné. keyCode est un entier unique qui identifie la touche pressée</p> |
| <p>getKeyModifiers()</p> | <p>Retourne un nombre entier représentant les touches spéciales du clavier enfoncées (Shift, Ctrl, etc...) et leurs combinaisons</p> |
| <p>makeTurtle(closeClicked = onCloseClicked)</p> | <p>Enregistre la fonction de rappel onCloseClicked() qui est appelée lorsque le bouton fermer de la barre de titre est actionné. La fenêtre doit ensuite être fermée manuellement en appelant dispose()</p> |
| <p>makeTurtle(turtleHit = onTurtleHit)</p> | <p>Enregistre la fonction de rappel onTurtleHit(x, y) qui est appelée lorsqu'on clique sur l'image de la tortue globale</p> |
| <p>t = Turtle(turtleHit = onTurtleHit)</p> | <p>Enregistre la fonction de rappel onTurtleHit(t, x, y) qui est appelée lorsqu'on clique sur l'image de l'objet tort</p> |

Texte, images et son

| | |
|--|--|
| <code>addStatusBar(20)</code> | Ajout une barre d'état de 20 pixels de hauteur |
| <code>beep()</code> | Émet un court son |
| <code>playTone(freq)</code> | Joue un son de fréquence <code>freq</code> (en Hz) et de durée 1000 ms. Fonction bloquante |
| <code>playTone(freq, block=False)</code> | Idem, mais en version non bloquante, ce qui permet de jouer plusieurs sons à la fois |
| <code>playTone(freq, duration)</code> | Joue le son de fréquence <code>freq`</code> et de durée <code>duration</code> (en ms) |
| <code>playTone([f1, f2, ...])</code> | Joue plusieurs sons consécutifs de fréquences <code>f1, f2, ...</code> et de durée 1000 ms |
| <code>playTone([(f1, d1), (f2, d2), ...])</code> | Joue plusieurs tons consécutifs indiqués par des tuples (fréquence, durée) |
| <code>playTone([("c",700), ("e",1500), ...])</code> | Joue plusieurs tons consécutifs en utilisant le nom de la note (A=la, H=si, C=do, D=ré, ..., et de durée indiquée en [ms]. La tessiture du générateur de son s'étend du <code>do1</code> , <code>do#1</code> , ... jusqu'au <code>si5</code> |
| <code>playTone([("c",700), ("e",1500), ...], instrument = "piano")</code> | Idem, mais en choisissant le type d'instrument à utiliser pour la restitution du son. Les instruments disponibles sont piano, guitar, harp, trumpet, xylophone, organ, violin, panflute, bird, seashore ... (voir la spécification MIDI) |
| <code>playTone([("c",700), ("e",1500), ...], instrument = "piano", volume=10)</code> | Idem, en rajoutant le choix du volume compris entre 0..100 |
| <code>label(param)</code> | Affiche le texte <code>text</code> à la position courante de la tortue (aligné à gauche) |
| <code>label(param1, param2, ...)</code> | Relie <code>str(params)</code> séparés par des espaces et affiche le texte |
| <code>label(param1, param2, ..., adjust = 'x')</code> | Idem, mais <code>x = 'l'</code> aligné à gauche (default), <code>'c'</code> centré, <code>'r'</code> à droite |
| <code>printerPlot(draw)</code> | Envoie vers l'imprimante les commandes de dessins contenues dans la fonction <code>draw()</code> dont la référence est passée en argument |
| <code>setFont(Font font)</code> | Définit la police utilisée par la fonction <code>label()</code> . <code>font</code> est un objet de la classe <code>Font</code> . Exemple: <code>Font("Courier New", Font.BOLD, 12)</code> . Default: <code>Font("SansSerif", Font.PLAIN, 24)</code> |
| <code>setFont(name)</code> | Définit le nom de la police en utilisant le style et la taille existants |
| <code>setFont(name, style)</code> | Définit le nom et le style de la police en utilisant la taille existante <code>style = Font.PLAIN, Font.BOLD, Font.ITALIC</code> |
| <code>setFont(name, style, size)</code> | Définit le nom, le style et la taille de la police |
| <code>setFontSize(size)</code> | Définit la taille de police en utilisant le nom et le style existants |
| <code>getTextHeight()</code> | Retourne la hauteur des caractères de la police actuellement en usage (en pixels) |
| <code>getTextAscent()</code> | Retourne la hauteur du jambage supérieur de la police actuellement en usage (en pixels). Pour plus de détails, consulter |
| <code>getTextDescent()</code> | Retourne la hauteur du jambage inférieur de la police actuellement en usage (en pixels). Cf. <code>getTextAscent()</code> . |
| <code>getTextWidth(text)</code> | Retourne la largeur en pixels du texte <code>text</code> avec la police actuellement en usage. |
| <code>setStatusText(text)</code> | Affiche le texte <code>text</code> dans la barre d'état. Tout texte présent dans la barre d'état au moment de l'appel est écrasé. |
| <code>setTitel(title)</code> | Affiche la chaîne de caractères <code>title</code> dans la barre de titre de la fenêtre |
| <code>img = getImage(path)</code> | Charge et retourne l'image par un fichier JAR, par un fichier du disque local ou par un serveur internet |
| <code>drawImage(img)</code> | Dessine l'image dans le fond d'écran avec le centre à la position actuelle de la tortue et orienté par l'angle actuelle de la tortue |
| <code>drawImage(path)</code> | Charge l'image par un fichier JAR, par un fichier du disque local ou par un serveur internet et dessine l'image dans le fond d'écran avec le centre à la position actuelle de la tortue et orienté par l'angle actuelle de la tortue |

