LECTURE 9

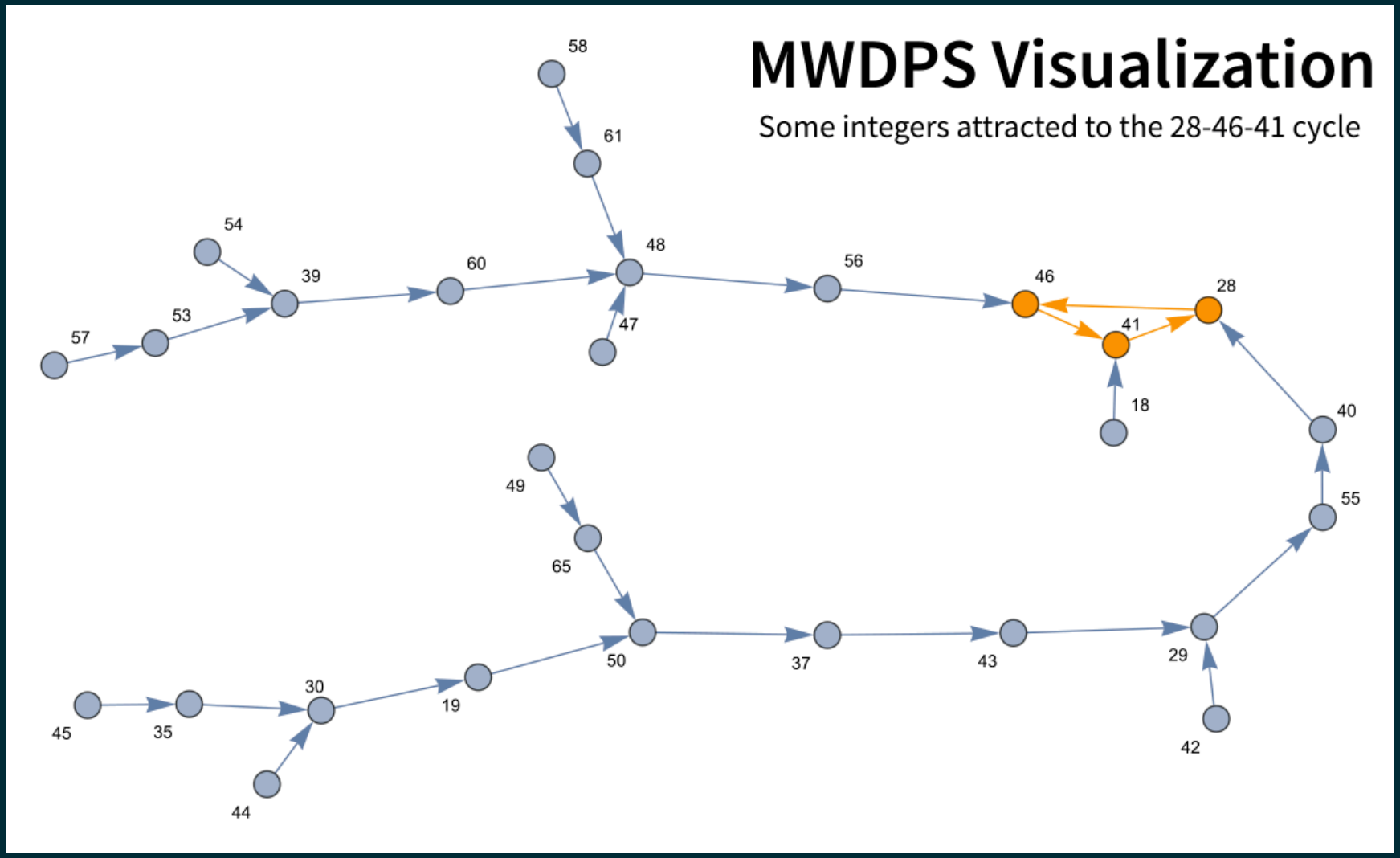
FUNCTIONS

MCS 260 Fall 2021

Emily Dumas

REMINDERS

- Project 1!
  - Due Friday 6pm
  - Submissions open now
  - Autograder enforces project description strictly; use its report to fix minor formatting issues
- Worksheet 4 coming today


MWDPS Visualization
Some integers attracted to the 28-46-41 cycle
58
61
54
48
39
60
56
46
28
53
47
41
57
18
40
49
55
65
50
37
43
29
30
19
45
35
42
44

We have seen lots of functions: `input()`, `print()`, `float()`, `len()`, `enumerate()`, ...

