Bedienungsanleitung

WebtigerTurtle ist die vierte Stufe der spiralartig aufgebauten Programmierumgebung XLogoOnline, die darauf ausgerichtet ist einen schülerfreundlichen Start in die Programmiersprache Python zu ermöglichen. Die Lernumgebung ist mitunter für den Unterricht mit dem Lehrmittel "Einfach Informatik: Programmieren 7-9" konzipiert.

Unterstützte Turtle Funktionen

Die folgenden Turtle spezifischen Befehle werden in WebTigerJython zusätzlich zum normalen Python Syntax unterstützt:

Befehl	**Abkürzung**	**Beschreibung**
from gturtle import *		Die Schildkrötenbefehle laden. Erst danach versteht der Computer die Befehle für die Turtle.
makeTurtle()		Die Turtle auf der Zeichenfläche erstellen.
forward(zahl)	fd(zahl)	Anzahl Schritte vorwärts gehen.
back(zahl)	bk(zahl)	Anzahl Schritte rückwärts gehen.
left(zahl)	lt(zahl)	Auf der Stelle um den Winkel nach links drehen.
right(zahl)	rt(zahl)	Auf der Stelle um den Winkel nach rechts drehen.
setPenColor("Farbe")	spc("farbe")	Die Farbe des Stiftes setzen. Die Farbe wird in Englisch angegeben. Die Turtle erhält eine Umrandung in der gewählten Stiftfarbe.
setPenWidth(zahl) setLineWidth(zahl)	spw(zahl)	Die Breite des Stiftes setzen. Die Breite wird in Pixeln angegeben.
print "Text" print zahl print("Text") print(zahl)		Den Text zwischen Anführungszeichen oder das Resultat des arithmetischen
delay(zeit)		Eine Anzahl Millisekunden warten, bevor das Programm weiterläuft.
speed(zahl)		Ändert die Geschwindigkeit der Turtle. Wird als Geschwindigkeit -1 angegeben, so läuft die Turtle, so schnell sie nur kann. Die Turtle läuft am langsamsten mit Geschwindigkeit 1.
hideTurtle()	ht()	Die Turtle unsichtbar machen, damit die Bilder schnell gezeichnet werden.
showTurtle()	st()	Die Turtle wieder sichtbar machen, damit du siehst, wie sie zeichnet.
repeat zahl: Körper		Einen Programmteil eine Anzahl Male ausführen. Der Körper (mit den Befehlen, die wiederholt werden) muss eingerückt sein.
penUp()	pu()	Den Stift hochheben und nicht mehr zeichnen.
penDown()	pd()	Den Stift senken und wieder zeichnen.
dot(zahl)		Zeichnet einen ausgefüllten Punkt mit Durchmesser d an der aktuellen Position der Turtle.
setHeading(zahl)	heading(zahl)	Setzt den Winkel nicht relativ zur Turtle sondern absolut, d.h. relativ zum Fenster. Ein Winkel von 0 bewirkt, dass die Turtle direkt nach oben zeigt, ein Winkel von 90 zeigt nach rechts usw.
heading()		Gibt den Winkel der Turtle absolut zur Zeichenfläche zurück.

setRandomHeading()		Setzt den Winkel der Turtle zufällig zwischen 0 und 360°.
x = input("Text")		Der Computer zeigt in einem Fenster die Frage an und speichert den vom Benutzer eingegebenen Wert anschliessend in der Variable ab. Es gibt diese Funktion in drei weiteren Varianten, bei welchen jeweils ein bestimmter Datentyp verlangt wird: inputInt("Frage"), inputFloat("Frage") und inputString("Frage").
sqrt(zahl)		Berechnet die Wurzel der Zahl.
isInteger(zahl)		Gibt einen Wahrheitswert zurück (true oder false) welcher angibt, ob die Zahl ein Integer ist oder nicht.
getPixelColorStr()		Gibt die Farbe des Pixels unter der Turtle zurück.
getKey()		Gibt die zuletzt gedrückte Taste als String zurück.
getKeyCode()		Gibt die zuletzt gedrückte Taste als ASCII Code zurück.
getPos()		Gibt die aktuelle Koordinaten Position der Turtle als Liste zurück. Die x-Koordinate kann mit getPos()[0] abgefragt werden und die y-Koordinate mit getPos()[1].
getX()		Gibt die x-Koordinaten der aktuellen Turtle Position zurück.
getY()		Gibt die y-Koordinaten der aktuellen Turtle Position zurück.
distance(x,y)		Gibt die Entfernung der Turtle zum Punkt (x, y) zurück.
towards(x,y)		Gibt die Richtung der Turtle (in Grad) zur Position (x, y) zurück.
moveTo(x,y)		Verschiebt die Turtle an den Punkt (x, y). Beim Verschieben zeichnet die Turtle.
setPos(x,y)		Verschiebt die Turtle an den Punkt (x, y), ohne dabei zu zeichnen.
setX(x)		Verschiebt die Turtle an die x-Koordinate, ohne dabei zu zeichnen.
setY(y)		Verschiebt die Turtle an die y-Koordinate, ohne dabei zu zeichnen.
clean()		Löscht die Turtlespuren. Die Turtle bleibt am Ort sichtbar.
clearscreen()	cs()	Löscht die Turtlespuren und setzt die Turtle zurück an die Anfangsposition in der Mitte der Zeichenfläche.
clear()		Löscht die Turtlespuren. Die Turtle bleibt am Ort und macht sich unsichtbar.
Screen().onkey(Funktion, "Taste")		Ruft die angegebene Funktion beim Drücken der angegebenen Taste auf. Schreibe den Funktionsnamen ohne Klammern. Damit die Tastendrucke von deiner externen Tastatur abgefangen werden können, musst du in die Zeichenfläche klicken nachdem du das Programm gestartet hast. Rufe anschliessend die Funktion Screen().listen() auf, damit die Zeichenfläche auf deine Tasteneingabe hört. Beispiel: Def moveForward(): Fd(50)

		Screen().onkey(moveForward, "w") Screen.listen()
Screen().onclick(Funktion)		Ruft die angegebene Funktion beim Klicken auf die Zeichenfläche auf. Schreibe den Funktionsnamen ohne Klammen. Er werden automatisch die x- und y-Koordinaten des Klicks der Funktion übergeben. Rufe anschliessend die Funktion Screen().listen() auf, damit die Zeichenfläche auf deine Eingabe hört. Beispiel: Def goTo(x,y): moveTo(x,y) Screen().onclick(goTo) Screen().lsiten()
Screen().listen()		Fängt Klicks und Tasteneingaben in der Zeichenfläche ab. Diese Funktion wird zusammen mit Screen().onclick(..) und Screen().onkey(..) verwendet. Es genügt, wenn du diese Funktion nur einmal pro Programm aufrufst.

Copyright

Die Applikation wird aktiv vom ABZ der ETH Zürich sowie dem TGT der Universität Trier entwickelt. Die Benutzung der Applikation ist kostenlos, sie darf jedoch nicht für kommerzielle Zwecke genutzt werden. Jede Anpassung oder Kopie der Applikation muss schriftlich bewilligt werden.