NYJC Workshop

Python Programming introduction

Matthieu DE MARI – 25/11/2020

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

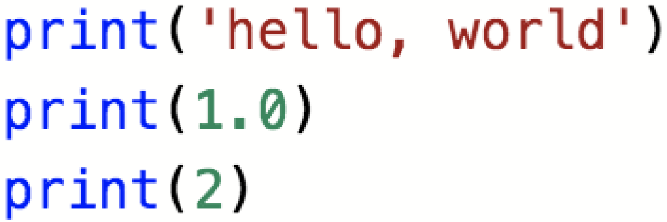
Programming concept – Commands tell the computer to do tasks for you

Programming concept – The computer is stupid; it does exactly what you tell it to do

Hello, World!

The **print** function displays information on the screen. Information can have different **datatypes**, **string**, **float** and **integer**.

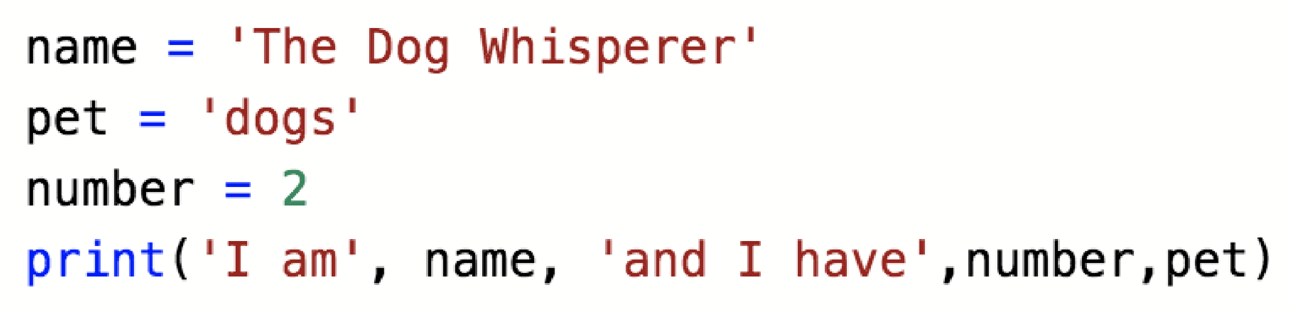
A string must be placed between matching single or double quotes.



```
print('hello, world')
print(1.0)
print(2)
```

You can store data in variables

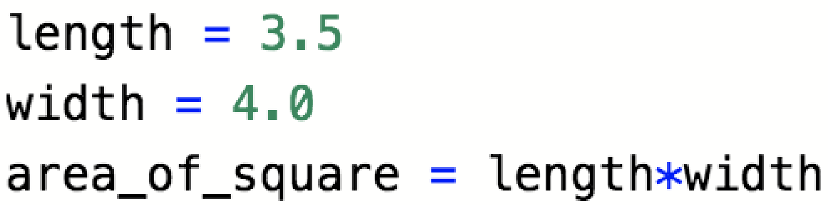
To store reuse data, you assign them to **variables**.



```python
name = 'The Dog Whisperer'
pet = 'dogs'
number = 2
print('I am', name, 'and I have',number,pet)
```

You can do numerical operations

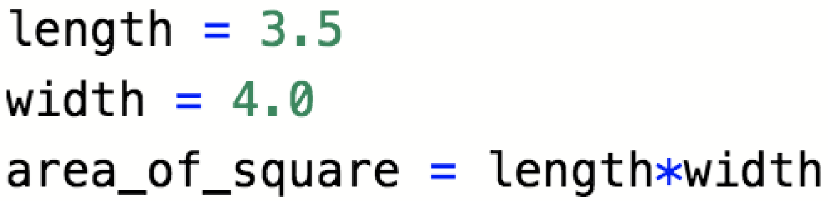
You can use the **+ - * /** operators to carry out numerical operations between integers and/or floats, and store the result in another variable.



```
length = 3.5
width = 4.0
area_of_square = length*width
```

Question

What code would you type in to
(a) Calculate the perimeter?
(b) Display the results on the screen?

