



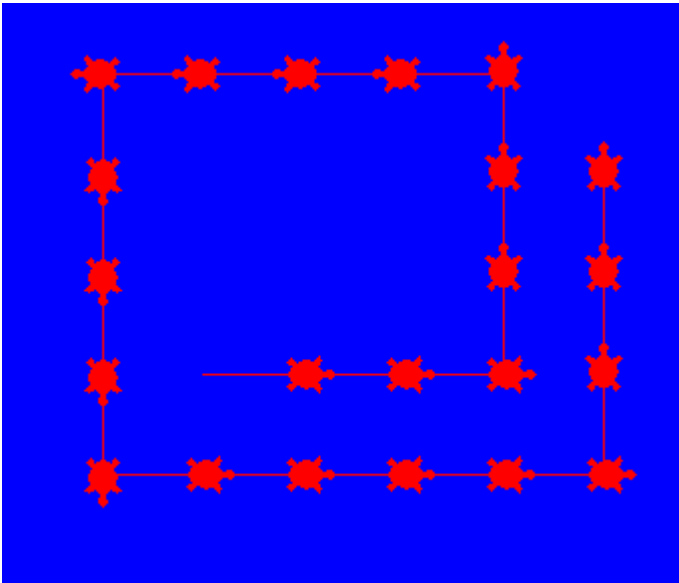
Lesson 16: Keyboard Control (Function onkeypress())

Today we continue to make our programs more interactive. As we told before event-driven programs basically do nothing, waiting until something –an event – happens. When an event does happen, they spring into action, doing whatever is necessary to handle the event. Python's turtle module includes some functions for handling user events, including mouse clicks and keypresses. **Now** investigate the Turtle module **function**: `wn.onkeypress()` or shortly `wn.onkey()`. The function `onkey` expects two arguments: a function and a keyboard value. The keyboard value is either a string containing the single character on a given keyboard key or a string reserved for a special key, such as an arrow key. The `onkey` function binds a given key to a given function, meaning that when the key is released, the function is called. For example, if you issue the command `onkey(down, 'a')`, where `down` is the name of the function. Python binds the 'a' key to the `down` function, and the pen is placed down when the user releases the 'a' key. You need to set up all your event handlers for the different keys and then call the `listen` function to start listening for events. To experiment with key events, we can bind the arrow keys to move the turtle forward or backward, turn it left or right, and clear the drawing, as shown in the following example:

1. Example #1

```
1 import turtle
2 t1=turtle.Turtle('turtle')
3 t1.color('red')
4 wn=turtle.Screen()
5 wn.bgcolor('blue')
6
7 def up():
8     t1.setheading(90)
9     t1.fd(50)
10    t1.stamp()
11
12
13 def down():
14     t1.setheading(-90)
15     t1.fd(50)
16     t1.stamp()
17
18 def left():
19     t1.setheading(180)
20     t1.fd(50)
21     t1.stamp()
22 def right():
23     t1.setheading(0)
24     t1.fd(50)
25     t1.stamp()
26
27 turtle.listen()
28 turtle.onkey(up, 'Up')
29 turtle.onkey(down, 'Down')
30 turtle.onkey(left, 'Left')
31 turtle.onkey(right, 'Right')
```

RESULT:



Functions of an example #1 let user move the turtle on the screen by pressing the arrow keys instead of clicking the mouse button. We built functions for moving the turtle for each arrow keypress on the keyboard, (**up** (\uparrow), **down** (\downarrow), **left** (\leftarrow), and **right** (\rightarrow)) keyboard and then we told the computer to listen (line #27) for those keys to be pressed.

2. Example #2

This code demonstrates change the screen colour using keyboard control

CODE:

```
import turtle
wn=turtle.Screen()
wn.setup(800,800)
wn.bgcolor('black')

tl=turtle.Turtle()
tl.up()
tl.hideturtle()
tl.color('white')
tl.goto(-150,300)
tl.write('Press s-key to get blue screen', font=("Arial",20,'bold'))
tl.goto(-300,250)
tl.color('white')
tl.write('Press a-key to get red screen', font=("Arial",20,'bold'))

def blue_screen():
    wn.bgcolor('blue')

def red_screen():
    wn.bgcolor('red')

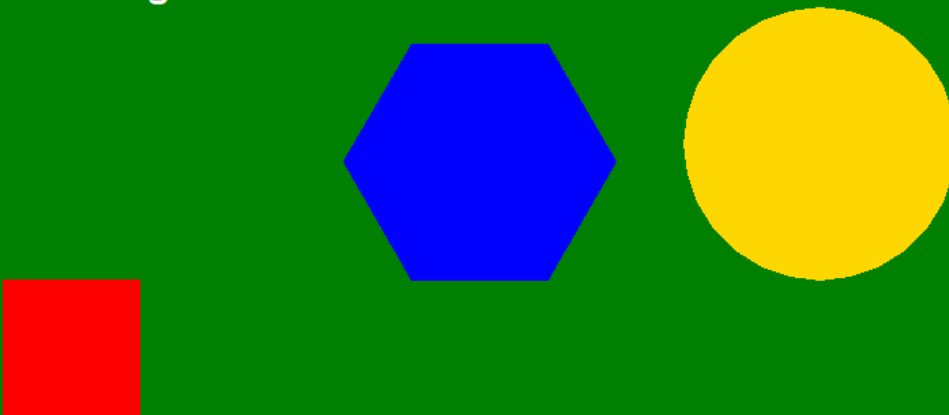
wn.listen()
wn.onkeypress(blue_screen, 's')
wn.onkey(red_screen, 'a')
```

