

CS105 Problem Solving

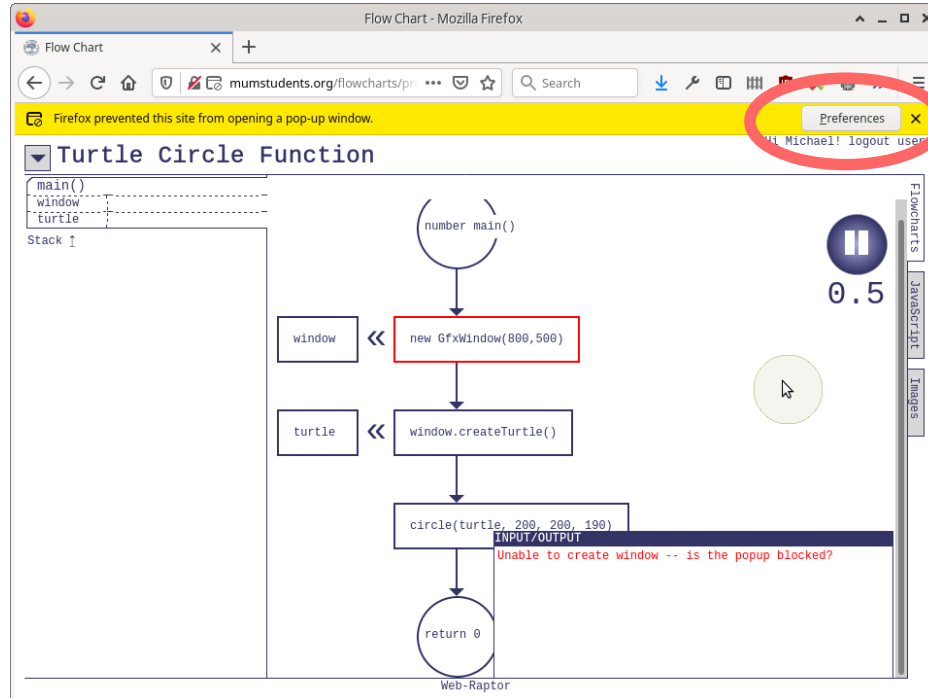
Turtle Graphics

Wholeness

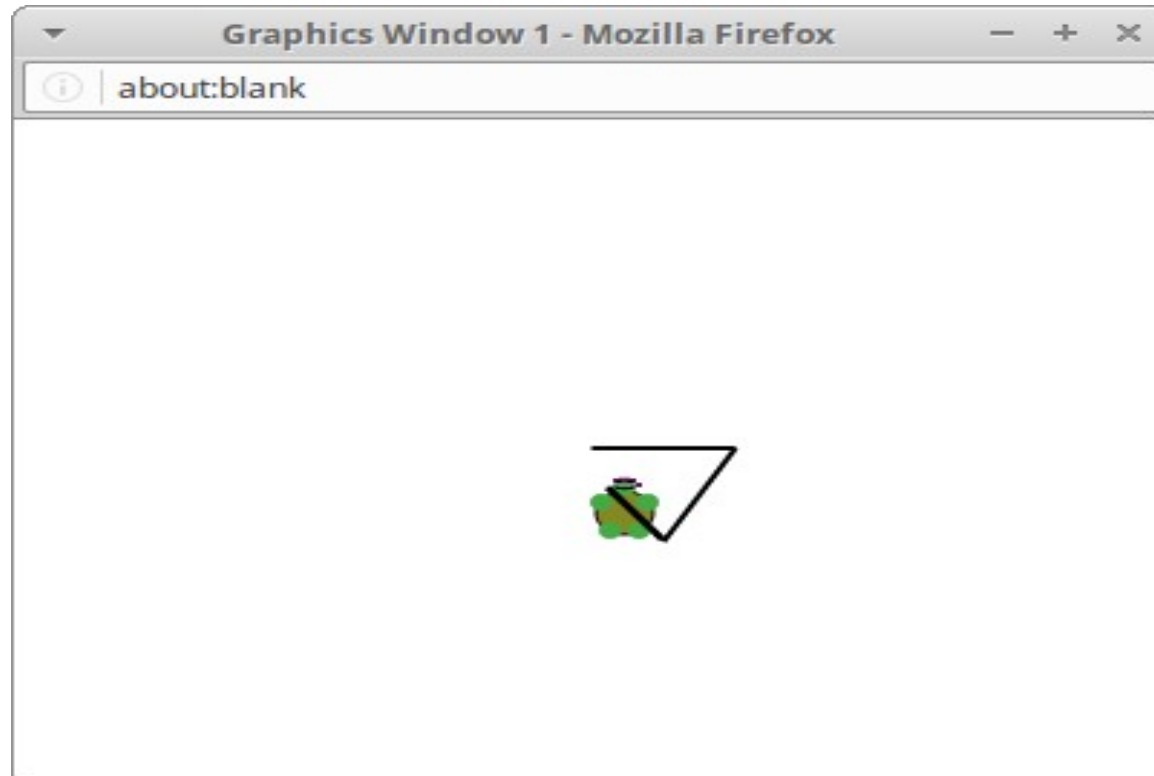
- So far we've seen text input and output. A simple way to learn how to write an algorithm is to create the instructions for making a graphical drawing.
- Turtle Graphics instructions allows us to make drawings with on the screen. This is similar to the principle of outer depends on inner – in order to show graphics on the screen (outer) a computer always need an algorithm (inner).