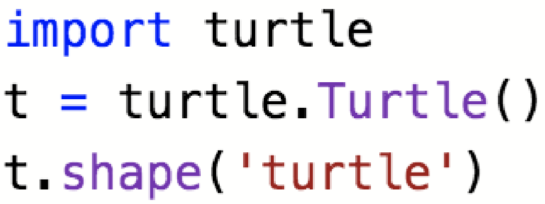


```
length = 3.5
width = 4.0
area_of_square = length*width
```

Get your turtle ready

Type the following commands. You need not worry if you find it hard to understand.

1. Import the turtle library
2. The variable t now refers to the triangle-shaped cursor on the screen.
3. Make the cursor look like a turtle.

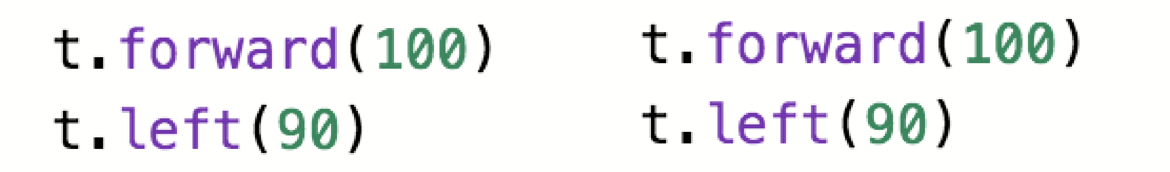


```
import turtle
t = turtle.Turtle()
t.shape('turtle')
```

Move your turtle

As your turtle moves, it will draw a line. You can also get your turtle to turn.

Which set of commands gets the turtle to draw a **vertical line**?



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

A

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

B

Both A and B

C

Programming concept – Nothing happens unless a command is executed

Programming concept – Commands are executed in sequence

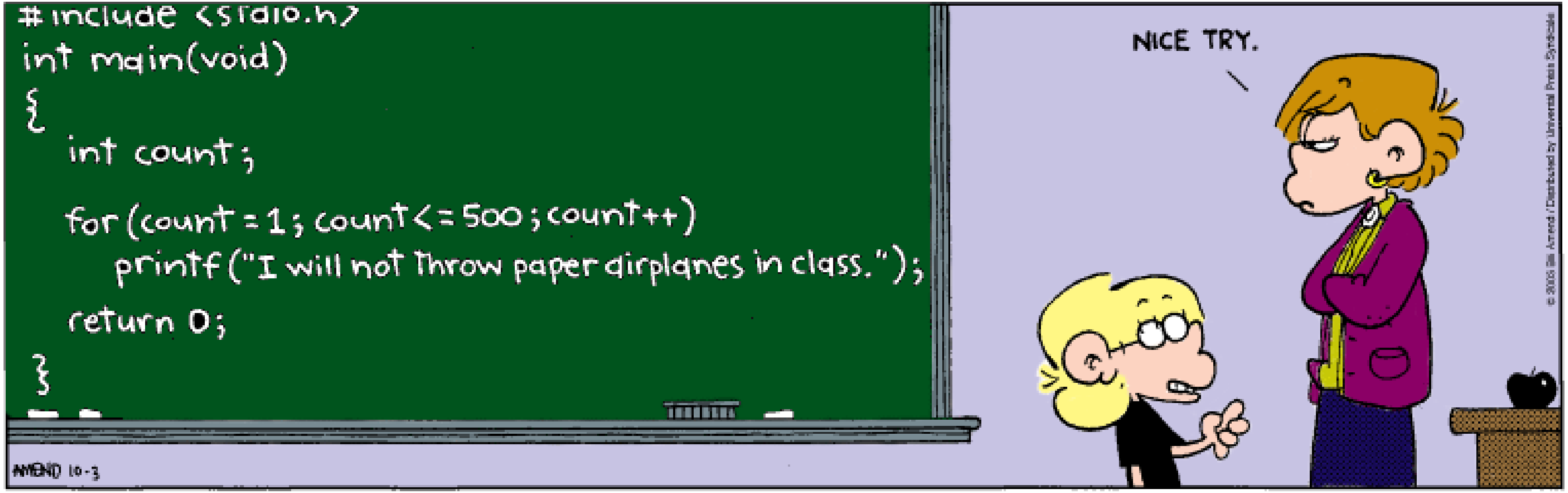
SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Draw A Square

With the commands **`forward()`** and **`left()`**, you know enough to draw a square.