Press s-key to get blue screen
Press a-key to get red screen

3. Example #3

RESULT:

Press space bar to clear screen
Press s-key to draw square
Press d-key to draw hexagon
Press digit 3 to draw circle



CODE:

```

import turtle
wn=turtle.Screen()
wn.bgcolor('green')
t1=turtle.Turtle()
t1.color('red')
t1.hideturtle()

t2=turtle.Turtle()
t2.color('blue')
t2.hideturtle()

t3=turtle.Turtle()
t3.color('blue')
t3.color('gold')
t3.hideturtle()
turtle.tracer(3)

t4=turtle.Turtle()
t4.up()
t4.color('white')
t4.hideturtle()
t4.goto(-300,300)
t4.write('Press s-key to draw square', font=("Arial",20,'bold'))
t4.goto(-300,250)
t4.write('Press d-key to draw hexagon', font=("Arial",20,'bold'))
t4.goto(-300,200)
t4.write('Press digit 3 to draw circle', font=("Arial",20,'bold'))
t4.goto(-300,350)
t4.write('Press space bar to clear screen', font=("Arial",20,'bold'))

# This function draw a square.
def square():
    t1.up()
    t1.goto(-300,0)
    t1.down()
    t1.begin_fill()
    for i in range (4):
        t1.fd(100)
        t1.rt(90)
    t1.end_fill()

# This function draw a hexagon.
def hexagon():
    t2.goto(0,0)
    t2.down()
    t2.begin_fill()
    for i in range (6):
        t2.fd(100)
        t2.lt(60)
    t2.end_fill()

# This function draw a circle.
def circle():
    t3.up()
    t3.goto(300,0)
    t3.down()
    t3.begin_fill()
    t3.circle(100)
    t3.end_fill()

def start_over():
    t1.clear()
    t2.clear()
    t3.clear()

wn.listen()

# The functions below draws a shape when certain keys are pressed.
wn.onkey(square, "s")
wn.onkey(hexagon, "d")
wn.onkey(circle,3)
wn.onkey(start_over,'space')

```

4. Example #4 (Control of traffic light colours)

```
import turtle

turtle.setup(800,800)
wn = turtle.Screen()
wn.title("Tess becomes a traffic light!")
wn.bgcolor("lightgreen")
t = turtle.Turtle()

t1=turtle.Turtle()
t1.up()
t1.hideturtle()
t1.color('red')
t1.goto(-100,300)
t1.write('Traffic Light', font=("Arial",20,'bold'))
t1.goto(-200,250)
t1.color('white')
t1.write('Tap spacebar to change colour', font=("Arial",20,'bold'))

def draw_housing():
    t.pensize(3)
    t.color("black","darkgrey")
    t.begin_fill()
    t.forward(80)
    t.left(90)
    t.forward(200)
    t.circle(40, 180)
    t.forward(200)
    t.left(90)
    t.end_fill()

draw_housing()

t.penup()
t.forward(40)
t.left(90)
t.forward(40)
t.shape("circle")
t.shapesize(3)
t.fillcolor("green")

state=0

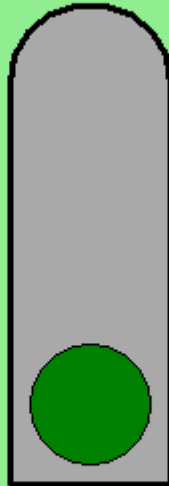
def next_state():
    global state
    if state == 0:
        t.forward(70)
        t.fillcolor("orange")
        state= 1
    elif state== 1:
        t.forward(70)
        t.fillcolor("red")
        state = 2
    else:
        t.back(140)
        t.fillcolor("green")
        state= 0

wn.onkey(next_state, 'space')
wn.listen()
```

RESULT:

