Féidearthachtaí as Cuimse
Infinite Possibilities

Programming for Analytics

Lecture 3: Data Structures, Strings and Errors

Bojan Božić
School of Computer Science
TU Dublin, Grangegorman

bojan.bozic@tudublin.ie

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

Overview

- Lists – ordered, mutable sequences
- Tuples – ordered, immutable sequences
- Sets – unordered, unique items
- Dictionaries – key-value mappings

Lists

- Lists are ordered collections of items: e.g., `[1,2,3]`
- Mutable – can be changed after creation
- Can hold mixed data types
- Useful for storing sequences of data

Common List Operations

- Access with index: `list[0]`
- Append: `list.append(item)`
- Insert: `list.insert(index, item)`
- Remove: `list.remove(item)`, `pop(index)`
- Slicing: `list[start:end]`
- Sorting: `list.sort()`

Activity: Lists

- Create a list of 5 numbers
- Add a number, remove a number, sort the list
- Print the sum and average of the list

Tuples

- Tuples are like lists but immutable: e.g., `(1,2,3)`
- Faster and safer than lists for fixed data
- Useful for returning multiple values from functions
- Tuple unpacking: `a, b = (1,2)`

Activity: Tuples

- Create a tuple with 3 values:

```
name, age, city
```

- Use unpacking to assign to variables
- Print each value

Sets

- Sets are unordered collections of unique items: e.g., `{1,2,3}`
- Duplicates are automatically removed
- Useful for membership tests and removing duplicates
- Set operations: union, intersection, difference, add, remove

Activity: Sets

- Create a set of student IDs
- Add and remove IDs
- Check if an ID is present

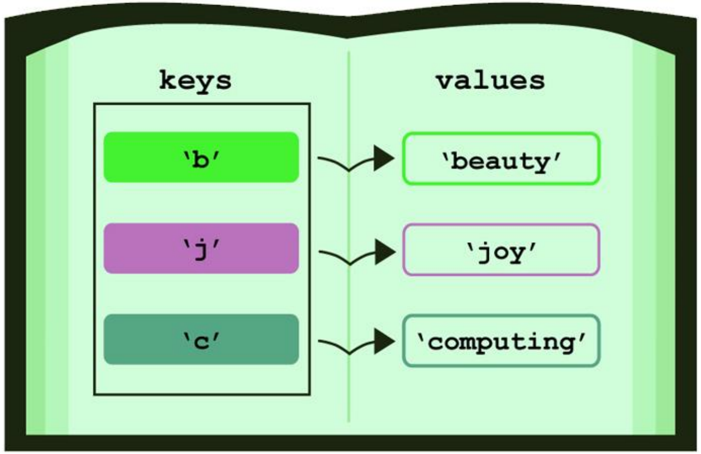
Dictionaries

- Dictionaries store key-value pairs: e.g., `{‘name’: ‘Alice’}`
- Keys must be unique and immutable
- Access: `dict[‘key’]`, Update: `dict[‘key’] = value`
- Remove with `del`
- Loop with:

```
for key, value in dict.items()
```

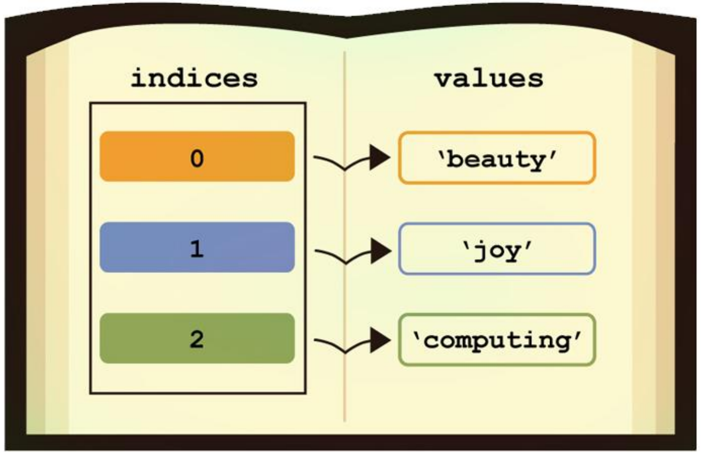
DICTIONARIES IN PYTHON

Dictionaries



keys	values
'b'	'beauty'
'j'	'joy'
'c'	'computing'

Lists



indices	values
0	'beauty'
1	'joy'
2	'computing'

Activity: Dictionaries

- Create a dictionary of students and their grades.
- Update a grade, add a new student, remove one.
- Loop through and print all students and grades.

Mutability and Immutability

- Mutable: Lists, Sets, Dictionaries – can be changed
- Immutable: Tuples, Strings, Numbers – cannot be changed
- Important for function parameters and performance

Built-in Functions

- len(), sum(), min(), max(), sorted()
- type(), list(), tuple(), set(), dict()
- Useful for working generically with containers

List Comprehensions

- Compact way to create lists:

```
[x for x in range(5)]
```

- With condition:

```
[x for x in range(10) if x % 2 == 0]
```

- Readability and performance benefit

Activity: List Comprehensions

- Generate a list of squares 1-10
- Filter for even numbers only
- Bonus: lowercase a list of strings

Strings in Python

- Strings are sequences of characters
- Immutable – cannot be changed in place
- Support indexing, slicing, and iteration

Common String Methods

- s.upper(), s.lower(), s.strip()
- s.replace(), s.split(), s.find()
- s.startswith(), s.endswith()

String Formatting

- F-strings: f”Hello, {name}”
- “Hello, {}”.format(name)
- “Hello %s” % name

Activity: String Cleanup

- `raw_name = “  Alice\n”`
- Strip whitespace, convert to uppercase
- Output: “WELCOME, ALICE”

Error Handling in Python

- Try-except blocks handle runtime errors
- Prevents program from crashing
- Optional: else and finally blocks

Common Exceptions

- ValueError: invalid type conversion
- ZeroDivisionError: dividing by 0
- IndexError: accessing out-of-bounds index
- KeyError: missing dictionary key

Try-Except Example

```
try:
    x = int(input())
except ValueError:
    print(“Invalid input”)
finally:
    print(“Done”)
```

Activity: Safe Division

- Ask user for two numbers
- Try to divide them, catch ValueError and ZeroDivisionError
- Print helpful messages

Recap

- Explored core data structures: list, tuple, set, dict
- Covered mutability and common methods
- String methods and formatting
- Basic error handling with try-except

Next Week

- File handling in Python
- Reading and writing CSV and JSON files
- Introduction to data persistance

Questions?