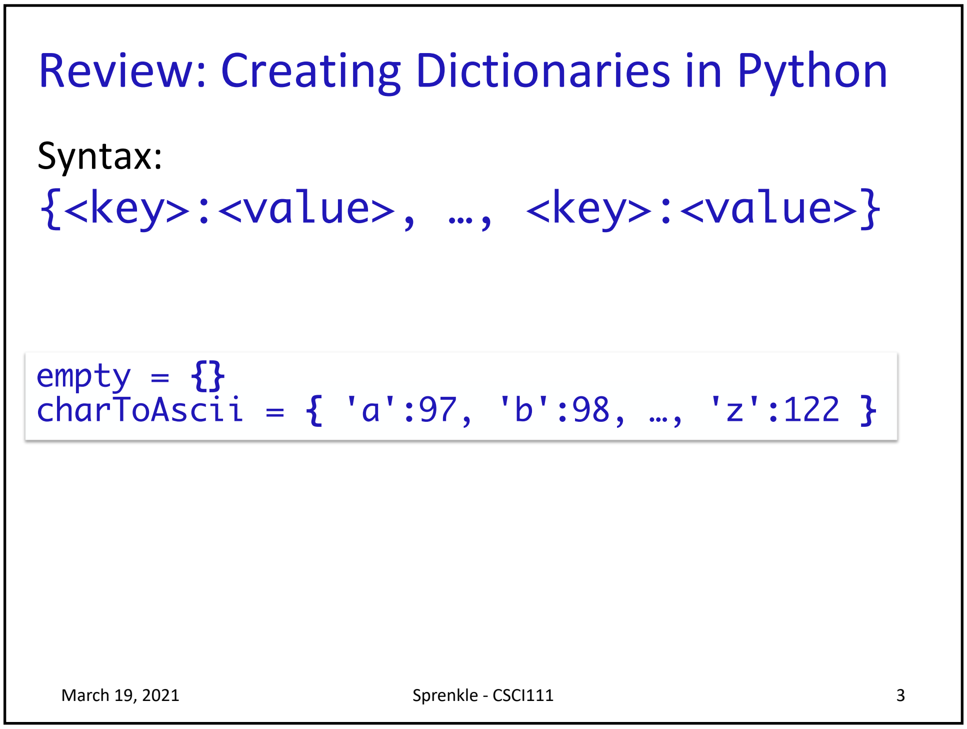
Objectives

- Continuing with dictionaries
- Exceptions
- Broader Issue: Application Data

Review: Dictionaries

- What is a dictionary in Python?
- What is the syntax for creating a new dictionary?
- How do we access a key’s value from a dictionary? (2 ways)
  - What happens if there is no mapping for that key?
- How do we create a key → value mapping in a dictionary?
- How can we iterate through a dictionary?

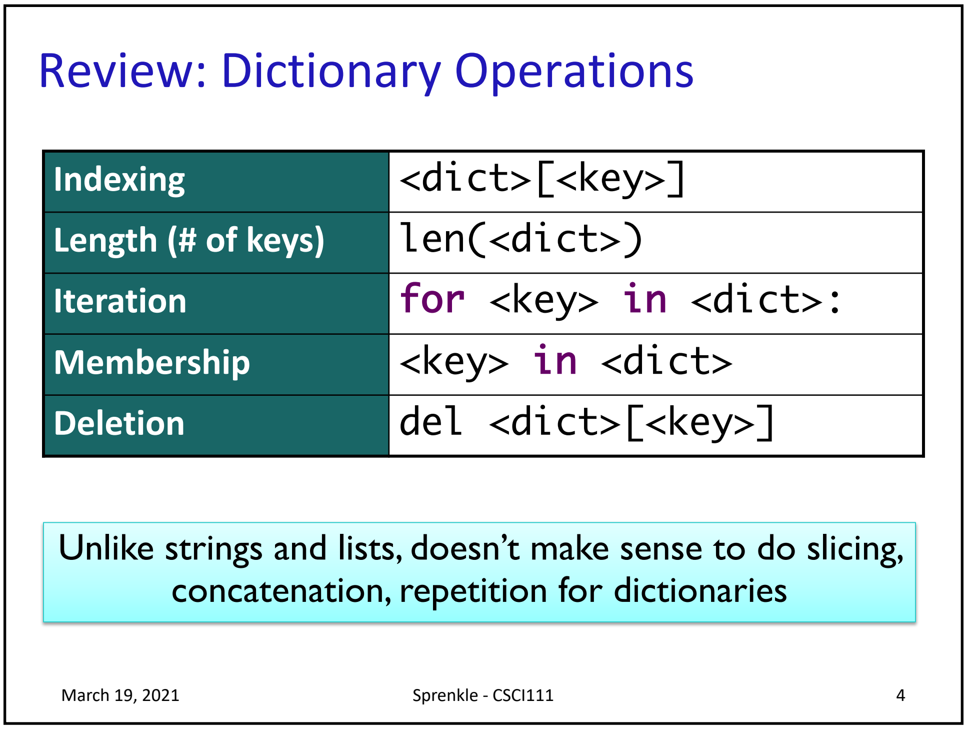


Review: Creating Dictionaries in Python

Syntax:

```
{<key>:<value>, …, <key>:<value>}
```

```
empty = {}
charToAscii = { 'a':97, 'b':98, …, 'z':122 }
```



