CSCI 141
Computer Programming I

Filip Jagodzinski
From Last Time

```python
for lectureDays in [ "Mon", "Wed", "Fri" ] :
    print ("Lectures are on", lectureDays)
```

Q: What does this code output?
From Last Time

Q: What does this code output?

- Count the number of items in the list
- For each item in the list, the for loop’s code block (or body) is executed
- During each iteration of the for loop, the variable lectureDays has a different value, one for each item in the list
From Last Time

```python
for lectureDays in ["Mon", "Wed", "Fri"]:
    print("Lectures are on", lectureDays)
```

Q: What does this code output?

- Count the number of items in the list
  - Three items
- For each item in the list, the for loop’s code block (or body) is executed
  - Three executions of the for loop’s code block
- During each iteration of the for loop, the variable `lectureDays` has a different value, one for each item in the list
  - During each iteration, “Lectures are on” and the value of the variable `lectureDays` are printed

Lectures are on Mon
Lectures are on Wed
Lectures are on Fri
Functions ... Just as the `range` and `print` functions have a name, and take a specified number of “inputs” and produce an “output” or perform a task ...
From Last Time

Functions ... Just as the `range` and `print` functions have a name, and take a specified number of “inputs” and produce an “output” or perform a task ...

\[
\begin{align*}
\text{range} & \quad 0,100 & \quad \text{A list of numbers} \\
\text{print} & \quad x & \quad \text{Outputs “text” to the screen}
\end{align*}
\]

... if you want to write your “own” function to perform a sequence of calculations, you must ...

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”

```python
def name(parameters):
    statements
```
Define a function name `printUserWord`, which prompts the user to input a word, and also prompts the user to specify how many times that word should be printed.

The function should then print that word to the screen as many times as the user has indicated.

**Live coding demonstration**
Today

Turtle Graphics
More about functions
Turtle Graphics

As already mentioned ...

When you declare and assign a value to a variable ...

\[
aNumber = 32
\]

Q: What does that mean?
Q: How should you think of it?
Turtle Graphics

As already mentioned ...

When you declare and assign a value to a variable ...

```plaintext
aNumber = 32
```

Q: What does that mean?
Q: How should you think of it?

Q: What type of object is this?
Q: If you don’t know, how would you find out?
Turtle Graphics

As already mentioned ...

When you declare and assign a value to a variable ...

\[
aNumber = 32
\]

Q: What does that mean?
Q: How should you think of it?

Q: What type of object is this?
Q: If you don’t know, how would you find out?

```
aNumber = 32
print(type(aNumber))
```
The object-oriented way to think of this is to realize that `aNumber` refers to an object of type `int`, that has a value, which can be manipulated using predefined functions.

**Q:** What’s all this got to do with Turtles?
## Turtle Graphics

<table>
<thead>
<tr>
<th>Turtle</th>
<th>Screen</th>
<th>int</th>
</tr>
</thead>
<tbody>
<tr>
<td># variable “values”</td>
<td># variable “values”</td>
<td># variable “values”</td>
</tr>
<tr>
<td># functions</td>
<td># functions</td>
<td># functions</td>
</tr>
<tr>
<td>Turtle()</td>
<td>Screen()</td>
<td>value = 32</td>
</tr>
<tr>
<td>forward(int)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>left(int)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Turtle Graphics
**Turtle Graphics**

These classes, Turtle and Screen, have code that somebody has written. To use these classes (and their functions), you ...

<table>
<thead>
<tr>
<th>Turtle</th>
<th>Screen</th>
<th>int</th>
</tr>
</thead>
<tbody>
<tr>
<td># variable “values”</td>
<td># variable “values”</td>
<td># variable “values”</td>
</tr>
<tr>
<td># functions</td>
<td># functions</td>
<td>value = 32</td>
</tr>
<tr>
<td>Turtle()</td>
<td>Screen()</td>
<td></td>
</tr>
<tr>
<td>forward(int)</td>
<td></td>
<td># functions</td>
</tr>
<tr>
<td>left(int)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Turtle</th>
<th>Screen</th>
<th>int</th>
</tr>
</thead>
<tbody>
<tr>
<td># variable “values”</td>
<td># variable “values”</td>
<td># variable “values”</td>
</tr>
<tr>
<td># functions</td>
<td># functions</td>
<td>value = 32</td>
</tr>
<tr>
<td>Turtle()</td>
<td>Screen()</td>
<td></td>
</tr>
<tr>
<td>forward(int)</td>
<td></td>
<td># functions</td>
</tr>
<tr>
<td>left(int)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
These classes, Turtle and Screen, have code that somebody has written. To use these classes (and their functions), you ...

