Objectives

- String Formatting

- Broader Issue: Responsibility of Social Media Companies

Review

- What is the special name for sequences, like newlines, tabs, …?

  ➢ How do we represent them in strings?

- How do computers represent numbers?

- What are the various things we can do with strings?
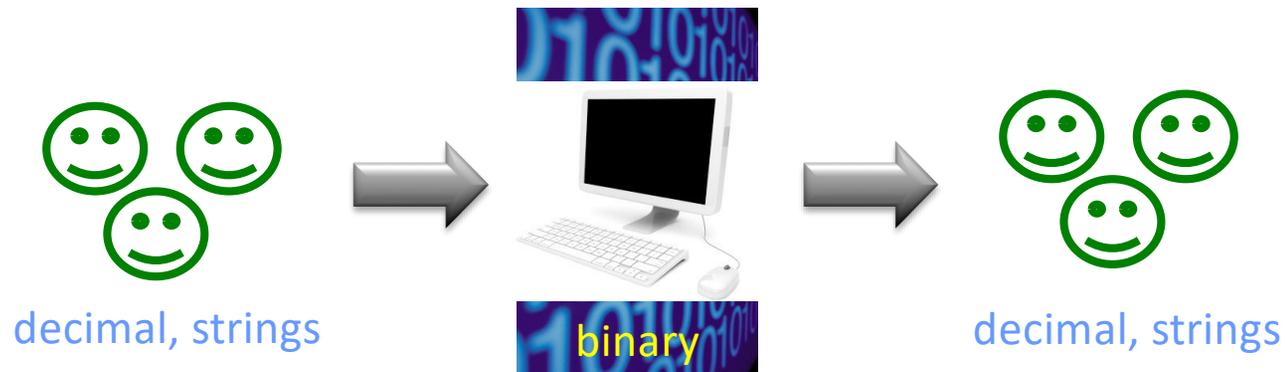
Review: Escape Sequences

- Escape sequences: Represent special characters within a string
  - ➤ Considered a single character

- Escape character: \

  - ➤ The character following the escape character tells you how to interpret the escape sequence

Escape Sequence	Meaning
\n	Newline character (carriage return)
\t	Tab
\" or \'	Quote
\\	Backslash

Review: Representations of Data

- Computer needs 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

decimal, strings   binary   decimal, strings

Seems like a divergence on strings but just wait…

FORMATTING STRINGS

Usability

- Users want output that is easy to read and understand

- Achieving that in a variety of circumstances requires a lot of customizability

Solution: format Method

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

Solution: `format` Method

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

Formatting Strings: `format` Method

- **`templatestring`** is a template for the resulting string with format specifiers instead of the values

  ➤ Each format specifier in `templatestring` requires a corresponding ***replacement value***

  `"{:.2f}".format(3.14159)`   Evaluates to `"3.14"`

  One format specifier
  in template string

  Corresponding replacement value

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

Format Specifiers (Simplified)

[ ] means 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	integer
f	float
e	floating point with exponent

Format Specifiers

[ ] mean optional

`conversion` options
`:[flags][width][.precision][code]`

- flags:

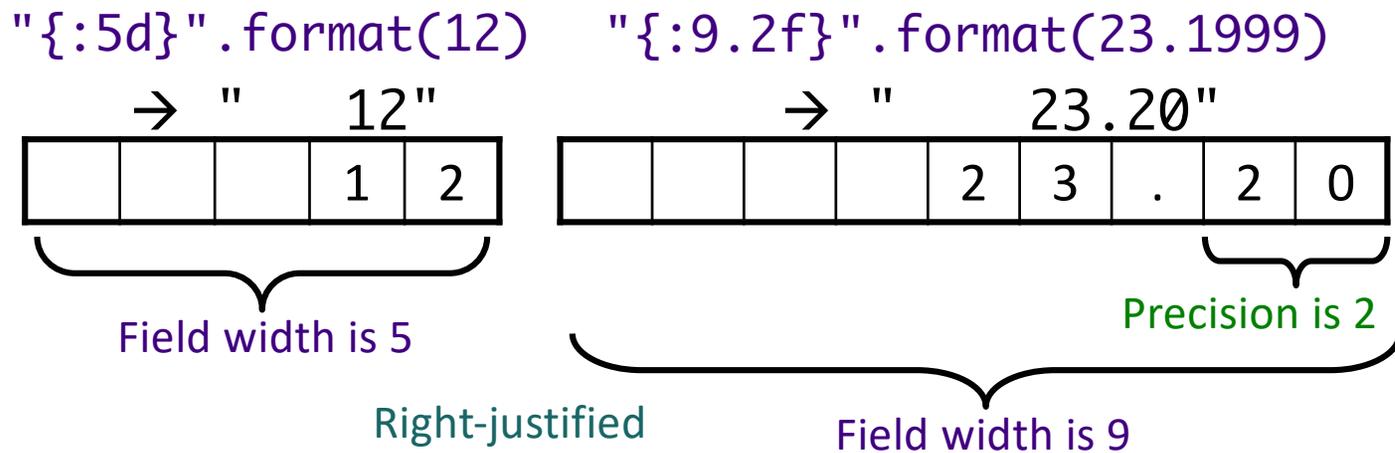| Flag	Meaning
0	Zero fill to width
+	Adds a + sign before positive values
<	Left justify (default for strings)
>	Right justify (default for numbers)
^	center