Review: Dictionary Operations

Indexing	`<dict>[<key>]`
Length (# of keys)	`len(<dict>)`
Iteration	`for <key> in <dict>:`
Membership	`<key> in <dict>`
Deletion	`del <dict>[<key>]`

Unlike strings and lists, doesn't make sense to do slicing, concatenation, repetition for dictionaries

Review: Dictionary Methods

Method Name	Functionality
`<dict>.clear()`	Remove all items from dictionary
`<dict>.keys()`	Returns a copy of dictionary's keys (a set-like object)
`<dict>.values()`	Returns a copy of dictionary's values (a set-like object)
`<dict>.get(x [, default])`	Returns `<dict>[x]` if `x` is a key; Otherwise, returns `None` (or default value)

Review: Accessing Values Using Indexing

- Syntax:
  `<dictionary>[<key>]`
- Examples:

```
charToAscii['z']

nameToPhoneNum['friendname']
```

- `KeyError` if key is not in dictionary
  - Runtime error; exits program

Review: Adding/Modifying Key-Value Pairs

- Syntax:

  `<dictionary>[<key>] = <value>`

- `nameToPhoneNum['registrar'] = 8455`
  - ➢ Adds mapping for `'registrar'` to 8455

  OR

  - ➢ If mapping already existed, modifies old mapping to 8455

Review: Problem

`years_dictionary.py`

- Part 1:
  - ➢ Given a file of the form
    - <firstname> <gradyear>
  - ➢ Goal: Quickly find out what a student’s grad year is
    - How do we want to model the data?
    - What is the key? What is the value?
- Part 2:
  - ➢ Prompt user for the first name of the student
  - ➢ Display the student’s graduation year

Part 3: Repeat Part 2

Review: Algorithm to Problem

- Create an empty dictionary
- Read in the file line by line
  - Split the line
  - From the split, get the name and the year
  - Add a mapping of the name to the year in the dictionary
    - (accumulate the data in the dictionary)
- Process the data in the dictionary, e.g.,
  - Display it, in sorted order
  - Get user input to get answers

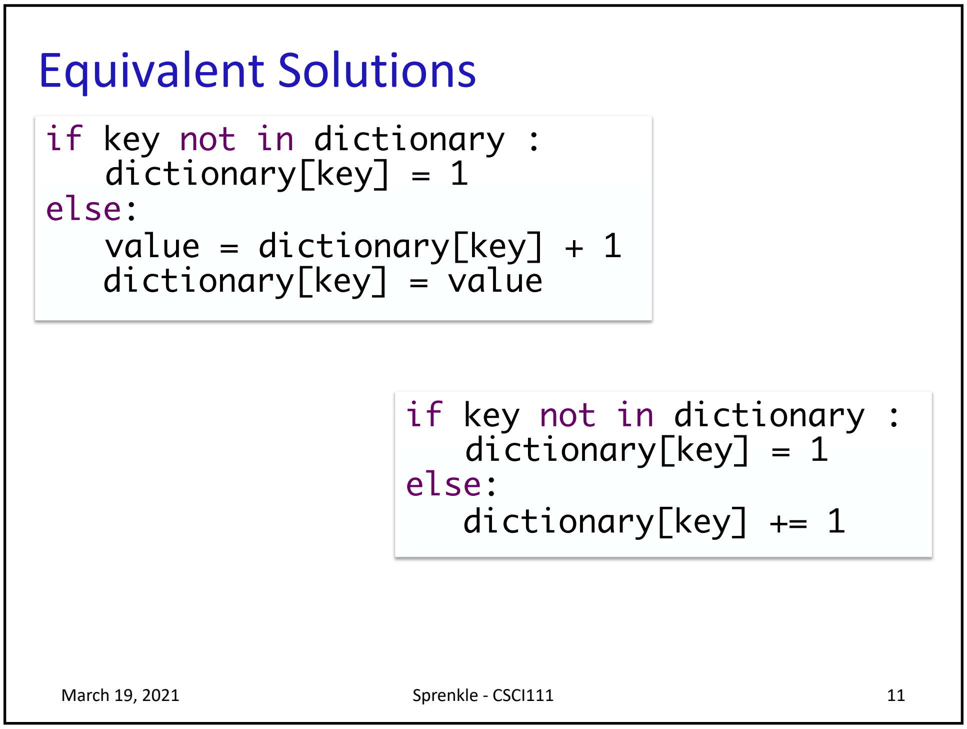
Problem

Example file:

```
Person1 2023
Person2 2021
Person3 2023
Person4 2024
Person5 2023
…
```

- Given a file of the form
  - <firstname> <classyear>
- Goal: Report the *number* of students in each graduation year
  - How do we want to model the data?
  - What is the key? What is the value?
- Problem-solving Approach:
  - Pretend you are the computer, how would you solve this problem?

