Objective

- More `for` loop
- More Functions
- Animation
- Broader Issues: Algorithm Bias

Review

- What is the *accumulator design pattern*?
- What are some variations in how we use the `print` function?
- What are benefits of functions?
- What is a *module*?
  - What are some available modules? What functionality do they have?
  - How can we find out what functionality is in a module?
- How do we get access to functionality defined in the modules (two ways)?
  - How does that choice affect how we use the functionality in our code?

Review: Using print

- `print(*objects, sep=' ', end='\n', file=sys.stdout)`

Semantics: default values for `sep` and `end` are `' '` and `'\n'`, respectively

- Examples:

```
print("Hi", "there", "class", sep='; ')
print("Put on same", end='')
print("line")
```

Output:

```
Hi; there; class
Put on sameline
```


Review: Accumulator Design Pattern

1. Initialize accumulator variable
2. Loop until done
   - Update the value of the accumulator
3. Display result

Review: Using Definitions from Modules

- To use the definitions in a module, you must first `import` the module
  - To find out what a module contains, use the `help` function
- Prepend constant or function with **`modulename`**.
  - Examples for constants:
    - `math.pi`
    - `math.e`
  - Examples for functions:
    - `math.sqrt`

`module_example_import.py`

`random` module

- Python provides the `random` module to generate pseudo-random numbers
- What is "pseudo-random"?
  - Generates a list of random numbers and grabs the next one off the list
  - A ***seed*** is used to initialize the random number generator, which decides which list to use
    - By default, the current time is used as the seed

Why use "pseudo-random" numbers?

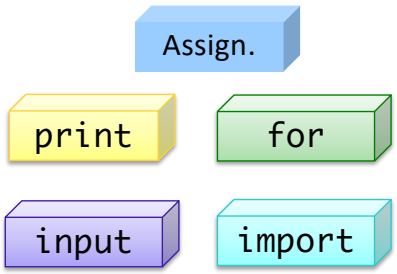
- No cost-effective source of real randomness
- Code usually doesn't *need* to be truly random
- Can replicate the code that depends on randomness by using the seed, when appropriate

VA Lottery: Pick 4

- To play: pick 4 numbers between 0 and 9
- To win: select the numbers that are selected by the magic ping-pong ball machine
- Your job: Simulate the magic ping-pong ball machines
  - Display the number on *one* line

Programming Building Blocks

- Adding to your tool set
- We can combine them to create more complex programs
  - Solutions to problems

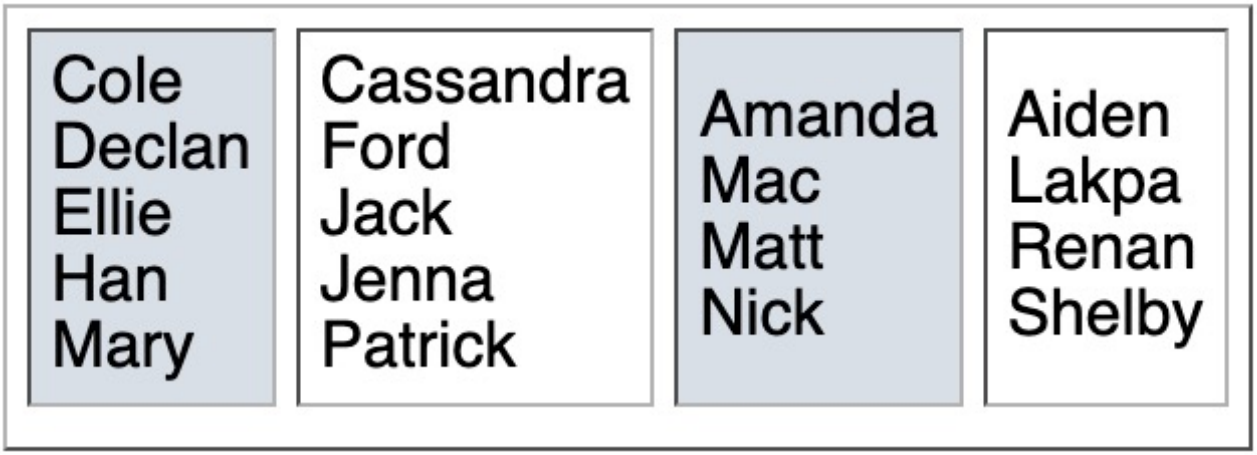


Moving a Circle According to the User

- Draw a circle in the upper left-hand corner of the screen
- Tell the user to click somewhere
- Move the circle to where the user clicked

Hmm…. Some of these steps seem very different from what we've been doing.
Can we even do them?
How can we figure out if we can?

Broader Issue Groups



11

Broader Issue: Human Bias in Algorithms

- Comment on this statement, in context of CSCI111: "Algorithms are opinions embedded in code."
- Reflect on "My department of education contact told me 'It's math and I wouldn't understand it.'"
  - Why is it beneficial to make the algorithm transparent? To keep it opaque?
- Consider the sentencing algorithm that considered likelihood of recidivism
  - What should be considered in sentencing?
  - How do we/should we "interrogate" algorithms?
- What algorithm are you questioning now?

12

Broader Issue: Human Bias in Algorithms

- Our definition of algorithms and the types of problems we solve are different than the ones described in the talk
  - Those algorithms: machine learning
    - Learn from data to categorize it or make predictions
  - Ours are likely not opinions
- BUT, you're learning more about programming and algorithms and it's a good idea to stop and question algorithms and results
  - You'll be a purchaser of software and I want you to be informed and ask good questions when making decisions
  - Yet another benefit of the liberal arts

Looking Ahead

- Pre Lab 3 due before class
- Lab 3 due Friday
- Broader Issue on Google Search for Friday