Do try it! Or your instructor might also demonstrate it.

Repeating commands



```
#include <stdio.h>
int main(void)
{
   int count;

   for(count=1;count<=500;count++)
      printf("I will not throw paper airplanes in class.");
   return 0;
}
```

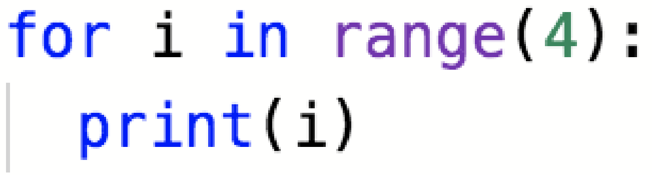
NICE TRY.
AMEND 10-3
© 2003 Bill Amend / Distributed by Universal Press Syndicate

This comic is from foxtrot.com ☺

Can you say what the programming language is?

Repeating commands

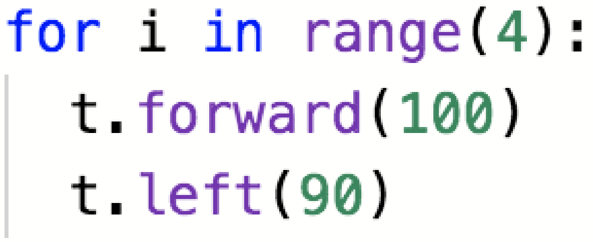
We can repeat commands using a **loop**. The variable i is known as the index. What is displayed on the screen?



```
for i in range(4):
    print(i)
```

Repeating commands

You notice that the commands to draw a square are repeated. We use a **loop** to repeat the commands**.** What is the advantage of using a loop?



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

Draw a hexagon (or any polygon)

You now know enough to draw any regular polygon e.g. equilateral triangle, hexagon etc

➔ Do try it! Or your instructor might also demonstrate it.

To get your turtle to move slower, try this command:

```
t.speed(1)
```

It takes in a number from 0 to 10, 10 being the fastest.

Programming concept – A loop repeats a set of commands

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Programmers read the manual

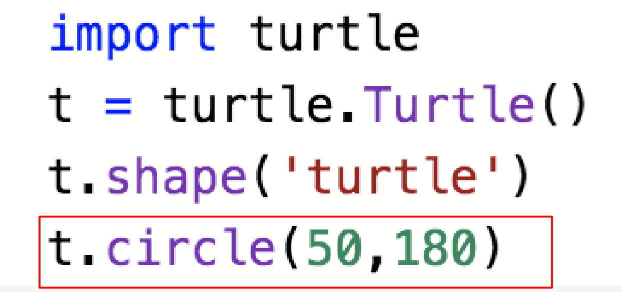
To learn more about the commands available, the python manual is here:

https://docs.python.org/3.3/library/turtle.html

However, we will continue to introduce some nice commands to you.

Draw an arc

The command **circle(radius, extent)** draws an arc. For example, the following command draws a semicircle of radius 100.