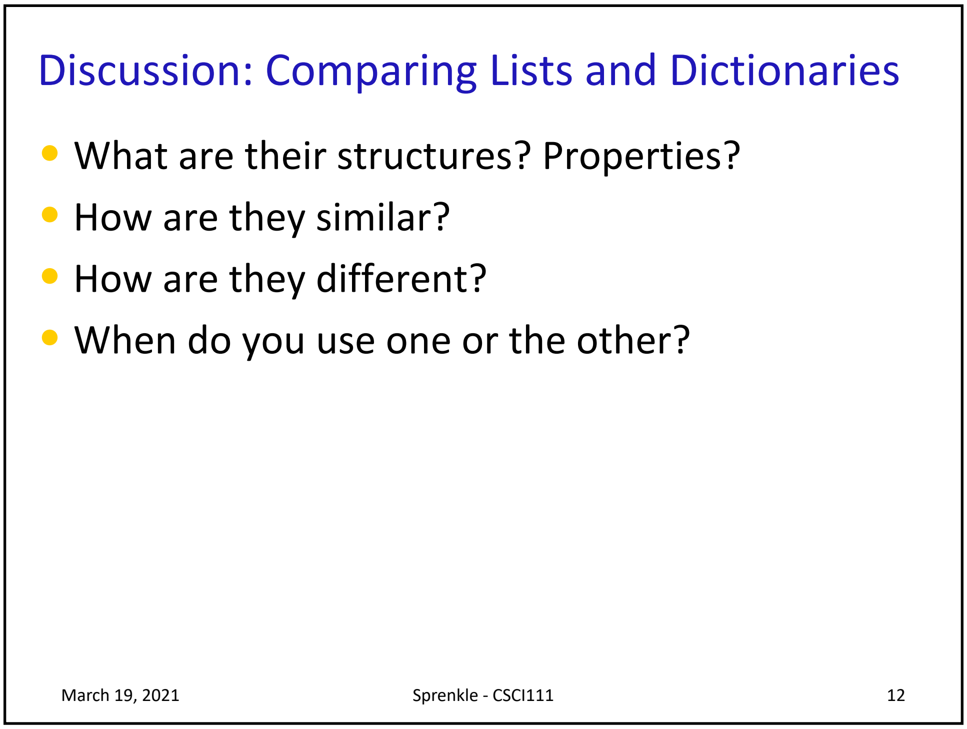
`years_dictionary2.py`



Equivalent Solutions

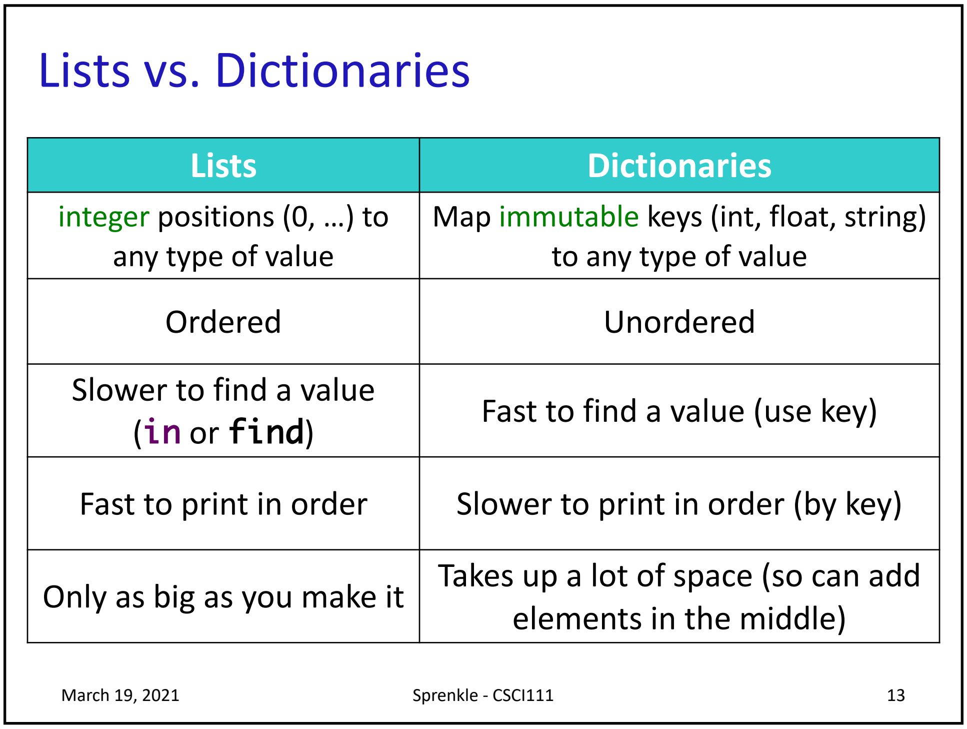
```
if key not in dictionary :
   dictionary[key] = 1
else:
   value = dictionary[key] + 1
   dictionary[key] = value
```

```
if key not in dictionary :
   dictionary[key] = 1
else:
   dictionary[key] += 1
```