Q: Once you’ve imported the turtle module, how do you use it?
Turtle Graphics

Q: What is achieved by writing the above code?

```python
import turtle
window = turtle.Screen()
```

turtle module

<table>
<thead>
<tr>
<th>Turtle</th>
</tr>
</thead>
<tbody>
<tr>
<td># variable “values”</td>
</tr>
<tr>
<td># functions</td>
</tr>
<tr>
<td>Turtle()</td>
</tr>
<tr>
<td>forward(int)</td>
</tr>
<tr>
<td>left(int)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Screen</th>
</tr>
</thead>
<tbody>
<tr>
<td># variable “values”</td>
</tr>
<tr>
<td># functions</td>
</tr>
<tr>
<td>Screen()</td>
</tr>
</tbody>
</table>
window is being declared as a variable, AND is being assigned the be the “output” of the Screen method in the turtle Module

Q: What does the Screen() method (function) do?
import turtle
window = turtle.Screen()

turtle module

<table>
<thead>
<tr>
<th>Turtle</th>
</tr>
</thead>
<tbody>
<tr>
<td># variable “values”</td>
</tr>
<tr>
<td># functions</td>
</tr>
<tr>
<td>Turtle()</td>
</tr>
<tr>
<td>forward(int)</td>
</tr>
<tr>
<td>left(int)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Screen</th>
</tr>
</thead>
<tbody>
<tr>
<td># variable “values”</td>
</tr>
</tbody>
</table>

| # functions |
| Screen() |
Turtle Graphics

```
import turtle
window = turtle.Screen()
myTurtle = turtle.Turtle()
```

Q: What is achieved by adding the above code in yellow?

**turtle module**

```
# variable “values”

# functions
Turtle()
forward(int)
left(int)

Screen

# variable “values”

# functions
Screen()
```
myTurtle is being declared as a variable, AND is being assigned the "output" of the Turtle method in the turtle Module

Q: What does the Turtle() method do?
Turtle Graphics

```python
import turtle
window = turtle.Screen()
myTurtle = turtle.Turtle()
```

It “Draws” a right-pointing arrow to designate the direction that a “turtle” points in

---

turtle module

<table>
<thead>
<tr>
<th>Turtle</th>
</tr>
</thead>
<tbody>
<tr>
<td># variable “values”</td>
</tr>
<tr>
<td># functions</td>
</tr>
<tr>
<td>Turtle()</td>
</tr>
<tr>
<td>forward(int)</td>
</tr>
<tr>
<td>left(int)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Screen</th>
</tr>
</thead>
<tbody>
<tr>
<td># variable “values”</td>
</tr>
<tr>
<td># functions</td>
</tr>
<tr>
<td>Screen()</td>
</tr>
</tbody>
</table>
Turtle Graphics

```python
import turtle
window = turtle.Screen()
myTurtle = turtle.Turtle()
myTurtle.forward(100)
myTurtle.left(160)
myTurtle.forward(75)
```

Q: What is achieved by adding the above code in yellow?

On the board explanation/discussion

Q: What does this entire program output?
Turtle Graphics

```python
import turtle
window = turtle.Screen()
myTurtle = turtle.Turtle()
myTurtle.forward(100)
myTurtle.left(160)
myTurtle.forward(75)
```

turtle module

<table>
<thead>
<tr>
<th>Turtle</th>
</tr>
</thead>
<tbody>
<tr>
<td># variable “values”</td>
</tr>
<tr>
<td># functions</td>
</tr>
<tr>
<td>Turtle()</td>
</tr>
<tr>
<td>forward(int)</td>
</tr>
<tr>
<td>left(int)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Screen</th>
</tr>
</thead>
<tbody>
<tr>
<td># variable “values”</td>
</tr>
<tr>
<td># functions</td>
</tr>
<tr>
<td>Screen()</td>
</tr>
</tbody>
</table>
Turtle Graphics

```python
import turtle
window = turtle.Screen()
myTurtle = turtle.Turtle()
myTurtle.forward(100)
myTurtle.left(160)
myTurtle.forward(75)
```

Q: How would you modify this code so that the “turtle” moves so that its path is a square?