Traffic Light

Tap spacebar to change colour



5. Example #5 (Combine two Events: onkey() and onclick())

```
import turtle
t = turtle.Turtle()
wn = turtle.Screen()

wn.title("How to handle mouse clicks on the window!")

wn.bgcolor("lightgreen")

t.color("purple")
t.pensize(30)
t.shape("circle")

def h(x, y):
    t.pendown()
    a=1
    t.goto(x, y)

def h1():
    t.penup()
    b=1
    t.setheading(0)
    t.fd(50)

def h2():
    t.penup()
    b=1
    t.setheading(180)
    t.fd(50)

def h3():
    t.penup()
    b=1
    t.setheading(90)
    t.fd(50)

def h4():
    t.penup()
    b=1
    t.setheading(-90)
    t.fd(50)

wn.onclick(h)
#wn.listen()
wn.onkey(h1, 'Right')
wn.onkey(h2, 'Left')
wn.onkey(h3, 'Up')
wn.onkey(h4, 'Down')
wn.listen()
```

RESULT:



6. Example #6 (Maze, for advanced users!!)

CODE: Use **Up**, **Down**, **Up**, and **Left** keyboard keys.

```
1 import turtle
2 t1=turtle.Turtle('square')
3 t2=turtle.Turtle('square')
4 wn=turtle.Screen()
5 walls=[]
6 MAP_ARRAY= [[1,1,1,1,1,1,1,1,1,1],
7             [1,0,0,0,0,0,0,0,0,1],
8             [1,0,1,1,1,1,1,1,1,1],
9             [1,0,0,0,0,1,1,1,1,1],
10            [1,1,1,1,0,1,1,1,1,1],
11            [1,1,1,0,0,1,1,1,1,1],
12            [1,1,1,1,0,0,0,0,0,1],
13            [1,1,1,1,1,1,1,1,0,1],
14            [1,0,0,0,0,0,0,0,0,1],
15            [1,0,1,1,1,1,1,1,1,1]]
16
17 wn.setup(650,650)
18 t2.shapesize(2.8)
19 t2.speed('fastest')
20 t2.penup()
21
22 for y in range(10):
23     for x in range(10):
24         q =MAP_ARRAY[x][y]
25         sx=-300+x*60
26         sy=300-y*60
27         if q==1:
28             t2.goto(sx,sy)
29             t2.stamp()
30             walls.append((sx,sy))
31
32 t1.color('blue')
33 t1.shapesize(2.6)
34 t1.speed(2.6)
35 t1.penup()
36 t1.goto(-240,-180)
37
38 def up():
39
40     X=t1.xcor()
41     Y=t1.ycor()+60
42     if (X,Y) not in walls:
43         t1.sety(Y)
44
45 def down():
46     X=t1.xcor()
47     Y=t1.ycor()-60
48     if (X,Y) not in walls:
49         t1.goto(X,Y)
50
51 def left():
52     X=t1.xcor()-60
53     Y=t1.ycor()
54     if (X,Y) not in walls:
55         t1.goto(X,Y)
56
57 def right():
58     X=t1.xcor()+60
59     Y=t1.ycor()
60     if (X,Y) not in walls:
61         t1.goto(X,Y)
62 turtle.listen()
63 turtle.onkey(up, 'Up')
64 turtle.onkey(down, 'Down')
65 turtle.onkey(left, 'Left')
66 turtle.onkey(right, 'Right')
```


7. Example #7 (Butterfly with function onclick)

```
import turtle
wn=turtle.Screen()
import time
wn.setup(1200,1100)
wn.bgcolor('pink')
wn.bgpic('grass.gif')
t1=turtle.Turtle()
t2=turtle.Turtle()
t3=turtle.Turtle()
t4=turtle.Turtle()
t5=turtle.Turtle()
t6=turtle.Turtle()
t7=turtle.Turtle()

t1.up()
t2.up()
t3.up()
t4.up()
t5.up()
t6.up()
t7.up()

image1='flo1.gif'
image2='flo2.gif'
image3='flo3.gif'
image4='flo4.gif'
image5='flo5.gif'
image6='flo6.gif'
image=[]
for i in range (118):
    #print(i)
    il=str(i)
    image.append(il+'.gif')

wn.addshape (image1)
t1.shape (image1)
t1.goto (-100,-230)

wn.addshape (image2)
t2.shape (image2)
t2.goto (-300,-420)

wn.addshape (image3)
t3.shape (image3)
t3.goto (300,-170)

wn.addshape (image4)
t4.shape (image4)
t4.goto (0,235)

wn.addshape (image5)
t5.shape (image5)
t5.goto (-150,380)

wn.addshape (image6)
t6.shape (image6)
t6.goto (400,-450)

while True:

    for i in range (18):

        wn.addshape (image[i])
        t7.shape (image[i])
        time.sleep(0.1)

    def fly(x, y):
        t7.goto(x, y)

    wn.onclick(fly)
```



https://youtu.be/tDESlsOs_0Y

