

Programming Training **kiddo**

Main Points:

- Python Statements
- Problems with selections.

print()



print(value1, value2,...)

```
print('var = ', var1)
```

prints the text 'var=' followed by the value of var

```
print('Here is a text')
```

prints the text 'Here is a text'

input()



input() returns a str value which must be converted

```
var = input('Input a str var')
```

reads a str value for var

```
var = int(input('Input var: '))
```

write the text 'Input var: ', then reads a str and converts that to int

C Arithmetic



Arithmetic operators:

- +, -, *, /, //, **
- ++, -- for incrementing and decrementing.
- % is the remainder of the integer division.

Relational operators:

- <, >, <=, >=, ==, !=

Logical: or, and, not

There are several math functions in the math module

math.sqrt(),
math.sin(), math.cos(),
math.asin(), math.acos() etc.

How to solve a problem



1. Find the inputs and outputs of the problem.
2. Ask yourself if you have done something similar.
3. Identify the elements of the solution.
4. Take a numerical example and see the changes of the variables.
5. Write the code.

Inputs - Outputs



Inputs are the elements / variables to start from.

- Inputs must be read.
- Input can be arguments within functions.

Output(s) is represented by the problem result.

- Outputs must be printed.
- Outputs can be returned by a function.

Python Statements.



Python statements give the flow of computation within a function.

1. Simple statements:

- expression - to evaluate a Python expression.
- block - used to collect several statements in one block.
- Jump:
 - break - jumps outside of repetitions.
 - return - gets outside of functions/routines

2. Selections

- simple if - selects an alternative
- complete if - selects from two alternatives

Python Blocks

Several lines which are identically indented form a block.

A block always starts after ‘:’

A block is usually required in
functions and statements.

INDENTATION MUST BE SIMILAR

```
line1 of code  
line2 of code  
....  
line of code :
```

```
    block line1  
    block line2  
    ....  
    # end of block
```


Python tests - if statement.

Select one choice

```
if test :  
    # block of if  
    statement1;  
    statement2;  
    ...  
# end if
```

Choose between two choices

```
if test :  
    # block of if  
    statement1;  
    statement2;  
    ...  
else :  
    # block of else  
    statement 1;  
    statement 2;  
    ...  
# end if
```

Choose between multiple choices

```
if test1 :  
    # block of choice 1  
    statement1;  
    statement2;  
    ...  
elif test2 :  
    # block of choice 2  
    statement 1;  
    statement 2;  
    ...  
elif test3 :  
    # block of choice 2  
    statement 1;  
    statement 2;  
    ...  
# end if
```

Minimum of two numbers.



Example: Find the maximum/minimum value of a,b

if $a < b$:

$\text{max} = b$; $\text{min} = a$;

else

$\text{max} = a$; $\text{min} = b$;

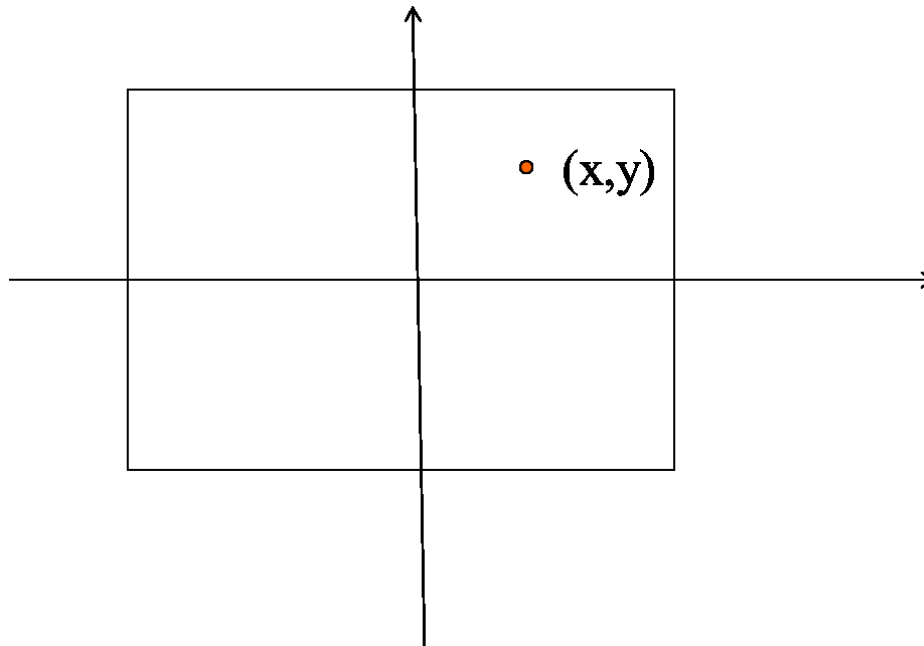
endif

Screen

The screen is a rectangle of pixels on which we can draw.

The screen origin is usually in the centre.

Each pixel will have two coordinates which are float numbers.



