CIS 122

Going Loopy
Loops so far...

- We've seen two types of loops
  - while loops
    - Repeat some task **while** a condition is true
    - General purpose
  - for loops
    - Repeat some task **for** each element in a sequence
    - Useful in specific scenario
Another Loopy Task

• What if we want to do some task a specific number of times?
  ○ Could use a while loop
  ○ But there's some overhead...

  ```python
  x = 0
  while x < 10:
      <do stuff>
      x = x + 1
  ```

• This is a very common task
  ○ So Python provides a shortcut
Another Loopy Task

- for loops do some task for each element in a sequence

- If we only had a sequence with exactly 10 elements
  - It would be easy to perform a task 10 times
  - The elements wouldn't even matter

```python
for x in <list of length 10>:
    <do stuff>
```
Another Loopy Task

● Python provides just the tool we need

● The `range(x)` function returns a list of integers
  ○ Starting at 0
  ○ Up to but not including x

```python
>>> range(5)
[0, 1, 2, 3, 4]
```

```python
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

● What does `range(0)` return?
Another Loopy Task

- But wait!
  - range(x) returns a list of length x

- Now we can rephrase our loop

```python
for i in range(10):
    <do stuff>
```

```python
for i in [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]:
    <do stuff>
```
Another Loopy Task

- You can use your *iterator* in your loop body

  ```python
  for i in range(10):
      print i
  ```

- But you don't have to...

  ```python
  for i in range(10):
      print "Hello World"
  ```
So many Choices

● Which loop should I choose?

● Do have a sequence you want to iterator over?
  ○ for element in sequence

● Do you know how many times you want to loop?
  ○ for x in range(n)

● None of the above?
  ○ while <some condition>
Homework Preview

- Part 0 - Summing Things Up
- Part 1 - Circular Reasoning
- Part 2 - Password Checker
- Part 3 - Guessing Game
Part 0 - Summing Things Up

- Write a function mySum(numbers)
  - Takes a list of numbers
  - Returns their sum

- What loop should we use?

- For inspiration, look over our max function from yesterday
Part 1 - Circular Reasoning

- Turtle graphics are back!

- Write a function `circle(radius)`
  - Draw circle of the given radius
  - This isn't an easy task
  - But what if we approximate our circle as a polygon

- Write a function `polygon(sides, sideLength)`
  - Draw a polygon with the given number of sides
  - Repeatedly move forward and turn
  - What loop should we use?
Part 2 - Password Checker

● Make sure passwords are sufficiently secure
  ○ At least 8 characters long
  ○ At least 1 letter
  ○ At least 2 numbers
  ○ Don't contain 'E' or 'e' (those letters are far too common)

● Write a function passwordChecker(password)
  ○ Returns False if password fails any tests
  ○ Returns True if password passes all tests
Part 2 - Password Checker

- Write helper functions to test individual cases
  - Does this string contain a letter?
  - Does this string contain two numbers?

- Call helper functions from main password checker

- What loops should we use?
Part 2 - Password Checker

- Special string methods
  - dot notation

  ```
  >>> 'a'.isalpha()
  True
  
  >>> 'b'.isdigit()
  False
  
  >>> myChar.isupper()
  ???
  ```
Part 3 - Guessing Game

● Write a function guessingGame()

● When called, Python should play a guessing game
  ○ Pick a random number
  ○ Ask the user to guess a number
  ○ If they guess wrong, give them a hint (too high, too low)
  ○ If they guess right, congratulate them
    ■ And tell them how many guesses they took

● What needs to loop?
  ○ And loop should we use?