❖turtle.onclick() function in Python

The turtle module provides turtle graphics primitives, in both object-oriented and procedure-oriented ways. Because it uses Tkinter for the underlying graphics, it needs a version of Python installed with Tk support.

- **turtle.onclick()**

This function is used to bind fun to a mouse-click event on this turtle or on canvas.

- **Syntax :**

```
turtle.onclick(fun, btn=1, add=None)
```

Parameters:

Arguments	Description
fun	a function with two arguments, to which will be assigned the coordinates of the clicked point on the canvas
btn	number of the mouse-button defaults to 1 (left mouse button)
add	True or False. If True, the new binding will be added, otherwise, it will replace a former binding

Example 1 :

```
import turtle

# method to action

def fxn(x,y):

        # some motion

        turtle.right(90)

        turtle.forward(100)

# turtle speed to slowest

turtle.speed(1)
```

```
# motion
turtle.fd(100)
# allow user to click
# for some action
turtle.onclick(fxn)
```

❖turtle.ondrag() function in Python

The turtle module provides turtle graphics primitives, in both object-oriented and procedure-oriented ways. Because it uses tkinter for the underlying graphics, it needs a version of Python installed with Tk support.

turtle.ondrag()

This function is used to bind fun to mouse-move event on this turtle on canvas.

Syntax :

turtle.ondrag(fun, btn, add)

Parameters :

- **fun :** a function with two arguments, to which will be assigned the coordinates of the clicked point on the canvas
- **btn :** number of the mouse-button defaults to 1 (left mouse button)
- **add :** True or False. If True, new binding will be added, otherwise it will replace a former binding
- Below is the implementation of the above method with an example :

Example :

```
# importing package
```

```
import turtle
# method to call on drag
def fxn(x, y):
        # stop backtracking
        turtle.ondrag(None)
        # move the turtle's angle and direction
        # towards x and y
        turtle.setheading(turtle.towards(x, y))
        # go to x, y
        turtle.goto(x, y)
        # call again
        turtle.ondrag(fxn)
# set turtle speed
turtle.speed(10)
# make turtle screen object
sc = turtle.Screen()
# set screen size
sc.setup(400, 300)
# call fxn on drag
turtle.ondrag(fxn)
# take screen in mainloop
sc.mainloop()
```