Popups

- Turtle Graphics uses a popup to create its separate window
- Be sure to allow popups from mumstudents.org



What it Looks Like



Flowchart Code for a Triangle

Variables:

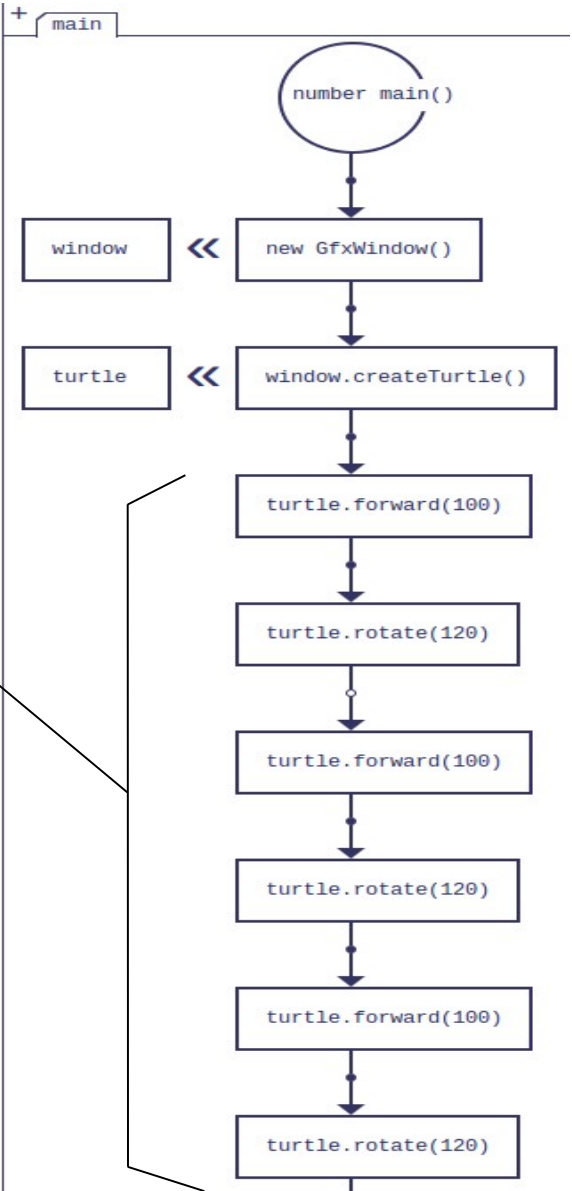
object

object

string

turtle

window



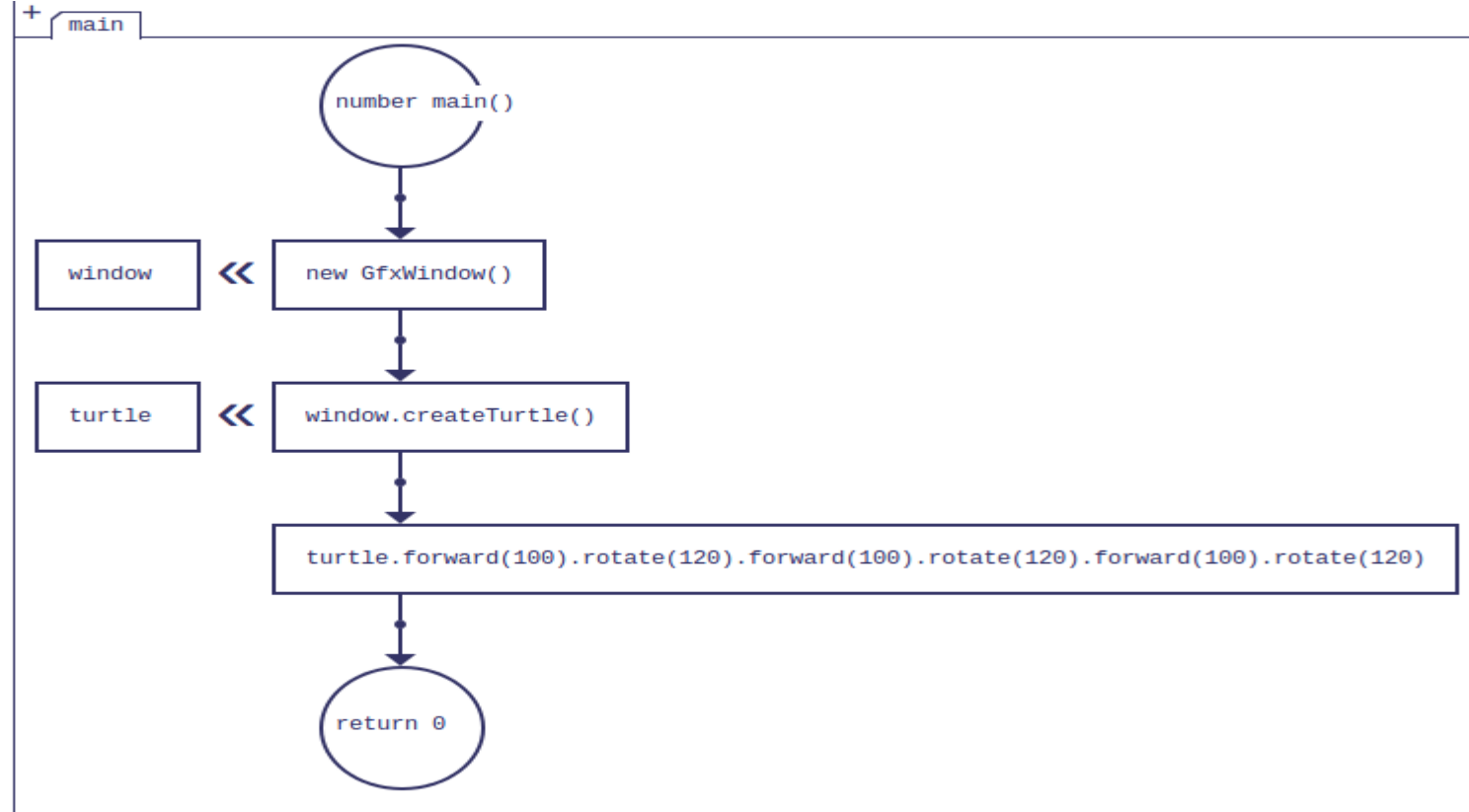
These are all "Call"
instructions

Same Program, Different Way

Variables:

object
object
string

turtle
window



Create a Window

- You can create a Graphics Window object with:



`new GfxWindow(width, height)`

- Be sure to assign the result to a variable with type object
- The width and height are optional
- Once you have your window object, you can create one or more turtle objects inside it

Create a Turtle

- You can create a turtle inside a window with:



- `win.createTurtle()`
 - Assuming that your window variable is called `win`.
 - If it's called something else then use that!
 - Be sure to also assign the result of this call to a variable of type object

Exercise

- Create a new flowchart program
 - Call the program “In Class Turtle Graphics”
 - Have it open a graphics window `window = new GfxWindow()`
 - Have it add the turtle `turtle = window.createTurtle()`

Main Point

- You can make drawings by creating a turtle graphics (popup) window, creating a turtle, and then giving the turtle commands
- These initial commands setup your environment so that you can start creating a drawing. This is similar to the principle of diving, you take the right angle (setup), and then let go.