- **width**:
  - *Minimum* number of character spaces reserved to display the entire value
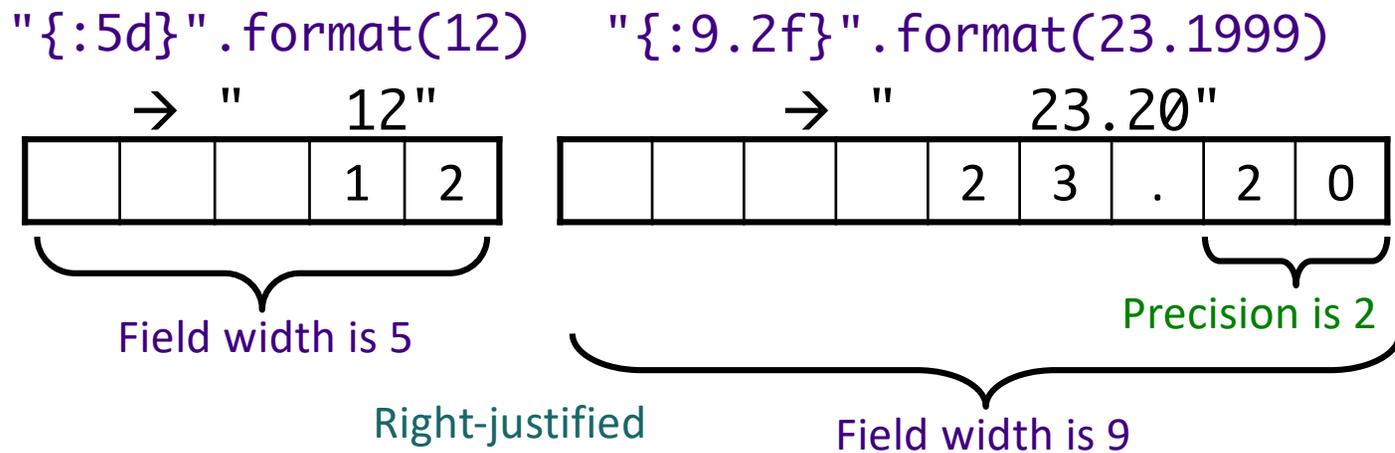 - Includes decimal point, digits before and after the decimal point and the sign

- **precision**:
  - Number of digits after the decimal point for **floating point** values

Example Format Specifiers

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

→ `"   12"`   → `"    23.20"`

			1	2

Field width is 5

Right-justified

				2	3	.	2	0

Precision is 2

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?

Sprenkle - CSCI111 Any guesses? Try out in interpreter.

Example Format Specifiers

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

→ `"   12"`   → `"    23.20"`

Field width is 5

Right-justified

Precision is 2

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

Formatting Practice

What is the resulting string?

```
x = 10
y = 3.5
z = "apple"
```

1. `"{:6d}".format(x)`
2. `"{:6.2f}".format(y)`
3. `"{:06.2f}".format(y)`
4. `"{:6.2f}".format(x)`
5. `"{:^11s}".format(z)`
6. `"{:5d} {:<7.3f}".format(x, y)`

Using `format` Method in `print`

- You often want to format data within a broader context.

- Example: printing out money values
  - ➤ How do you want that data formatted?

Using `format` Method in `print`

- Printing money values

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))
```

Using `format` Method in `print`

- Printing money values

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))
```

How is this different from using the round function?

String Formatting

- There is a lot more you can do with string formatting
  - I presented a subset of the most commonly used functionality
- When formatting strings, consider

  - What is the data type of your data?
    - If a float, how many decimal places do you want?
  - How wide do you want the data to be?
  - What justification? Zero fill? Other flags?

- The answer to these questions help guide your creation of format specifiers

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?

  ➤ For above, not as simple as using tabs. Why not?

String Formatting Conclusion

- There is a lot more you can do with string formatting
  - This is a subset of the most commonly used functionality

- When formatting strings, create the format specifiers by asking:
  - What is the data's type?
  - How do I want it to look?

BROADER ISSUE: SOCIAL MEDIA COMPANY RESPONSIBILITY

BI Groups

Pod 1	Pod 2	Pod 3	Pod 4	Pod 5
Ben Teague Julia Liliane Wesley	Abrar Berkley Renee Ruoan	Brett Caleb Devin Hudson	Ben Rosen Jaz Juyoung Sam	Brielle Cheng Ilaria Rowen

Broader Issue: Social Media Company Responsibility

- Recap the problems and challenges
  - Make clear which of the optional articles you read
- Discuss the solutions, their domains/source/actors, and their tradeoffs
  - What are the most promising?
- For each position, what are the extremes? Are they reasonable?
- Compare and contrast social media to opioids
  - EFF says, "When harmful speech takes place, it's the speaker that should be held responsible, not the service that hosts the speech."
  - "did Meta intentionally addict its users in a way the law recognizes" and "did those efforts result in the victims' harms in a way the law recognizes."

Looking Ahead

- Pre-Lab 7
  - Repeat sections on simple tables (with escape characters), string methods (which includes the subsection on format method), and character classifications