Python Turtle

turtle – a Python module / library for drawing.

What can turtles (as animals) do?

Move forward or backward.

Turn left or right.

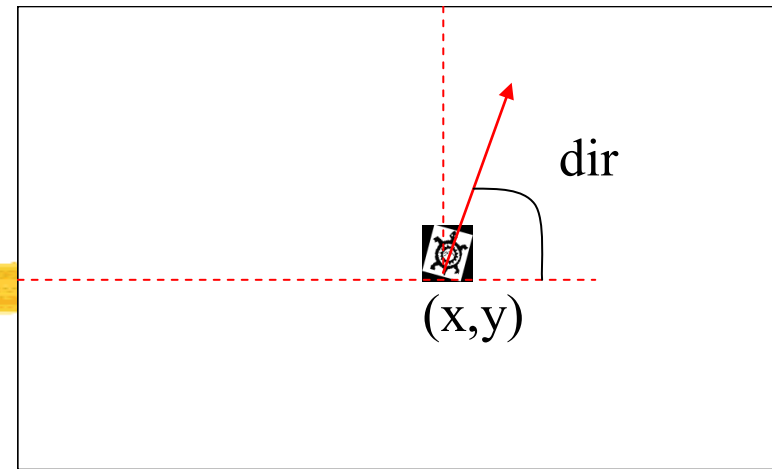
Retreat the head into the shell.

Turtle-like Graphics Pens have:

$x, y \rightarrow$ give the turtle pen coordinates.

$dir \rightarrow$ the direction to go in radians.

$pen \rightarrow$ the pen's status ($pen=1$ draws line)



Python Turtle

turtle – a Python module / library for drawing.

Function to Move and Draw

forward(d) – pen moves forward for distance d on the direction to go

backward(d) – pen moves backward for distance on the direction to go

right(angle) – pen turns right with angle (degrees)

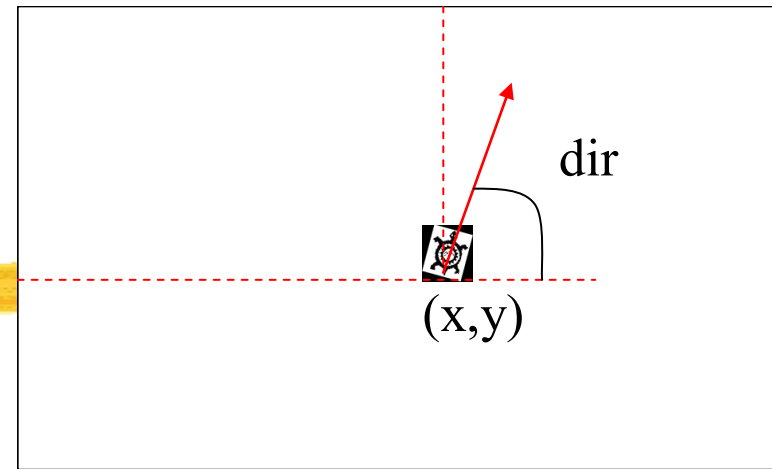
left(angle) – pen turns left with angle (degrees)

goto(x, y) – pen moves to the point (x,y) drawing a line

dot(size, color) – pen draws a dot of specified size with the given color

circle(radius) – pen draws a circle with the specified radius

circle(radius)



Python Turtle

turtle – a Python module / library for drawing.

Function to Control the pen

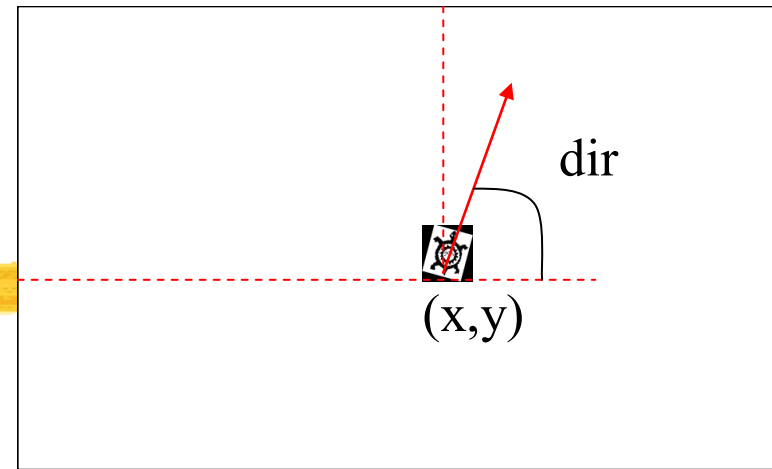
penup() – pen moves up so no drawing

pendown() – pen goes down so it can draw

width(w) – sets the thickness of the line

color(colorname) – pen color is set as colorname

color(r, g, b) – the color is set based on the r, g, b values.



