Carnegie Mellon University
Computer Science Academy

CS1: Unit 8 Review

Operators, and Functions

More Operators

- / is the division operator while // is the integer division operator
- // truncates (ignores) the part after the decimal point
  - Ex. 5 / 2 = 2.5 while 5 // 2 = 2
- The % operator is called the mod operator (or the remainder operator)
  - It computes the remainder when the first value is divided by its second value

More Math Functions

- Use the abs() function to compute the absolute value of a number
  - It is a builtin Python function
- Use the distance() function to find the distance between two points
  - Use distance(x1, y1, x2, y2) to find the distance between points (x1, y1) and (x2, y2)
- Use the angleTo() function to compute the angle from one point to another where 0 degrees is straight up and 90 degrees is to the right
  - Use angleTo(x1, y1, x2, y2) to find the angle between points (x1, y1) and (x2, y2)
- Use getPointInDir() to find the point that lies a certain distance at a given angle from another point
  - x, y = getPointInDir(x1, y1, angle, distance) where (x, y) is the point at the given distance at the given angle from the point (x1, y1)

Random Values

Random Values

- randrange(low, high) returns a random integer between low and high
  - Will never return the high value
- We can use a local variable to store random values in order to use it multiple times

Nested For Loops

More For Loops

- Previously, we have used range(n) to loop n times
- range(lo, hi, step) starts at a value of lo and will add step to its value each pass of the loop up to but not including hi

- Ex. `for i in range(0, 100, 10)` will loop 10 times and the looping variable `i` will have values 0, 10, 20, 30, … 90

Nested For Loops

- If there's an **inner loop** inside an **outer loop**, we call them collectively **nested loops**
- For each pass of the **outer loop**, the **inner loop** runs entirely
- Ex.
```
for x in range(50, 351, 100):
    for y in range(100, 301, 100):
        Circle(x, y, 10)
```
- We have **nested loops**. The **outer loop** is this:
  - `for x in range(10, 80, 50):`
- And the **inner loop** is this:
  - `for y in range(20, 90, 30):`
- On the first pass of the outer loop, x is set to 10
- Then, we run the entire inner loop, so `y` will be set to 20, then 50, then 80. And so we get circles centered at `(10, 20)`, `(10, 50)`, and `(10, 80)`
- We just completed one pass of the outer loop, so `x` is now set to 60, and we start the second pass of the outer loop
- Then, we run the entire inner loop again! So `y` will be set to 20, then 50, then 80. And so we get circles centered at `(60, 20)`, `(60, 50)`, and `(60, 80)`
- So we see: the outer loop runs only once, but for each pass of the outer loop, the inner loop runs entirely