Objectives

- Escape Sequences
- String Formatting

Big Step Forward

- Reflection: How far have I come in Computer Science?
- A lot of String operations
  - Previously: 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

Pair Programming

- Getting the vocabulary down
  - Reinforcing the knowledge
  - Despite "ugh, I hate explaining"
- Frequent role switch
- Discussions of strategy
- Push each other

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

Escape Sequences

- Escape character: \
- Escape sequences
  - newline character (carriage return) → \n
  - tab → \t
  - quote → \" or \'
  - backslash → \\

Interactive demonstration

- Example:
  - `print("To print a \\, you must use \"\\\\\"")`
    - What does this display?

Practice

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

escape_sequence.py

FORMATTING STRINGS

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

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

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

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

Update handout for integer

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

Default if code isn't given

(There are more...)

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

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))
```


Example Format Specifiers

`"{:5d}".format(12)` → `"   12"`

			1	2

Field width is 5

`"{:9.2f}".format(23.1999)` → `"    23.20"`

				2	3	.	2	0

Precision is 2

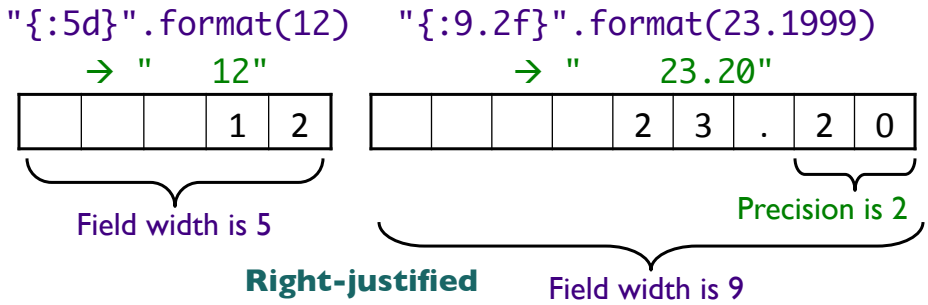
Field width is 9

Right-justified

- 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.

Example Format Specifiers



- 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

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

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?

String Formatting Note

- There are a lot more things you can do with String formatting
- Presenting just a subset of the most commonly used functionality

Looking Ahead

- Lab 6 due Friday