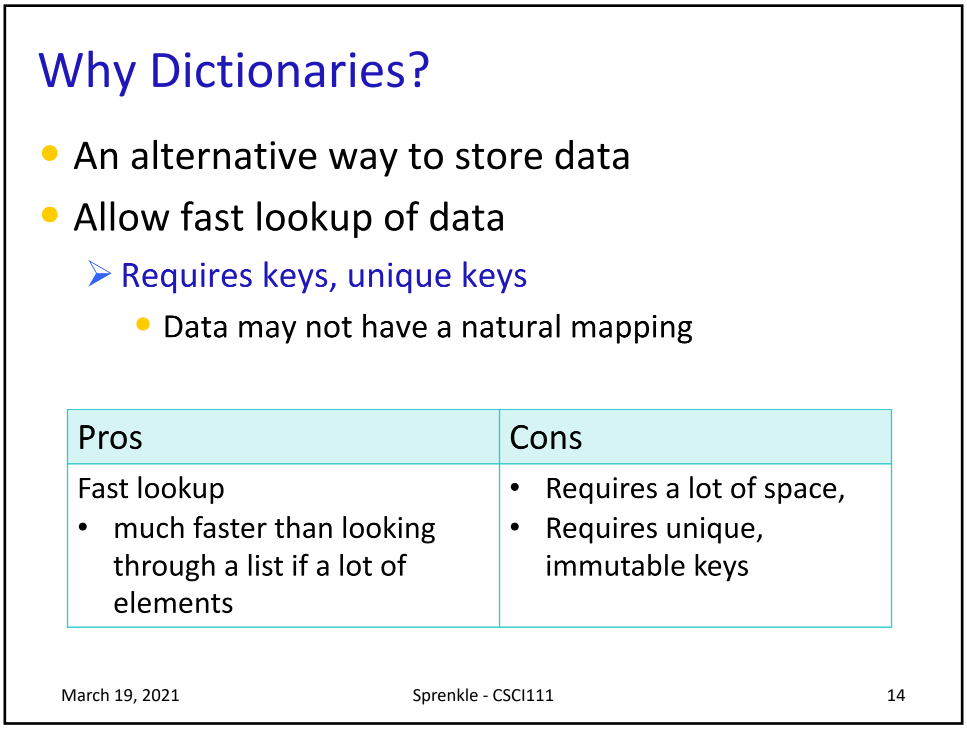
Discussion: Comparing Lists and Dictionaries

- What are their structures? Properties?
- How are they similar?
- How are they different?
- When do you use one or the other?



Lists vs. Dictionaries

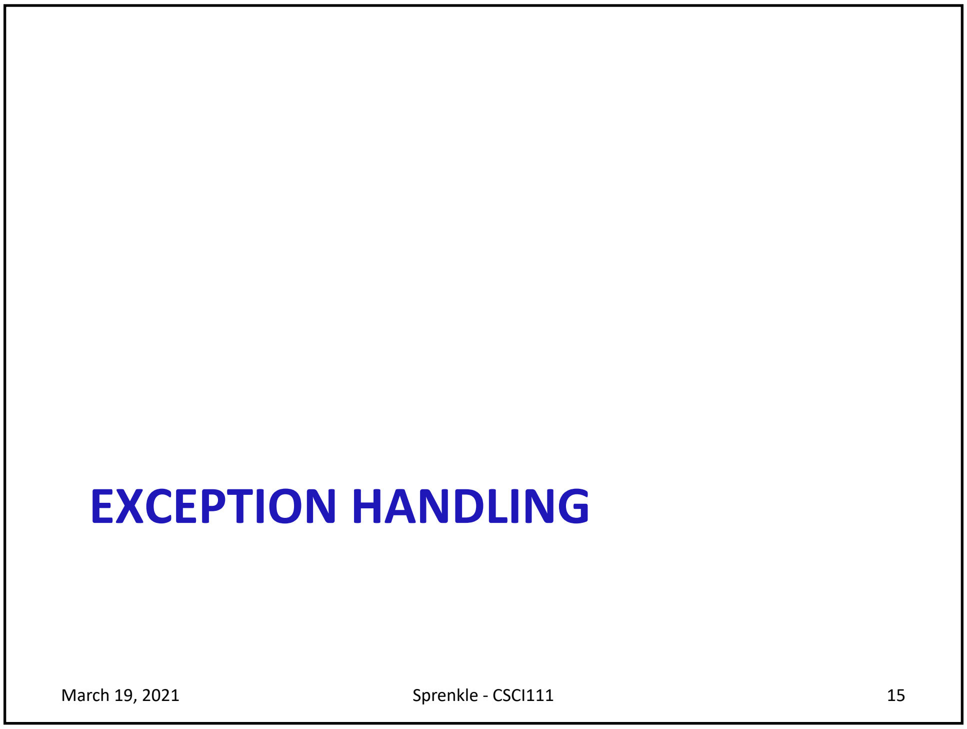
Lists	Dictionaries
integer positions (0, …) to any type of value	Map immutable keys (int, float, string) to any type of value
Ordered	Unordered
Slower to find a value (`in` or `find`)	Fast to find a value (use key)
Fast to print in order	Slower to print in order (by key)
Only as big as you make it	Takes up a lot of space (so can add elements in the middle)