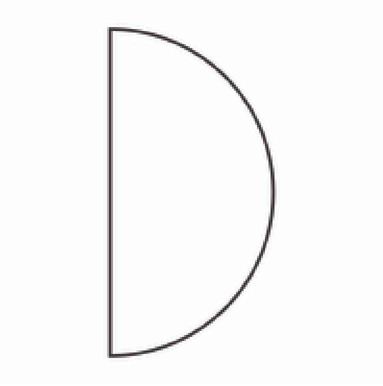


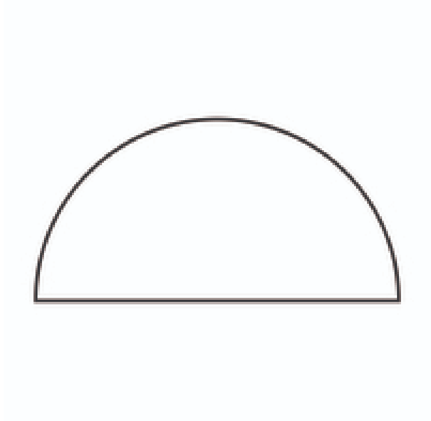
```
import turtle
t = turtle.Turtle()
t.shape('turtle')
t.circle(50,180)
```

Draw these shapes

You now know enough to draw these shapes!

➔ Do try it! Or your instructor might also demonstrate it.





To fill the shape with colour, try this

```
t.fill_color("green")
t.begin_fill()
#Your commands to draw
t.end_fill()
```

More drawing control

To control the thickness of the line

```
t.pensize(5)
```

Experiment with the number in the brackets!

To lift up the pen
(stop drawing the line)

```
t.penup()
```

To put down the pen
(start drawing the line)

```
t.pendown()
```

The modulo operator

The modulo operator **%** gives the remainder of dividing two numbers.

What is the result?

```
print( 5 % 2 )
```

What is the result?

```
print( 4 % 2 )
```

The == operator

The comparison operator **==** evaluates to **True** or **False**

What is the result?

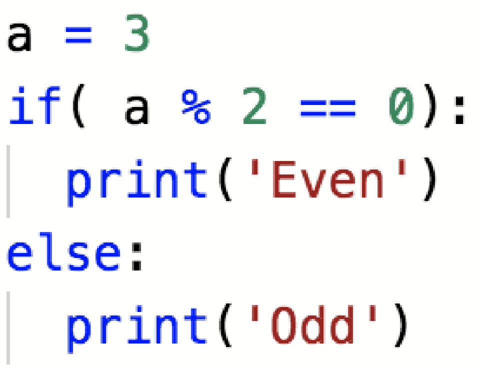
```
print( 4 % 2 == 1 )
```

What is the result?

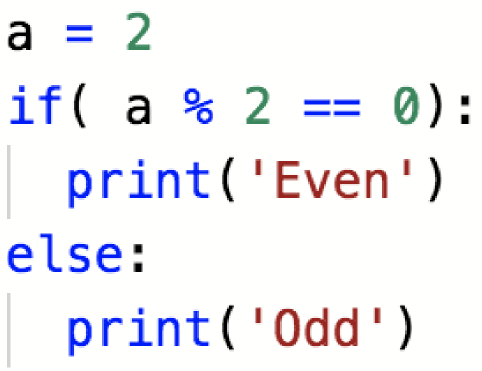
```
print( 4 % 2 == 0)
```

The if/else statement

The if /else statement tells the computer to check a condition, and does one thing if it is true, and another if it is false. What does the computer do in each of the following cases?



```
a = 3
if( a % 2 == 0):
    print('Even')
else:
    print('Odd')
```



```
a = 2
if( a % 2 == 0):
    print('Even')
else:
    print('Odd')
```

The == operator

The comparison operator **==** evaluates to **True** or **False**

What is the result?

```
print( 4 % 2 == 1 )
```

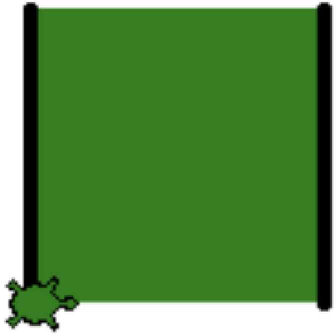
What is the result?

```
print( 4 % 2 == 0)
```

Let’s put everything together

Draw the following shape! Notice that only two sides of the square are drawn.

➔ Do try it! Or your instructor might also demonstrate it.



Programming concept – If/else statements allow the computer to choose

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Your task now:
Think of your own shape to draw
And post on padlet

SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN