

Objectives

- String Formatting
- Computer's representations of data types

Big Step Forward

- Reflection: How far have I come in Computer Science?
- A lot of String operations
 - A lot of arithmetic operations, but you're familiar with those
- As we move forward, requires a lot more "play" and practice
 - Handouts and your notes help with review
 - Textbook

The Rules

- No “I don’t know” → “I’ll figure it out”
 - We are problem-solving
 - Part of problem-solving is figuring out what you know and putting the pieces together until you solve the whole thing
 - “figuring out” step improves learning
- Break down problems into smaller pieces
 - Also part of problem solving
 - Wait on user input
 - Hardcode a value to start

Oct 25, 2017

Sprenkle - CSCI111

3

Review

- If a method or function *returns* something, what does that usually mean we should do?

Oct 25, 2017

Sprenkle - CSCI111

4

Review

- If a method or function *returns* something, what does that usually mean we should do?
 - Save in a variable
 - OR use immediately
 - Example: Included within another function
 - `print(myfunction(param))`
 - `test.testEqual(myfunction(param), expected)`

Oct 25, 2017

Sprenkle - CSCI111

5

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

Oct 25, 2017

Sprenkle - CSCI111

6

Escape Sequences

- Escape character: `\`
- Escape sequences
 - newline character (carriage return) → `\n`
 - tab → `\t`
 - quote → `\"` or `\'`
 - backslash → `\\`
- Example:
 - `print("To print a \\, you must use \"\\\\\\\\")`
 - What does this display?

[Interactive demonstration](#)

Oct 25, 2017

Sprenkle - CSCI111

`demo_str.py`

7

Practice

- Display To print a tab, you must use `'\t'`.
- Display I said, "How are you?"

`escape_sequence.py`

Oct 25, 2017

Sprenkle - CSCI111

8

FORMATTING STRINGS

Oct 25, 2017

Sprenkle - CSCI111

9

Solution: format Method

- How to use:
 - `"templatestring".format(<whattoformat>)`
- **templatestring** allow us to control how output is displayed to user
 - Right, left justification
 - Number of decimals to display

Oct 25, 2017

Sprenkle - CSCI111

10

Solution: format Method

- How to use:
 - `"templatestring".format(<whattoformat>)`
- Semantics: creates a **formatted string**
 - Means “format the `templatestring`, using the format(s) specified by **format specifiers** on the corresponding replacement values”
 - Returned as the `str` data type
- Typically used with print statements

Oct 25, 2017

Sprenkle - CSCI111

11

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 **IndexError** if not enough replacements for specifiers in `templatestring`

```
"{: .2f}".format(3.14159)
```

Evaluates to "3.14"

↑
One format specifier
in template string

↑
Corresponding replacement value

Oct 25, 2017


Sprenkle - CSCI111

12

Format Specifiers

[] mean optional

- General format:
`{[field_name]:conversion}`

 index number of the argument,
i.e., which field in the template string

- **conversion**

- conversion code of the data type

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

Default if code isn't given

Oct 25, 2017

Sprenkle - CSCI111

13

Format Specifiers

[] mean optional

Conversion options : [flags][width][.precision][code]

- **flags:**
 - 0: zero fills
 - +: adds a + sign before positive values
 - <: left-justification (default for strings)
 - >: right-justify (default for numbers)
 - ^: center
- **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
- **precision:**
 - Number of digits after the decimal point for **floating point** values

Oct 25, 2017

Sprenkle - CSCI111

14

Example using Format Operator

Format specifier

```
print("Your item that cost ${:.2f}".format(value))
print("costs ${:.2f} with tax".format(tax))
```

Alternative:

```
print("Your item that cost ${:.2f} costs ${:.2f} with tax".format(value, tax))
```

Oct 25, 2017

Sprenkle - CSCI111

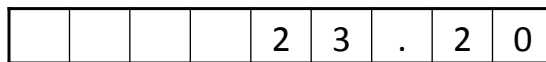
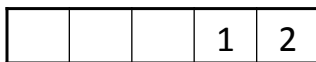
15

Example Format Specifiers

```
"{:5d}".format(12)    "{:9.2f}".format(23.1999)
```

→ " 12"

→ " 23.20"



Field width is 5

Precision is 2

Right-justified

Field width is 9

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

Any guesses? Try out in interpreter.

Oct 25, 2017

Sprenkle - CSCI111

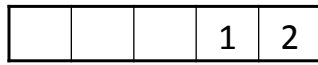
16

Example Format Specifiers

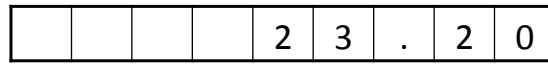
`"{:5d}".format(12)` `"{:9.2f}".format(23.1999)`

→ " 12"

→ " 23.20"



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

Oct 25, 2017

Sprenkle - CSCI111

17

Formatting Practice

- `x = 10`
- `y = 3.5`
- `z = "apple"`
- `"{:6d}".format(x)`
- `"{:6.2f}".format(x)`
- `"{:06.2f}".format(y)`
- `"{:6.2f}".format(y)`
- `"{: ^10s}".format(z)`
- `"{:5d} {:<7.3f}".format(x,y)`

Oct 25, 2017

Sprenkle - CSCI111

18

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	255.2
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?

Oct 25, 2017

Sprenkle - CSCI111

`temp_table.py`

19

REPRESENTING DATA

Oct 25, 2017

Sprenkle - CSCI111

20

Representations of Data

- Computer needs ways to represent different types of data
 - Eventually, all boils down to 1s and 0s
- Computer needs to translate between what humans know to what computer knows and back again



Oct 25, 2017

Seems like a divergence on strings but just wait...

21

Decimal Representations

- Decimal is base 10
- Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Each *position* in a decimal number represents a *power of 10*

Oct 25, 2017

Sprenkle - CSCI111

22

Decimal Representations

- Decimal is base 10
- Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Each *position* in a decimal number represents a **power of 10**
- Example: 54,087

5	4	0	8	7
10^4	10^3	10^2	10^1	10^0

- $= 5 \cdot 10^4 + 4 \cdot 10^3 + 0 \cdot 10^2 + 8 \cdot 10^1 + 7 \cdot 10^0$
- $= 5 \cdot 10,000 + 4 \cdot 1000 + 0 \cdot 100 + 8 \cdot 10 + 7 \cdot 1$

Oct 25, 2017

Sprenkle - CSCI111

23

Number Representations

Characteristic	Decimal	Binary
Base	10	2
Digits	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0, 1
Position represents	Power of 10	Power of 2

- Binary: two values (0, 1)
 - Like a light switch (either **off** or **on**) or booleans (either True or False)
- 0 and 1 are *binary digits* or **bits**
 - 64-bit machine: represents numbers (and other data) with 64 bits

Oct 25, 2017

Sprenkle - CSCI111

24

Binary Representation

- Binary number: 1101

1	1	0	1
2^3	2^2	2^1	2^0

- $= 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0$
 - $= 1*8 + 1*4 + 0*2 + 1*1$
- Decimal value: 13

Practice: what is the decimal value of the binary number **10110**?

Oct 25, 2017

Sprenkle - CSCI111

25



Looking Ahead

- Lab 6 due Friday
- Broader Issue on Friday

Oct 25, 2017

Sprenkle - CSCI111

26