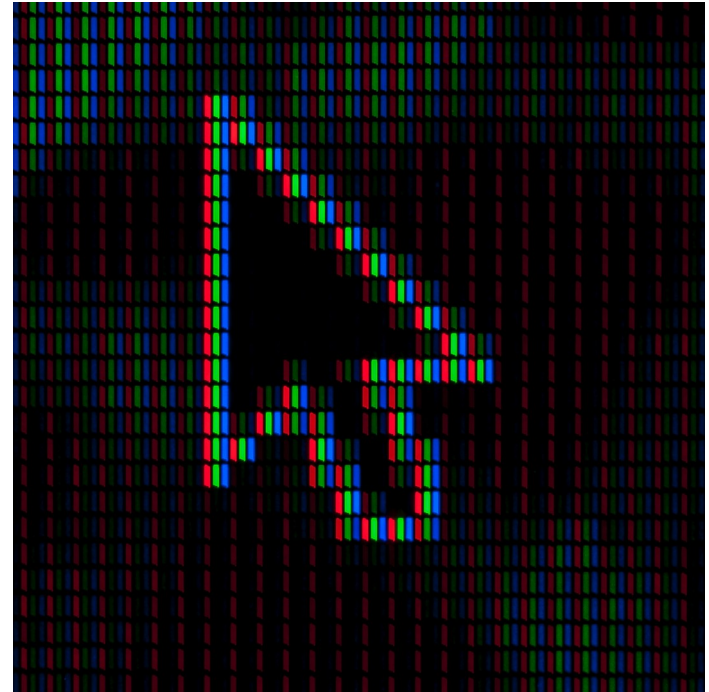
Turtle API

Application Programming Interface

- `forward(number pixels)`
- `rotate(number degree)`
- `penUp()`
- `penDown()`
- `penColor(string color)`
- `penWidth(number pixels)`
- `show()`
- `hide()`

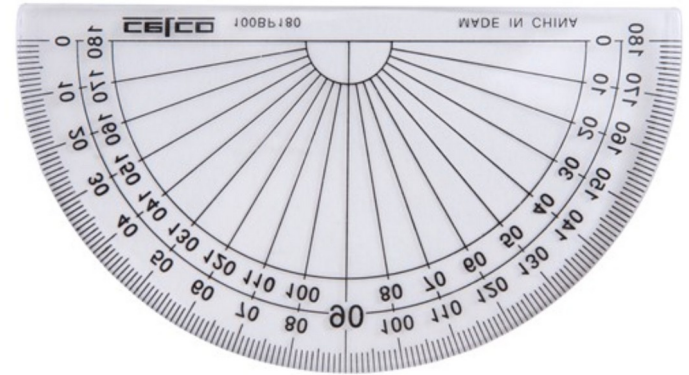
Pixels

- A computer screen is made out of pixels
 - Pixel literally means picture element
 - When you move the turtle forward you tell it how many pixels to move in its current direction



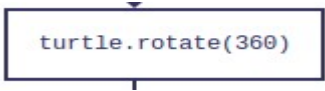
Degrees

- In mathematics every angle has a degree
 - There are 360 degrees in a circle
 - 180 degree means turn around to the other side
 - 90 degrees makes the turtle take a right turn
 - -90 degrees makes a left turn



Call element

- You can use the call element to give instructions to the turtle



A rectangular box containing the code `turtle.rotate(360)`. Above the box is a small downward-pointing arrow, and below the box is a small upward-pointing arrow.

```
turtle.rotate(360)
```

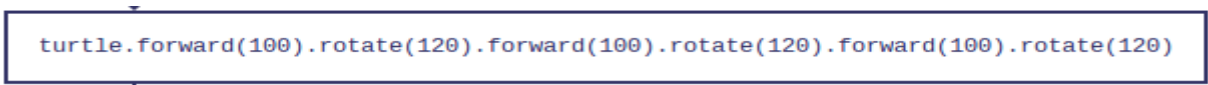
- Alternately you can also use the assignment element, any instruction to a turtle returns the updated turtle



Two rectangular boxes are shown. The left box contains the code `turtle`. To its right is a double left-pointing arrow (`<<`). To the right of the arrow is another rectangular box containing the code `turtle.forward(50)`. Above the right box is a small downward-pointing arrow, and below the right box is a small upward-pointing arrow.

```
turtle << turtle.forward(50)
```

