

Objectives

- Usability
- Problems with print
- Built-in functions
- Import statements
- Puzzles from Cyberspace

Jan 22, 2010

Sprenkle - CSCI111

1

Review

- How can I covert one primitive type to another primitive type?
- What is the short cut for adding 1 to a variable and saving it in that variable?
- String operators:
 - How do I concatenate two strings?
 - How can I concatenate a string x times?
- How do I put a tab in a string?

Jan 22, 2010

Sprenkle - CSCI111

2

Review

- Type conversion
 - Use type's constructor
- Shorthands, such as $x+=1$
- String operators
 - $+$: concatenate strings together
 - $*$: concatenate the string n number of times
- Escape Sequences
 - Example: `'\t'` → prints a tab

Jan 22, 2010

Sprenkle - CSCI111

3

Software Qualities

- Beyond functionality, what qualities do you like in software?
 - Web included

Jan 22, 2010

Sprenkle - CSCI111

4

Usability

- Want users to *like* to use your software
 - More revenue
 - Develop even better software
- How Apple makes money: best user interfaces → user buys products

Jan 22, 2010

Sprenkle - CSCI111

5

Usability Goals

- Pretty output, display, layout
 - Easy to read, understand, interpret
- Clear navigation
- Easy to perform frequent tasks
- Undoability
- Difficult to make irrecoverable errors



Jan 22, 2010

Sprenkle - CSCI111

6

Problem with print

- By default, `print` puts spaces around numbers when they get printed out
 - Example:

```
x = 13.54
print "You owe $", x, "."
```

Displays:
You owe \$ 13.54 .

Extra spaces

Jan 22, 2010

Sprenkle - CSCI111

7

Solution: using str()

- Recall: `str()` is constructor/converter function to convert other data types to strings
 - Example: `str(33) → '33'`

- Use constructor with the `+` (i.e., *concatenation*) operator when printing output

```
print "You owe $" + str(x) + "."
```

Jan 22, 2010

Sprenkle - CSCI111

8

Another problem with print

```
SALES_TAX=.05 # the sales tax in VA
value = input("How much does your item cost? ")
with_tax = value * (1+SALES_TAX)
print "Your item that cost ($", value, ")",
print "costs $", with_tax, "with tax"
```

Jan 22, 2010

Sprenkle - CSCI111

sales_tax.py

9

Solution: Format Operator & Specifiers

- Format operator: `%`
- Format specifiers allow us to control how output is displayed to user
 - Right, left justification
 - Number of decimals to display

Jan 22, 2010

Sprenkle - CSCI111

10

FORMATTING STRINGS

Jan 22, 2010

Sprenkle - CSCI111

11

Formatting Strings

- Syntax is
 - ```
> <templatestring> % (<value1>,
 <value2>, ..., <valuen>)
```
- Semantics: creates a **formatted string**
  - Means "format the templatestring, using the format(s) specified by **format specifiers** on the corresponding replacement values"
- Typically used with print statements

Jan 22, 2010

Sprenkle - CSCI111

12

### Formatting Strings

- **templatestring** is a template for the resulting string with format specifiers instead of the values
  - For each format specifier in templatestring, should have a **replacement value**
  - Throws **TypeError** if not enough replacements for specifiers in templatestring
  - If only one replacement value, don't need ()

`"%.2f" % 3.14159` result is "3.14"

One format specifier      Corresponding replacement value

Jan 22, 2010      Sprenkle - CSC1111      13

### Example Format Specifiers

`"%5d" % 12`      `"%9.2f" % 23.1999`

Field width is 5      Precision is 2

Right-justified      Field width is 9

Any guesses?

- What if precision is bigger than the decimal places?
- What if field width is smaller than the length of the value?

