

Lab 5 Feedback

- Creative artwork
 - Getting hang of OO programming
- Not really animation--> just a direct click and move
 - If slowly moved to clicked spot, that's animation
- Efficiency of finding the alphabetical first of three strings
 - A few people wrote an efficient solution
 - Later this semester, we'll discuss the most efficient solution

Feb 26, 2008

Sprenkle - CS111

1

Lab 5 Feedback

- Refactor to type output only once
 - Separate display from computation

```
if word1 < word2 and word1 < word3:
    print "The alphabetically first word is", word1
elif word2 < word1 and word2 < word3:
    print "The alphabetically first word is", word2
else:
    print "The alphabetically first word is", word3
print "The alphabetically first word is", first
```

Vs.

```
if word1 < word2 and word1 < word3:
    first = word1
elif word2 < word1 and word2 < word3:
    first = word2
else:
    first = word3
print "The alphabetically first word is", first
```

Feb 26, 2008

Sprenkle - CS111

2

Which loop to use ...

- Prob 4: Removing spaces

```
for char in string:
    if char != " ":
        no_spaces += char
```

vs.

```
for pos in xrange(len(string)):
    if string[pos] != " ":
        no_spaces += string[pos]
```

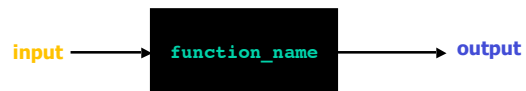
Feb 26, 2008

Sprenkle - CS111

3

Functions

- In general, a function can have
 - 0 or more inputs
 - 0 or 1 outputs
- When we define a function, we know its **inputs** and if it has **output**



Feb 26, 2008

Sprenkle - CS111

4

Function Input and Output

- This function has 1 **input**, **meters**, and 1 **output** (the converted miles)

```
def metersToMiles(meters):
    METERS_TO_MILES = .0006215
    miles = meters * METERS_TO_MILES
    return miles
```

Feb 26, 2008

Sprenkle - CS111

5

Function Input and Output

- This function has 2 **inputs**, **animal** and **sound**, and 0 **outputs**
 - It *displays* something but does not **return** anything

```
def printVerse(animal, sound):
    print BEGIN_END + EIEIO
    print "And on that farm he had a " + animal + EIEIO
    print "With a " + sound + ", " + sound + " here"
    print "And a " + sound + ", " + sound + " there"
    print "Here a", sound
    print "There a", sound
    print "Everywhere a " + sound + ", " + sound
    print BEGIN_END + EIEIO
    print
```

6

Function Input and Output

- This function has 0 **inputs** and 0 **outputs**
 - Again, it *displays* something but does not **return** anything

```
def printMenu():
    print "You have some options for what to do: "
    print "Enter an 'F' to find a song"
    print "Enter an 'S' to sort by Song title"
    print "Enter an 'A' to sort by Album"
    print "Enter an 'R' to sort by aRtist name"
    print "Enter an 'H' to list your options again"
    print "Enter a 'Q' to quit"
```

Feb 26, 2008

Sprenkle - CS111

7

Writing Comments for Functions

- Good style: Each function **must** have a comment
 - Written at a high-level
 - Include the precondition, postcondition
 - Describe the parameters (their types) and the result of calling the function (precondition and postcondition may cover this)

Feb 26, 2008

Sprenkle - CS111

8

Writing Comments for Functions

- Include the function's pre- and post-conditions
- **Precondition:** Things that must be true in order for the function to work correctly
 - E.g., num must be even
- **Postcondition:** Things that will be true when function finishes (if precondition is true)
 - E.g., the returned value is the max

Feb 26, 2008

Sprenkle - CS111

9

Example Comment

- Describes at high-level
- Describes parameters

```
# prints a verse of Old MacDonald, plugging in the animal
# and sound parameters (which are strings), as appropriate
def printVerse(animal, sound):
    print BEGIN_END + EIEIO
    print "And on that farm he had a " + animal + EIEIO
    ...
```

Feb 26, 2008

Sprenkle - CS111

10

Pre/Post Conditions

```
# pre: binary_string is a string that contains only 0s and 1s
# post: returns the decimal value for the binary string
```

```
def binaryToDecimal( binary_string ):
    exponent = len(binary_string)-1
    dec_value = 0
```

```
    for bit in binary_string:
```

```
        bit = int(bit)
```

```
        # print bit,"* 2^%d" % exponent
```

```
        dec_value += bit * (2 ** exponent)
```

```
        exponent -= 1
```

```
    return dec_value
```

Commented out
print statement
(No side effects)

Feb 26, 2008

Sprenkle - CS111

11

Defining Functions

- Consider binaryToDecimal.py
- Take code that determines if the string is a binary string and turn into a function
 - What is the function's input?
 - What is the function's output?
 - What is a good name for the function?

Feb 26, 2008

Sprenkle - CS111

12

Lab 6 Overview

- Advanced string problems
 - Using string methods, loops, ASCII values, substring operator
 - We will use some string methods that we basically implemented in the previous lab
 - See benefit of methods?
- Practice using functions
 - Comments on functions for full credit