These are **built-in functions**, provided by Python. They do useful things, sometimes using data you provide, and sometimes returning a value.

It is also possible to create your own functions.

Syntax for a function definition:

```
def function_name(param0, param1, ...):
    statement
    ...
    statement
    return value
```

The `param`i are **parameters**.

Syntax for **calling** a function:

```
function_name(arg0, arg1, ...)
```

The `arg`i are **arguments**. The statements in the function body will run with `param0=arg0`, `param1=arg1`, ...

STRING METHODS

```
s = "Chapter 11"
print(s.lower())    # chapter 11
print(s.upper())    # CHAPTER 11
```

(We'll discuss more of these soon.)

Example: Write a function `input_yes_no()` that is like `input()` but only accepts yes or no.

Make it flexible enough to accept yes, no, y, n, with any capitalization.

Regardless of how user enters their answer, the return value should be either `"yes"` or `"no"`.

```
def input_yes_no():
    while True:
        s = input()     # Read string from keyboard
        s = s.lower()   # Make all lower case
        if s in ["y","yes"]:
            s = "yes"
            break
        elif s in ["n","no"]:
            s = "no"
            break
        else:
            print("Please enter y/yes or n/no.")
    return s
```

Now we can use this e.g. as:

```
print("Set all quiz scores to 100?")
if input_yes_no() == "yes":
    for i,student in enumerate(roster):
        scores[i] = 100.0
```

A **return** is not required; a function can perform tasks without returning a value.

A return can appear anywhere in the function body to return to the caller immediately.

```
def input_yes_no2():
    """
    Read yes/no from keyboard, allowing single letter or full
    word answers.  Returns one of the strings "yes" or "no".
    """
    while True:
        s = input()     # Read string from keyboard
        s = s.lower()   # Make all lower case
        if s in ["y","yes"]:
            return "yes"
        elif s in ["n","no"]:
            return "no"
        else:
            print("Please enter y/yes or n/no.")
```

PARAMETERS

Parameters allow a function to accept and use data. The syntax is a list of names in parentheses after the function name. Example:

```
def trim(s, maxlen):
    """Return the initial segment of sequence s,
    consisting of at most `maxlen` items."""
    return s[:maxlen]  # Works even if s is short!
```

Now if we call `trim("picnic",3)`, the body of the function runs with `s="picnic"` and `maxlen=3`.

These are called **positional arguments**, as they correspond to parameters by position.

Parameters can be given default values:

```
def increase(x, addon=5):  # Note the default value for addon
    "Return the sum of `x` and `addon` (defaults to 5)"
    return x+addon
```

When calling a function, arguments can be given positionally, or by name. The latter are **keyword arguments**.

```
increase(3)             # result is 8
increase(3,addon=1)     # result is 4
increase(addon=2,x=3)   # result is 5
increase(addon=2,11)    # ERROR: pos. args must be first
increase(addon=2)       # ERROR: arg without default omitted
```

REFERENCES

- In *Downey*:
  - Chapter 3 and Chapter 6 both discuss functions, though the latter has a lot of material we didn't cover today (e.g. recursion)
  - Section 13.5 discusses keyword args

ACKNOWLEDGEMENT

- Some of today's lecture was based on teaching materials developed for MCS 260 by Jan Verschelde.

REVISION HISTORY

- 2021-09-13 Initial publication
- 2021-09-14 Moved unused slides forward to Lecture 10