Python Turtle

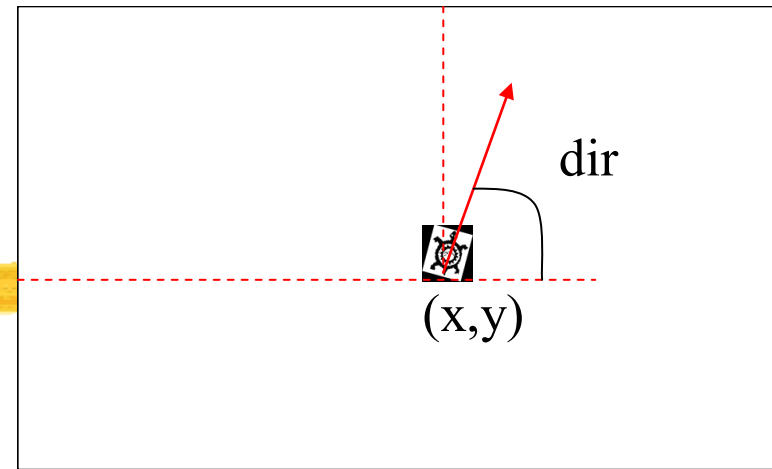
turtle – a Python module / library for drawing.

Function to Control the Screen

bgcolor(color) – the screen background color is set to color.

clear() – the screen is cleared.

screensize() – it returns the width and the height of the screen



Python Turtle



How to work with:

1. Make a pen object

```
pen = Pen()
```

2. Set the pen features like color, width, etc

```
pen.color('red')
```

```
pen.width(3)
```

3. Make you drawing using the move/draw functions

```
pen.forward(100)
```

```
pen.left(90)
```


Python Turtle



1. What is the outcome of

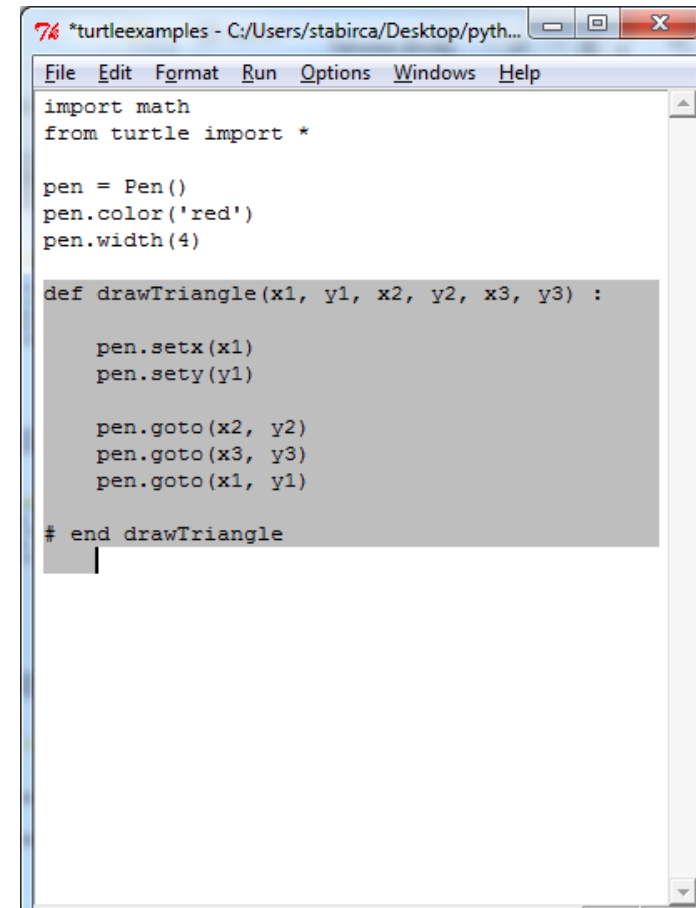
```
pen.forward(100)
pen.left(90)
pen.forward(100)
pen.left(90)
pen.forward(100)
pen.left(90)
pen.forward(100)
pen.left(90)
```

1. Can be re written as

```
for i in range(4) :
    pen.forward(100)
    pen.left(90)
```

Python Turtle – Draw a Triangle

1. Make a pen and set some features
2. Draw the triangle A(x1, y1)B(x2, y2)C(x3,y3)
 - move the pen to A(x1, y1)
 - Draw AB using goto(x2, y2)
 - Draw BC using goto(x3, y3)
 - Draw CA using goto(x1, y1)



```
*turtleexamples - C:/Users/stabirca/Desktop/pyth...
File Edit Format Run Options Windows Help
import math
from turtle import *

pen = Pen()
pen.color('red')
pen.width(4)

def drawTriangle(x1, y1, x2, y2, x3, y3) :
    pen.setx(x1)
    pen.sety(y1)

    pen.goto(x2, y2)
    pen.goto(x3, y3)
    pen.goto(x1, y1)

# end drawTriangle
```

To do List



1. Read more about it from the e-tutorial.
2. Solve the HW problems.