On the board exercise
Turtle Graphics

```python
import turtle
window = turtle.Screen()
myTurtle = turtle.Turtle()
myTurtle.forward(100)
myTurtle.left(160)
myTurtle.forward(75)
```

```python
import turtle
window = turtle.Screen()
myTurtle = turtle.Turtle()
myTurtle.forward(100)
myTurtle.left(90)
myTurtle.forward(100)
myTurtle.left(90)
myTurtle.forward(100)
myTurtle.left(90)
myTurtle.forward(100)
```
Turtle Graphics

```python
import turtle
window = turtle.Screen()
myTurtle = turtle.Turtle()
myTurtle.forward(100)
myTurtle.left(160)
myTurtle.forward(75)
```

**Turtle**

<table>
<thead>
<tr>
<th># variable “values”</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turtle()</td>
</tr>
<tr>
<td>forward(int)</td>
</tr>
<tr>
<td>left(int)</td>
</tr>
</tbody>
</table>

Don’t worry too much about classes and objects. You’ll learn MUCH more about that in CSCI 145.

What is important is this … there a functions, that “do” stuff. A developer wrote the functions in the Turtle class.
Functions

From last time ...

Just as the `range` and `print` functions have a name, and take a specified number of “inputs” and produce an “output” or perform a task ... if you want to write your “own” function to perform a sequence of calculations, you must ...

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”

```python
def name(parameters):
    statements
```

Task 1 : Write (declare) a function that adds two numbers and prints their sum
Task 2 : Use that function in a python program
Functions

Task 1: Write (declare) a function that adds two numbers and prints their sum
Task 2: Use that function in a python program

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”
Functions

Task 1: Write (declare) a function that adds two numbers and prints their sum.
Task 2: Use that function in a python program.

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”

Q: What are good names for this function?
Functions

Task 1: Write (declare) a function that adds two numbers and prints their sum.

Task 2: Use that function in a python program.

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”

```python
def add2Nums
```
Functions

Task 1: Write (declare) a function that adds two numbers and prints their sum.
Task 2: Use that function in a Python program.

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”

Keyword `def`, which tells Python, “I am declaring my custom function”...

`def add2Nums`
Functions

Task 1: Write (declare) a function that adds two numbers and prints their sum.

Task 2: Use that function in a python program.

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”

```python
def add2Nums:
    pass
```

Q: How many parameters (inputs) does this function require?
Functions

Task 1: Write (declare) a function that adds two numbers and prints their sum

Task 2: Use that function in a Python program

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”

```python
def add2Nums(number1, number2):
    # Function implementation
```

Parentheses specify the parameter list
Functions

Task 1: Write (declare) a function that adds two numbers and prints their sum.
Task 2: Use that function in a Python program.

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”

def add2Nums(number1, number2):

Q: Can you define a function with zero parameters? With 1 parameter, with 2 parameters? With 27 parameters?

Parameters are given “local” variable names. These are the variables that will be used to refer to the inputs.

(much more on this later)
Functions

Task 1: Write (declare) a function that adds two numbers and prints their sum.
Task 2: Use that function in a python program.

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”

```python
def add2Nums(number1, number2):
    pass  # Semicolon to specify “done” declaring the function ...
```
Functions

Task 1: Write (declare) a function that adds two numbers and prints their sum

Task 2: Use that function in a python program

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”

```python
def add2Nums(number1, number2):
    # Code to add two numbers and print the sum
```

Q: What does this function do?
Q: What is the “code” that achieves this”

(on the board discussion)
Functions

Task 1: Write (declare) a function that adds two numbers and prints their sum.
Task 2: Use that function in a Python program.

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”

```python
def add2Nums(number1, number2):
    summation = number1 + number2
    print("The sum is", summation)
```

Q: Once you’ve declared this function, how do you use it?

Task: Give sample invocations of this function.
Functions

Task 1: Write (declare) a function that adds two numbers and prints their sum.
Task 2: Use that function in a python program.

- Give your function a name.
- Specify its parameters (input(s)).
- Designate what the function “does”.

```python
def add2Nums(number1, number2):
    summation = number1 + number2
    print("The sum is", summation)

add2Nums(3, 6)
add2Nums(6, -98)
```

Q: What does this python code output to the screen?
Functions

Task 1: Write (declare) a function that adds two numbers and prints their sum
Task 2: Use that function in a python program

- Give your function a name
- Specify its parameters (input(s))
- Designate what the function “does”

```python
def add2Nums(number1, number2):
    summation = number1 + number2
    print("The sum is", summation)

add2Nums(3, 6)
add2Nums(6, -98)

The sum is 9
The sum is -92
```

On the board explanation / walk through of how the code is executed
Take home exercise

Write a program that

- Declares a function that computes the fourth power of an integer
- Prompts the user for an integer input
- Uses that function on three separate occasions
Up next

More about functions