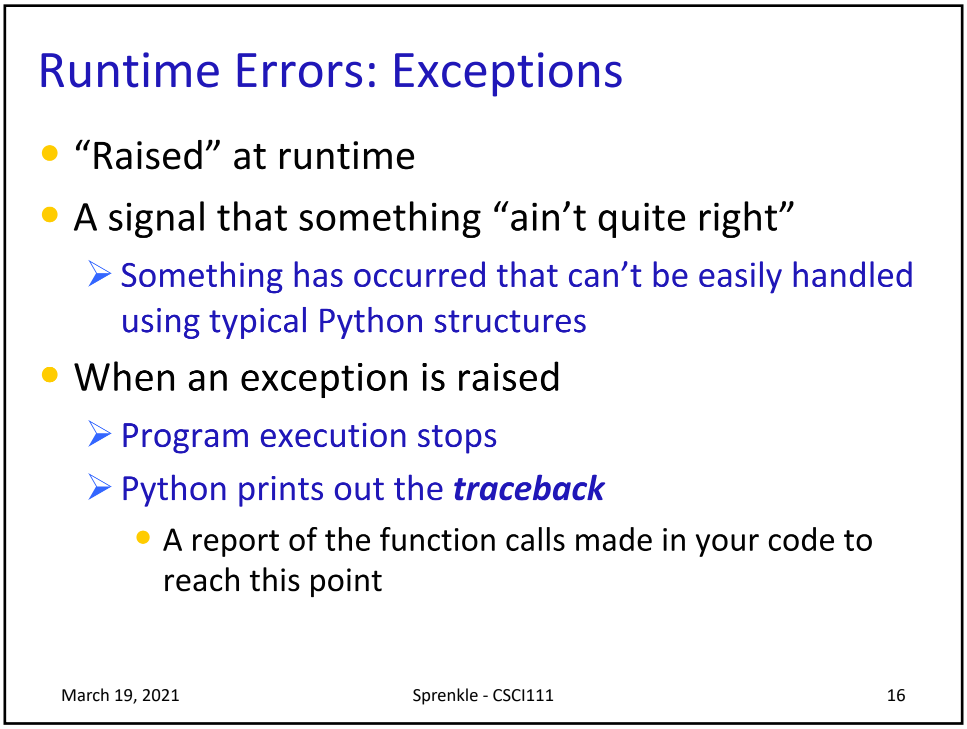
Why Dictionaries?

- An alternative way to store data
- Allow fast lookup of data
  - Requires keys, unique keys
    - Data may not have a natural mapping

Pros	Cons
Fast lookup • much faster than looking through a list if a lot of elements	• Requires a lot of space, • Requires unique, immutable keys

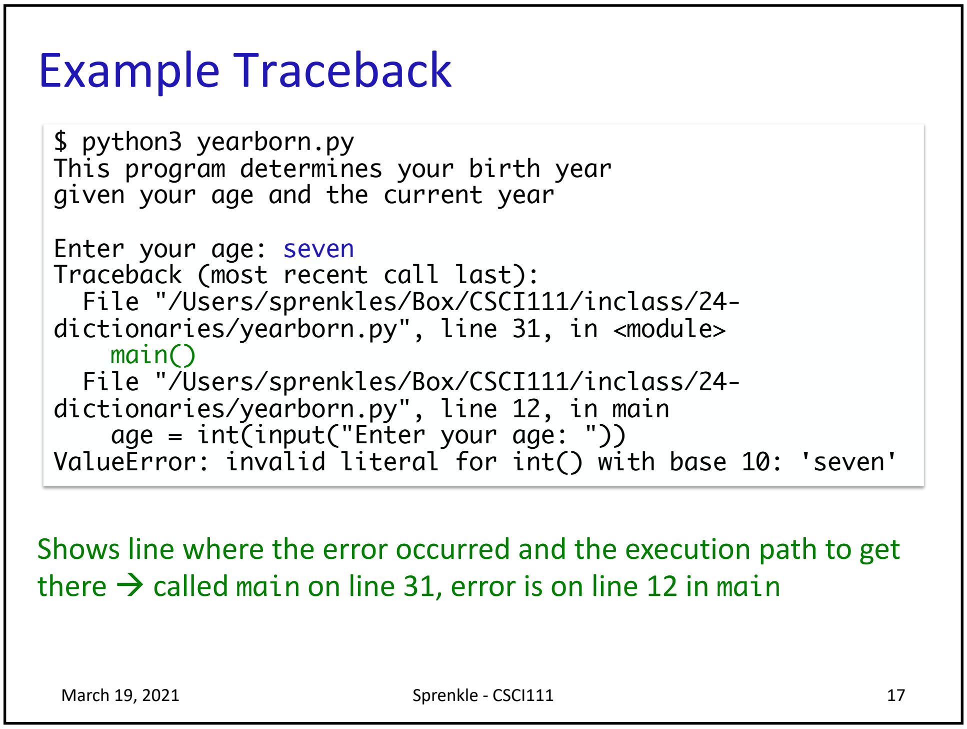


EXCEPTION HANDLING



Runtime Errors: Exceptions

- “Raised” at runtime
- A signal that something “ain’t quite right”
  - Something has occurred that can’t be easily handled using typical Python structures
- When an exception is raised
  - Program execution stops
  - Python prints out the ***traceback***
    - A report of the function calls made in your code to reach this point

