Announcements

- Homework
  - Homework 5 is posted!
  - No scanned work on this one, everything in the repository.
  - I'm still working on grading HW4
- Still working on grade reports, but now that my other tests are graded it may actually happen!
- Polling: `rembold-class.ddns.net`

Review Question

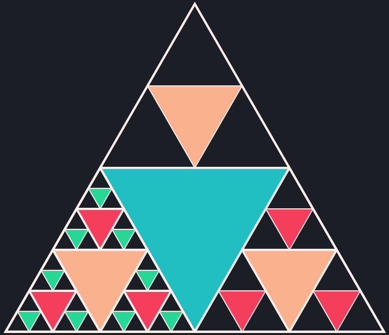
We saw on Monday that the Fibonacci sequence can be written as a recursive function using the code to the right. Suppose you then wanted to know the 5th Fibonacci number, and thus called `fib(5)`. How many times does the `fib` function get called in total before the final value is returned to you?

A) 1
B) 4
C) 6
D) 9

```
def fib(n):
  if n == 1 or n == 2:
    return 1
  else:
    return fib(n-1) + fib(n-2)
```

Not just for numbers

- Recursion is in no way restricted to numbers!
- Shows up graphically and visually
  - Fractals for example
- Can be useful in dealing with characters or strings as well



Reversing a String

- Consider a general string:

```
"This is a great string"
```

Reversing a String

- Consider a general string:

  `"This is a great string"`

- Could break it up and simplify the problem as:

Reversing a String

- Consider a general string:

  `"This is a great string"`

- Could break it up and simplify the problem as:
  - Move first character to the end

Reversing a String

- Consider a general string:

```
"This is a great string"
```

- Could break it up and simplify the problem as:
  - Move first character to the end
  - Reverse the remaining of the characters

```
reverse("his is a great string") + "T"
```

Reversing a String

- Consider a general string:

```
"This is a great string"
```

- Could break it up and simplify the problem as:
  - Move first character to the end
  - Reverse the remaining of the characters

```
reverse("his is a great string") + "T"
```

- What is the base case?

Reversing a String

- Consider a general string:

```
"This is a great string"
```

- Could break it up and simplify the problem as:
  - Move first character to the end
  - Reverse the remaining of the characters

```
reverse("his is a great string") + "T"
```

- What is the base case?
  - Only a single letter left

Reversing a String

- Consider a general string:

```
"This is a great string"
```

- Could break it up and simplify the problem as:
  - Move first character to the end
  - Reverse the remaining of the characters

```
reverse("his is a great string") + "T"
```

- What is the base case?
  - Only a single letter left
  - Return just that letter

Pruning Duplicates

- Consider the string:

  `"aabbccdeff"`

- Want to return the string with no consecutive duplicates
- Base case:
  - When a string length is 1
- If first two values duplicates, skip one
- Otherwise store the first and run the function on the rest

Reading and Writing

- Frequently want to save the result of some computation
  - Maybe time intensive and do not want to run again
  - Maybe need to manipulate results in a different program
  - Maybe just need to store to view later

Reading and Writing

- Frequently want to save the result of some computation
  - Maybe time intensive and do not want to run again
  - Maybe need to manipulate results in a different program
  - Maybe just need to store to view later
- Also need to be able to read in data from stored files

Reading and Writing

- Frequently want to save the result of some computation
  - Maybe time intensive and do not want to run again
  - Maybe need to manipulate results in a different program
  - Maybe just need to store to view later
- Also need to be able to read in data from stored files
- Different operating systems have their own file systems

Reading and Writing

- Frequently want to save the result of some computation
  - Maybe time intensive and do not want to run again
  - Maybe need to manipulate results in a different program
  - Maybe just need to store to view later
- Also need to be able to read in data from stored files
- Different operating systems have their own file systems
- Python uses file handles to refer to an accessed file
  - `name_of_handle = open('text.txt', 'w')`

The Flow of File I/O

- Start by opening a file and giving it a handle
  - Use appropriate option of 'w', 'r', or 'a' depending on your use
- Read or write lines or characters into file
  - Frequently using loops, but to necessarily
- Close the file!

```
fhandle = open('Names.txt', 'w')
for i in range(3):
    name = input('Enter a name: ')
    fhandle.write(name + '\n')
fhandle.close()
```

Tis a New Line

- Backslash is a special character in Python strings
- Signifies that the next character has special important
- `n` signifies to the Python interpreter that a new line should start at that point.
- Can look invisible when printed but *definitely* exist when comparing strings!
- Can always use `repr(some_string)` to see all the special characters

Reading from Files

- Need the option `'r'` after the filename in `open`
- Python treats the file as a sequence of lines
    - Can use `for` to iterate over all lines of the file
- Be careful reading in lines to realize that you get all special characters as well!
    - May want to use `.strip()` to remove trailing newline characters before comparisons
    - Or index it out by including everything up to but not including the last character

Intro to String Methods

- Python has a multitude of built-in string functions that you can use
- Accessed in a different manner than you are used to
  - Use “dot” notation
  - Technically methods, a distinction which we’ll talk more about later
  - Can think of it as a function where the main argument comes first, then the function name and extra arguments
- Examples:
  - `'EXCITING'.lower()`
  - `"excellent".count('e')`

Intro to String Methods

- Python has a multitude of built-in string functions that you can use
- Accessed in a different manner than you are used to
  - Use "dot" notation
  - Technically methods, a distinction which we'll talk more about later
  - Can think of it as a function where the main argument comes first, then the function name and extra arguments
- Examples:
  - `'EXCITING'.lower()`
  - `"excellent".count('e')`

Getting Help!

Curious about a potential string method you could use but don't know exactly what it does? Call the method but leave off the parentheses and add a ? at the end!