Jan 22, 2010      Sprenkle - CSC1111      14

### Example Format Specifiers

`"%5d" % 12`      `"%9.2f" % 23.1999`

Field width is 5      Precision is 2

Right-justified      Field width is 9

- What if precision is bigger than the decimal places?
  - Fills decimal with 0s
- What if field width is smaller than the length of the value?
  - String contains entire value

Jan 22, 2010      Sprenkle - CSC1111      15

### Format Specifiers

- General format: `%[flags][width][.precision]code`
  - flags:
    - 0: zero fills
    - +: adds a + sign before positive values
    - -: left-justification (default is right-justification)
  - width:
    - *Minimum* number of character spaces reserved to display the entire value
    - Includes decimal point, digits before and after the decimal point and the sign

[ ] mean "optional"

Jan 22, 2010      Sprenkle - CSC1111      16

### Format Specifiers

- General format: `%[flags][width][.precision]code`
  - precision:
    - Number of digits after the decimal point for **floating point** values
  - code:
    - Indicates the value's **type**/way to format

| Code   | Type                         |
|--------|------------------------------|
| s      | string                       |
| d or i | integer                      |
| f      | float                        |
| e      | floating point with exponent |

Jan 22, 2010      Sprenkle - CSC1111      17

### Example using Format Operator

print "Your item that cost (\$%.2f)" % value,  
 print "costs \$%.2f with tax" % tax

Formatting operator      Format specifier      Replacement values

Jan 22, 2010      Sprenkle - CSC1111      18

## Formatting Practice

- `x = 10`
- `y = 3.5`
- `z = "apple"`
- `"%6.2f" % x`
- `"%6.2d" % x`
- `"%06.2f" % y`
- `"%+6.2f" % y`
- `"%-10s" % z`
- `"%5d %-7.3f" % (x,y)`

Jan 22, 2010

Sprenkle - CSCI111

19

## Example: Printing Out Tables

- A table of temperature conversions

| Temp F | Temp C | Temp K |
|--------|--------|--------|
| -459.7 | -273.1 | 0.0    |
| 0.0    | -17.8  | 255.4  |
| 32.0   | 0.0    | 273.1  |

- If we want to print data in rows, what is the template for what a row looks like?
  - How do we make the column labels line up?

Jan 22, 2010

Sprenkle - CSCI111

20

## Broader Issues

- Let's say I have 21 people in a class
- I want 5 groups of approximately equal size
- How many people will be in each group?

Jan 22, 2010

Sprenkle - CSCI111

21

## Four Puzzles in Cyberspace

- Context: Book *Code* v2 by Lawrence Lessig
- You read Chapter 2
  - Presents the problems, not the author's proposed solutions

|                                                |                                    |                               |                                     |                              |
|------------------------------------------------|------------------------------------|-------------------------------|-------------------------------------|------------------------------|
| Ben<br>George<br>Sirocco<br>Shannon<br>Collier | Logan<br>Amy<br>Harrison<br>Dalena | Luke<br>James<br>Hank<br>Jeni | Andrew<br>Will<br>Phil<br>Kelly Mae | CJ<br>Nick<br>Taylor<br>Dave |
|------------------------------------------------|------------------------------------|-------------------------------|-------------------------------------|------------------------------|

Jan 22, 2010

Sprenkle - CSCI111

22

## Four Puzzles from Cyberspace

- How many of you knew about MMOGs before reading this article?
  - How many of you participated in this or something similar?
  - What do people gain and lose by using an MMOG?
- Do you consider Facebook to be "cyberspace"? If so, why? If not, how close is it?
- W&L administrators tried to regulate students use of juicycampus and acb
  - Why did they feel it should be regulated?
  - What types of regulation did they do? Was it effective?
- Every advancement in technology has positive and negative effects
  - What are the positive and negative effects of email? IM? Facebook?

Jan 22, 2010

Sprenkle - CSCI111

23

## Relation to Class

- Problem solving through code

Jan 22, 2010

Sprenkle - CSCI111

24

## Looking Ahead

- Read rest of “chapter” of *Four Puzzles from Cyberspace*
  - Other two problems