Boîtes de dialogue

| | |
|-------------------------------|---|
| msgDlg(message) | Ouvre une boîte de dialogue modale avec le message message et un bouton OK |
| msgDlg(message, title = text) | Idem, en spécifiant le titre de la fenêtre |
| inputInt(prompt) | Ouvre une boîte de dialogue modale avec les boutons OK/Annuler. OK retourne le nombre entier entré. Un clic sur Annuler ou Fermer termine le programme. |
| inputInt(prompt, False) | Idem, sauf qu'un clic sur Annuler/Fermer n'interrompt pas le programme mais retourne la valeur None |
| inputFloat(prompt) | Ouvre une boîte de dialogue modale avec les boutons OK/Annuler. OK retourne le nombre à virgule flottante entré. Un clic sur Annuler ou Fermer termine le programme. |
| inputFloat(prompt, False) | Idem, sauf qu'un clic sur Annuler/Fermer n'interrompt pas le programme mais retourne la valeur None |
| inputString(prompt) | Ouvre une boîte de dialogue modale avec les boutons OK/Annuler. OK retourne la chaîne entrée. Un clic sur Annuler ou Fermer termine le programme. |
| inputString(prompt, False) | Idem, sauf qu'un clic sur Annuler/Fermer n'interrompt pas le programme mais retourne la valeur None |
| input(prompt) | Ouvre une boîte de dialogue modale avec les boutons OK/Annuler. OK retourne le nombre entier, le nombre flottant ou la chaîne de caractères saisie par l'utilisateur. Les boutons Annuler ou Fermer terminent le programme. |
| input(prompt, False) | Idem, sauf qu'un clic sur Annuler/Fermer n'interrompt pas le programme mais retourne la valeur None |
| askYesNo(prompt) | Ouvre une boîte de dialogue modale avec les boutons Oui/Non. Oui retourne True et Non retourne False. Les boutons Annuler ou Fermer terminent le programme. |
| askYesNo(prompt, False) | Idem, sauf qu'un clic sur Annuler/Fermer n'interrompt pas le programme mais retourne la valeur None |