- This is also why you can chain instructions



A single wide rectangular box containing the chained code `turtle.forward(100).rotate(120).forward(100).rotate(120).forward(100).rotate(120)`. Above the box is a small downward-pointing arrow, and below the box is a small upward-pointing arrow.

```
turtle.forward(100).rotate(120).forward(100).rotate(120).forward(100).rotate(120)
```

Demo Triangle

- Let's look at how to make a triangle

Exercise

- Update your Turtle Graphics program
 - To draw a square

Objects

- An object is a complex data type
 - Objects have their own data (variables) inside it
 - Most objects have their own functions (methods) to manipulate their data
- We've seen the methods of the turtle in the API
 - Every turtle object also has the following data:
x, y, deg, lineWidth, penIsColor, penIsDown,



Main Point

- The Turtle API (Application Programming Interface) lists the things that you can do with the turtle.
- Although you can draw almost anything, it is all done through the same API. This is an example of Harmony in Diversity.

Advanced Methods

- There are two more methods on the Turtle API
 - rotateTo(Number degree)
 - moveTo(Number x, Number y)

Rotate To

- The `.rotateTo()` method takes a degree to rotate to.
 - 0 is towards the right
 - 90 is downwards
 - 180 is towards the left
 - 270 is upwards
- This is useful if you want to 're-center' after rotating
 - Or just a different way of rotating (use it instead of normal `.rotate()`)

Move To

- The `.moveTo(x, y)` method moves the turtle in a straight line from where it is to the given coordinate on the screen
- Important to know is that 0,0 is in the top left
 - The X axis counts up towards the right
 - The Y axis counts up towards the bottom
- This is typical for computer screens
 - Very different from math, where the Y axis goes counts up upwards

Example

- Let's make a triangle with the moveTo() function

Exercise

- Make a square using only the `.moveTo()` method
 - No other methods like `rotate()` or `forward()` allowed.

Too Powerful

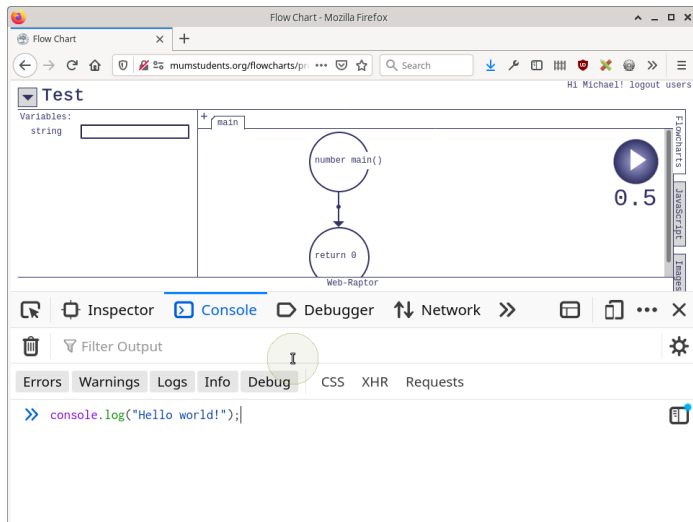
- A problem with the moveTo() method is that it lets you get away with not breaking the problem down as far
 - Don't even have to look what angle is needed
 - Not how a real / robot turtle would move
- As such you are not allowed to use it for many of the exercises
 - Being able to break a problem down is an important skill

Main Point

- The rotateTo() and moveTo() methods are powerful in that they don't care about your current position or rotation.
- Because of its power moveTo() can let you get away with not understanding deeper details. As such you are not allowed to use it for most of the homework.
- Nevertheless, for the larger homework assignments it's great to be able to quickly put things together.

Writing Code

- If you're already comfortable writing JavaScript code
 - You can also solve all the homework exercises by writing code
 - Then paste it into the JS Console on the Flowcharts page
 - Use CTRL+SHIFT+I



Demo: Writing Code

- Let's make our triangle program in code
- Then copy / paste it into the console (CTRL-SHIFT-I)

Exercise

- Try it yourself
 - Write (or copy from your flowchart) the code to make a triangle
 - Copy / Paste it into the browser console
 - This will only work if you're on the flowcharts page!

Remember writing code is fully optional at this point!

Summary

- Today we've seen:
 - How to make a graphics window and a turtle
 - The Turtle API that allows us to make drawings
 - The moveTo() method is restricted for most homework
 - Visual Studio Code can be used to write code