

An Example of a Nested Loop

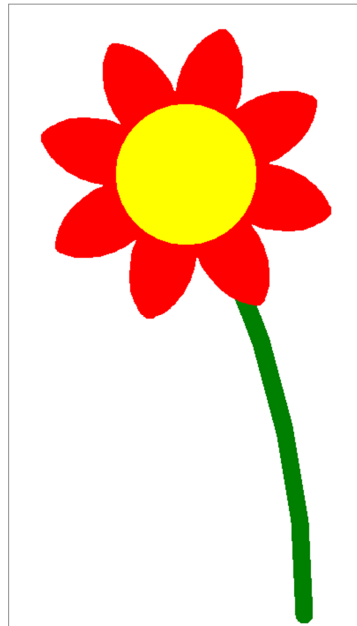
David Rossiter and Gibson Lam

Outcomes

- After completing this presentation, you are expected to be able to:
 1. Use nested while loops to create a target pattern

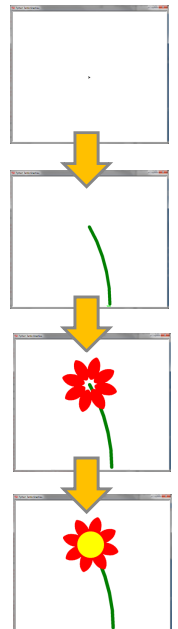
Using Nested Loops

- On the right is a flower image created by a single program
- The petals are a good example of using nested loops



The Program Stages

- Stage 1: Get the graphics started
 - Import the turtle module, fast speed
- Stage 2: Create the curved stem
 - Draw a small part of a circle
- Stage 3: Draw the petals
 - Uses a nested loop
- Stage 4: Draw the flower centre
 - Draw a yellow circle

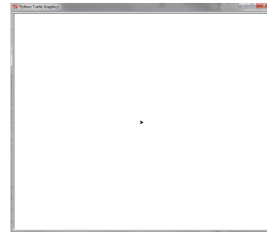


Stage 1 – Get the Graphics Started

- Like many of the programs we have seen, the first step is to import the turtle module and set some initial parameters i.e.:

```
import turtle

turtle.speed(0)
```



COMP1021

An Example of a Nested Loop

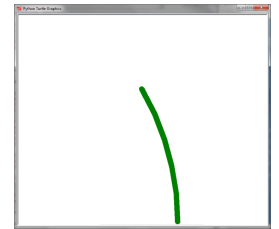
Page 5

Stage 2 – Create the Curved Stem

- We can create the stem of the flower using the `turtle.circle()` command:

```
turtle.width(20)
turtle.color("green")
```

```
turtle.up()           # Don't draw while we move
turtle.goto(100, -400) # Move the turtle to bottom right
turtle.left(90)        # Point the turtle upwards
turtle.down()          # Start drawing from now onwards
turtle.circle(1000, 30) # Draw part of a large circle
```



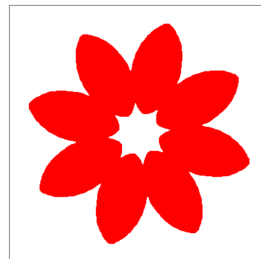
COMP1021

An Example of a Nested Loop

Page 6

Stage 3 – Draw the Petals

```
while ...condition... :
    ...statement(s)...
    while ...condition... :
        ...statement(s)...
```



- As you already know, a loop inside another loop is called a *nested loop*
- It doesn't matter what type of loop it is; any type of loop inside any type of loop is called a nested loop
- So far we know about *while* loops, in another presentation we will learn about *for* loops

Designing the Nested Loop Structure

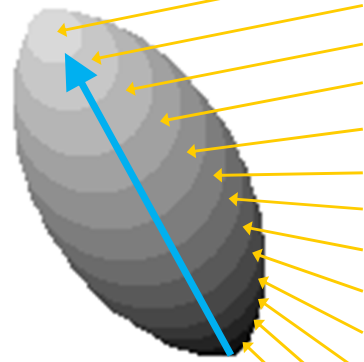
- Let's consider how we can use a nested loop

- Outer loop:** repeat 8 times, for drawing 8 petals
 - Move to the position of the first circle
 - Inner loop:** repeat 13 times, for drawing 13 circles
 - Draw a circle of the appropriate size
 - Move to the position of the next circle
 - Go backwards, to the centre position of the flower
 - Rotate the turtle by 45 degrees, ready for the next petal



- We will first show the inner loop, then the outer loop

A Petal



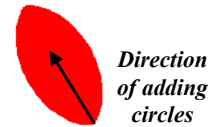
circle_number= 12 diameter= 19.5
 circle_number= 11 diameter= 36.0
 circle_number= 10 diameter= 49.5
 circle_number= 9 diameter= 60.0
 circle_number= 8 diameter= 67.5
 circle_number= 7 diameter= 72.0
 circle_number= 6 diameter= 73.5
 circle_number= 5 diameter= 72.0
 circle_number= 4 diameter= 67.5
 circle_number= 3 diameter= 60.0
 circle_number= 2 diameter= 49.5
 circle_number= 1 diameter= 36.0
 circle_number= 0 diameter= 19.5

- In this slide different shades of grey are used just to help you see the different circles
- To make the leaf shape a clever formula is used which uses the circle number to determine an appropriate diameter

The Inner Loop

`gap_between_circle = 10`
`total_circles = 13`

These 3 variables are used in the following code



```
circle_number = 0
while circle_number < total_circles:
    diameter = (circle_number + 1) * 1.5
               * (total_circles - circle_number)
    turtle.dot(diameter)
    turtle.forward(gap_between_circle)
    circle_number = circle_number + 1
```

Repeat 13 times

Calculate the diameter using a clever formula, based on the circle number (you don't need to understand the maths)

Draw a circle and then move forward (away from the center of the flower) to get in position for the next circle

`starting_distance = 40`
`total_petals = 8`

These 3 variables are used in the following code

`petal_number = 0`

```
while petal_number < total_petals:
    turtle.forward(starting_distance)
```

The code shown in the previous slide goes here

```
turtle.backward(starting_distance
    + (total_circles * gap_between_circle) )
```

```
turtle.left(360/ total_petals)
```

If there's 8 petals this angle will be $360/8 = 45$ degrees

```
petal_number = petal_number + 1
```

The turtle moves forward when it makes a petal; now go backwards to reach the flower center once again, ready for the creating the next petal



The Outer Loop

Stage 4 – Draw the Flower Centre

```
# Set the turtle drawing colour
turtle.color("yellow")
```

```
# Make a circle, using the drawing colour
turtle.dot(160)
```

```
# Sometimes we need this:
turtle.done()
```