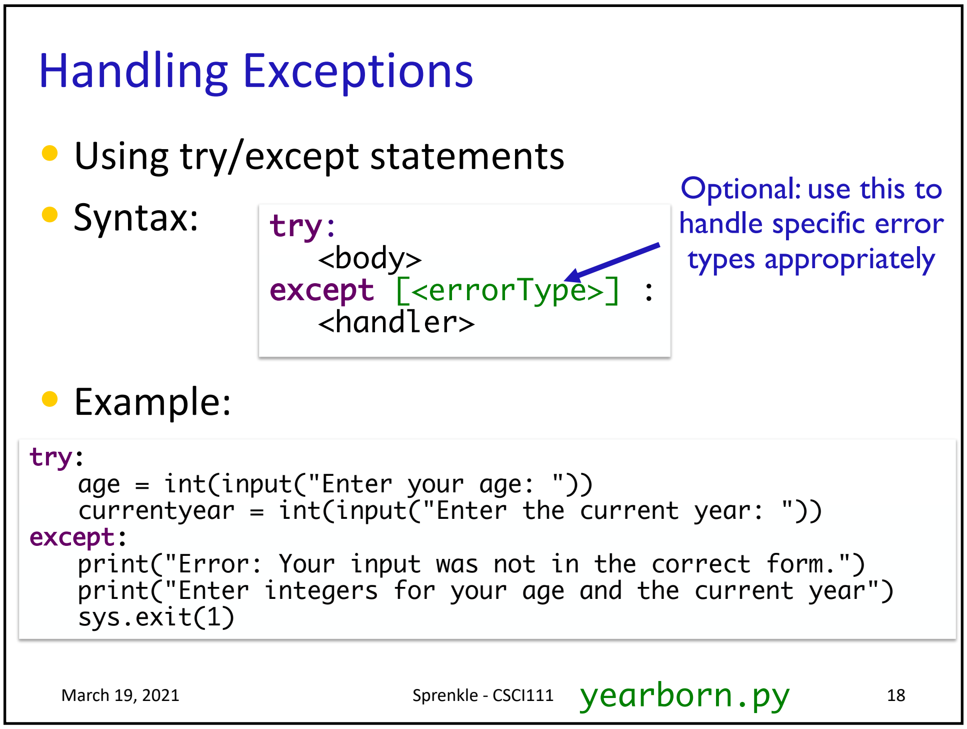


Example Traceback

```
$ python3 yearborn.py
This program determines your birth year
given your age and the current year

Enter your age: seven
Traceback (most recent call last):
  File "/Users/sprenkles/Box/CSCI111/inclass/24-
dictionaries/yearborn.py", line 31, in <module>
    main()
  File "/Users/sprenkles/Box/CSCI111/inclass/24-
dictionaries/yearborn.py", line 12, in main
    age = int(input("Enter your age: "))
ValueError: invalid literal for int() with base 10: 'seven'
```

Shows line where the error occurred and the execution path to get there → called `main` on line 31, error is on line 12 in `main`



Handling Exceptions

- Using try/except statements
- Syntax:

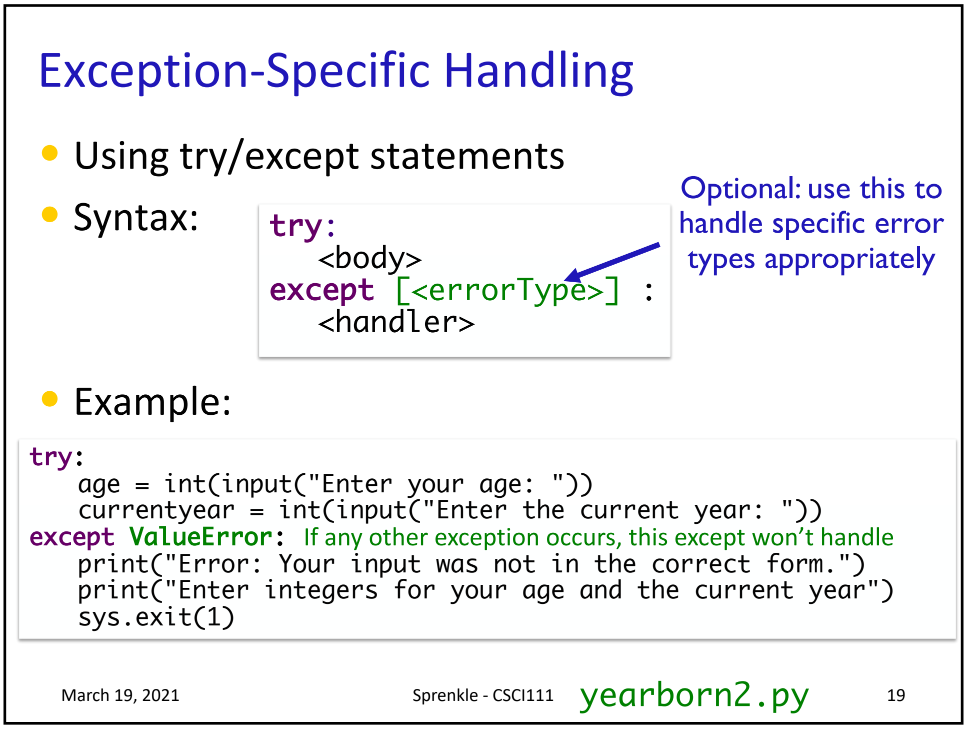
```
try:
    <body>
except [<errorType>] :
    <handler>
```

