CSCI 141

Scott Wehrwein

Turtles

Goals

- Know how to use the `turtle` module to:
  - Create a `Turtle` object
  - Call the turtle object's methods (functions) to move it around the screen and draw simple shapes:
    (`forward, left, right, penup, pendown`)

`turtle` module

Python has a module called `turtle`!

```
import turtle
```

`turtle` module

Python has a module called `turtle`!

```
import turtle
scott = turtle.Turtle()
```

What does this do?

`turtle` module

Python has a module called `turtle`!

```
import turtle
scott = turtle.Turtle()
```

What does this do?
Let’s play with it.

Demo: basic turtle usage

Demo: basic turtle usage

- `forward`, `backward`
- `left`, `right`
- `pendown`/`down`
- `penup`/`up`

Creating and Using Objects

```
import turtle
scott = turtle.Turtle()
```

What is this about?

No new syntax here:
We import a module called `turtle`
that has a function called `Turtle`

Creating and Using Objects

```
import turtle
scott = turtle.Turtle()
```

The `Turtle()` function starts with a capital letter.
By convention this indicates that a constructor that creates (and returns) new objects of type `Turtle`.

The variable `scott` now refers to a newly created `Turtle` object.

what **is** an object? what can it **do**?

Creating and Using Objects

```
import turtle
scott = turtle.Turtle()
```

Objects can have functions associated with them, accessed via the dot notation:

```
# move the turtle forward 10 units:
scott.forward(10)
# turn the turtle left 90 degrees:
scott.left(90)
```

functions that belong to an object are called its methods

What methods do Turtles have? Lots!
Check the docs: https://docs.python.org/3/library/turtle.html

Modules vs Objects

import a module

call one of its functions

```
import random
num = random.randint(0,9)
```

import a module

call one of its functions which creates an object

call one of that object's methods

```
import turtle
scott = turtle.Turtle()
scott.forward(100)
```

Demo: make more than one `turtle`

Basic turtle methods

- `forward`: moves the turtle forward
- `left`/`right`: turns the turtle
- `penup`/`pendown`: turns drawing on and off

Algorithms with Turtles

Task: Write pseudocode for an algorithm to draw a square with side length 100:

Algorithms with Turtles

Task: Write pseudocode for an algorithm to draw a square with side length 100:

1. Move forward 100
2. Turn left 90 degrees
3. Move forward 100
4. Turn left 90 degrees
5. Move forward 100
6. Turn left 90 degrees
7. Move forward 100
8. (Turn left 90 degrees)

Can we do better?

Algorithms with Turtles

Task: Write pseudocode for an algorithm to draw a square with side length 100:

Repeat 4 times:

1. move forward 100
2. turn left 90

Demo

Demo

- turtle_square.py: Write a loop-based program that makes a turtle and draws a square with it.