



Introduction to Programming

Spring 2022

Sequences: Strings, Lists, and Files


- The String Data Type
- Simple String Processing
- List as Sequences
- String Representation and Message Encoding
- String Methods
- Lists Have Methods too
- From Encoding to Encryption
- **Input/Output as String Manipulation**
- File Processing

Input/Output as String Manipulation

- Often we will need to do some string operations to prepare our string data for output (“pretty it up”)
- Let’s say we want to enter a date in the format “05/24/2015” and output “May 24, 2015.” How could we do that?

Input/Output as String Manipulation

- Algorithm to format a date:
- Python Code
- Split into month, day, and year
- Convert month string into a number
- Using eval to convert month string into a number
- We can use the int function to convert month string into a number



Input/Output as String Manipulation

- Sometimes we want to convert a number into a string.
- We can use the str function.

Input/Output as String Manipulation

- We now have a complete set of type conversion operations:

Function	Meaning
float(<expr>)	Convert expr to a floating point value
int(<expr>)	Convert expr to an integer value
str(<expr>)	Return a string representation of expr
eval(<string>)	Evaluate string as an expression

String Formatting

- String formatting is an easy way to get beautiful output!
- Example – Change Counter
- `<template-string>.format(<values>)`
- `{ }` within the template-string mark “slots” into which the values are inserted.
- Each slot has description that includes format specifier telling Python how the value for the slot should appear.

String Formatting

- Form of description:
`<index> : <format-specifier>`
- Index tells which parameter to insert into the slot.
- The formatting specifier has the form:
`<width>.<precision><type>`
- `f` means "fixed point" number
- `<width>` tells us how many spaces to use to display the value. 0 means to use as much space as necessary.
- `<precision>` is the number of decimal places.

String Formatting

- Numeric values are right-justified and strings are left-justified, by default.
- You can also specify a justification before the width.

Class Work

• Show the string that would result from each of the following formatting operations:

- a) `"Looks like {1} and {0} for breakfast".format("eggs", "spam")`
- b) `"There is {0} {1} {2} {3}".format(1, "spam", 4, "you")`
- c) `"Hello {0}".format("Susan", "Computewell")`
- d) `"{0:0.2f} {0:0.2f}".format(2.3, 2.3468)`
- e) `"{7.5f} {7.5f}".format(2.3, 2.3468)`
- f) `"Time left {0:02}:{1:05.2f}".format(1, 37.374)`
- g) `"{1:3}".format("14")`

Class Work

- Download the file `futval.py` from Canvas. This program prompts the user for the amount of the investment, the annualized interest rate, and the number of years of investment.
- Run the program.
- Fix the print statements so that output is nicely formatted in the form of a table.