Optional: use this to handle specific error types appropriately

- Example:

```
try:
    age = int(input("Enter your age: "))
    currentyear = int(input("Enter the current year: "))
except:
    print("Error: Your input was not in the correct form.")
    print("Enter integers for your age and the current year")
    sys.exit(1)
```

yearborn.py



Exception-Specific Handling

- Using try/except statements
- Syntax:

```
try:
   <body>
except [<errorType>] :
   <handler>
```

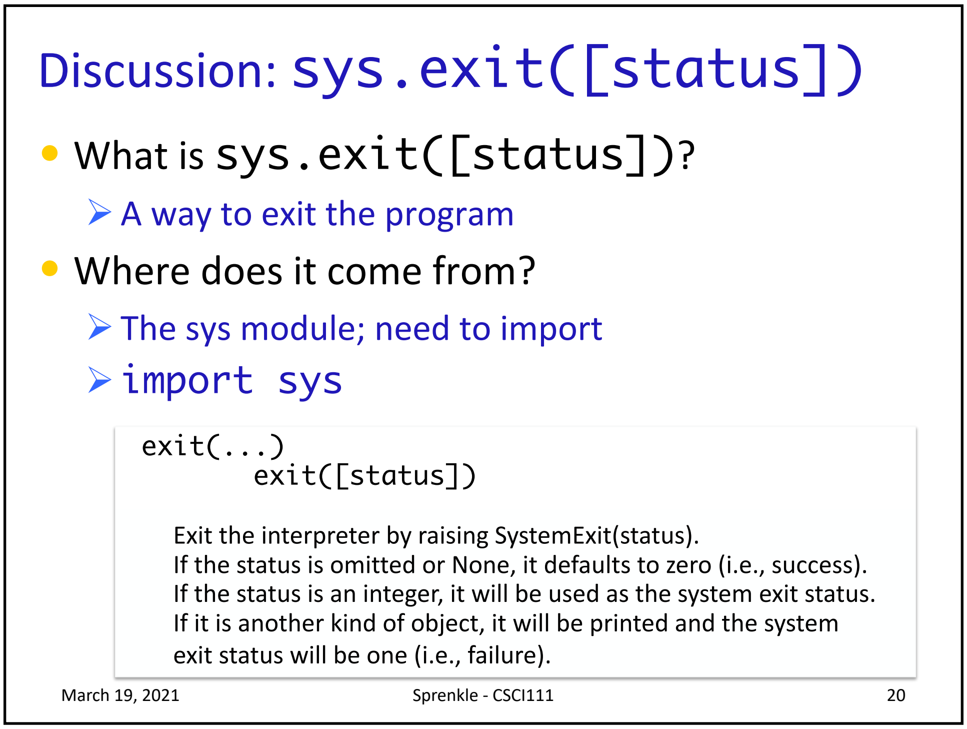
Optional: use this to handle specific error types appropriately

- Example:

```
try:
   age = int(input("Enter your age: "))
   currentyear = int(input("Enter the current year: "))
except ValueError:
   print("Error: Your input was not in the correct form.")
   print("Enter integers for your age and the current year")
   sys.exit(1)
```

If any other exception occurs, this except won't handle

yearborn2.py



Discussion: `sys.exit([status])`

- What is `sys.exit([status])`?
  - A way to exit the program
- Where does it come from?
  - The sys module; need to import
  - `import sys`

```
exit(...)
        exit([status])
```

Exit the interpreter by raising SystemExit(status).
If the status is omitted or None, it defaults to zero (i.e., success).
If the status is an integer, it will be used as the system exit status.
If it is another kind of object, it will be printed and the system exit status will be one (i.e., failure).

Examples of Types of Exceptions

- IndexError
  - When index is not found in the sequence
- KeyError
  - When a key is not found in the dictionary
- IOError:
  - FileNotFoundError: File doesn’t exist
  - PermissionError: Don’t have permission to read/write file

Exception Handling

```
try:
    inFile = open(infileName, "r")
    # normally, would process file here.
    inFile.close()
except IOError as exc :
    print("Error reading \"" + infileName + "\".")
    # could be a variety of different problems,
    # so print out the exception
    print(exc)
    print(type(exc))
    sys.exit(1)
```

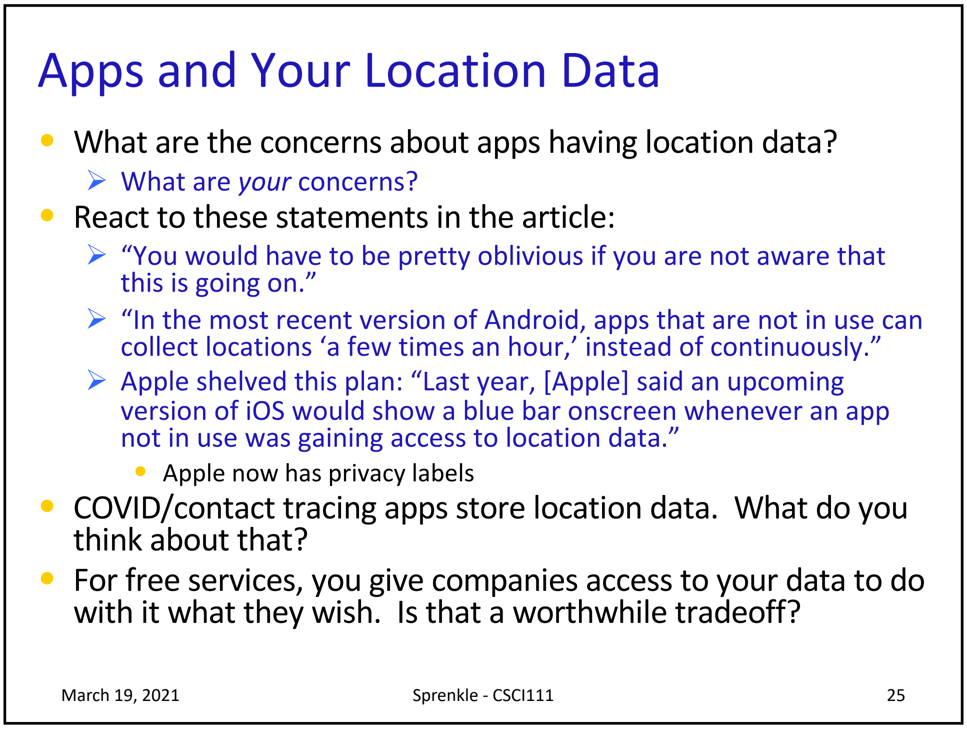
- Exceptions are *objects*
- We can get more information about the exception by printing them out

file_handle.py

Best Practices

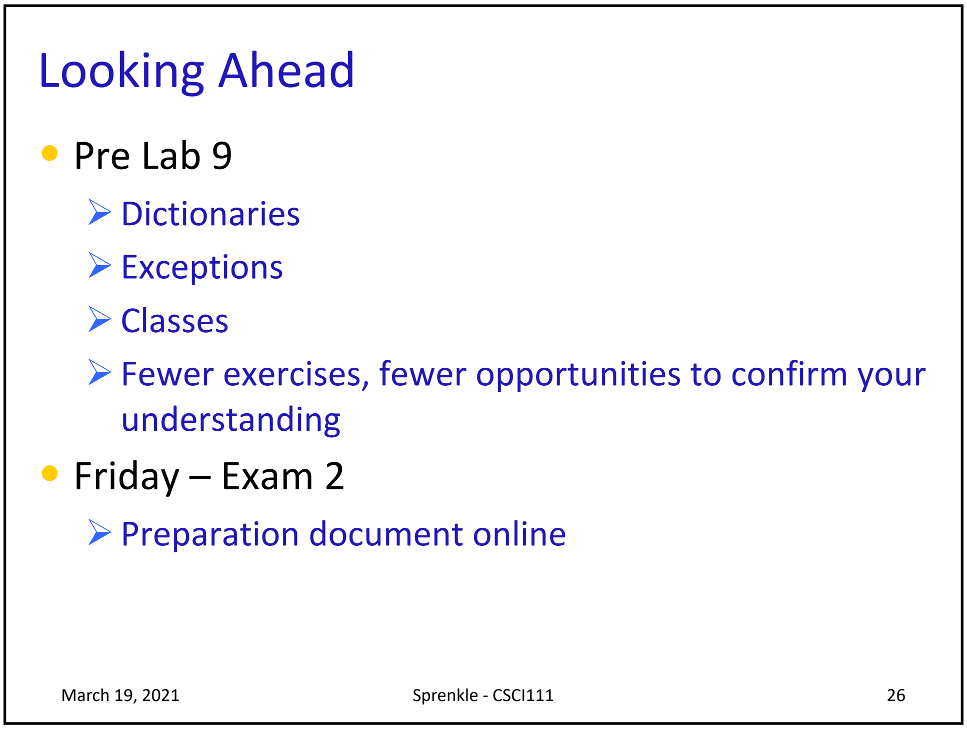
- Prevent errors as best you can
  - Example: use if statements to verify data
- For errors you can't prevent, handle them!
  - Example: We can check if a file exists before trying to read it BUT between the check and actually reading the file, the file could be deleted from the system!

BROADER ISSUE: APPLICATION DATA



Apps and Your Location Data

- What are the concerns about apps having location data?
  - What are *your* concerns?
- React to these statements in the article:
  - “You would have to be pretty oblivious if you are not aware that this is going on.”
  - “In the most recent version of Android, apps that are not in use can collect locations ‘a few times an hour,’ instead of continuously.”
  - Apple shelved this plan: “Last year, [Apple] said an upcoming version of iOS would show a blue bar onscreen whenever an app not in use was gaining access to location data.”
    - Apple now has privacy labels
- COVID/contact tracing apps store location data. What do you think about that?
- For free services, you give companies access to your data to do with it what they wish. Is that a worthwhile tradeoff?



Looking Ahead

- Pre Lab 9
  - Dictionaries
  - Exceptions
  - Classes
  - Fewer exercises, fewer opportunities to confirm your understanding
- Friday – Exam 2
  - Preparation document online