JANUARY 27TH, 2025

Collections

CE 311K - L07

Review: Lists

Python *list* is a general-purpose collection

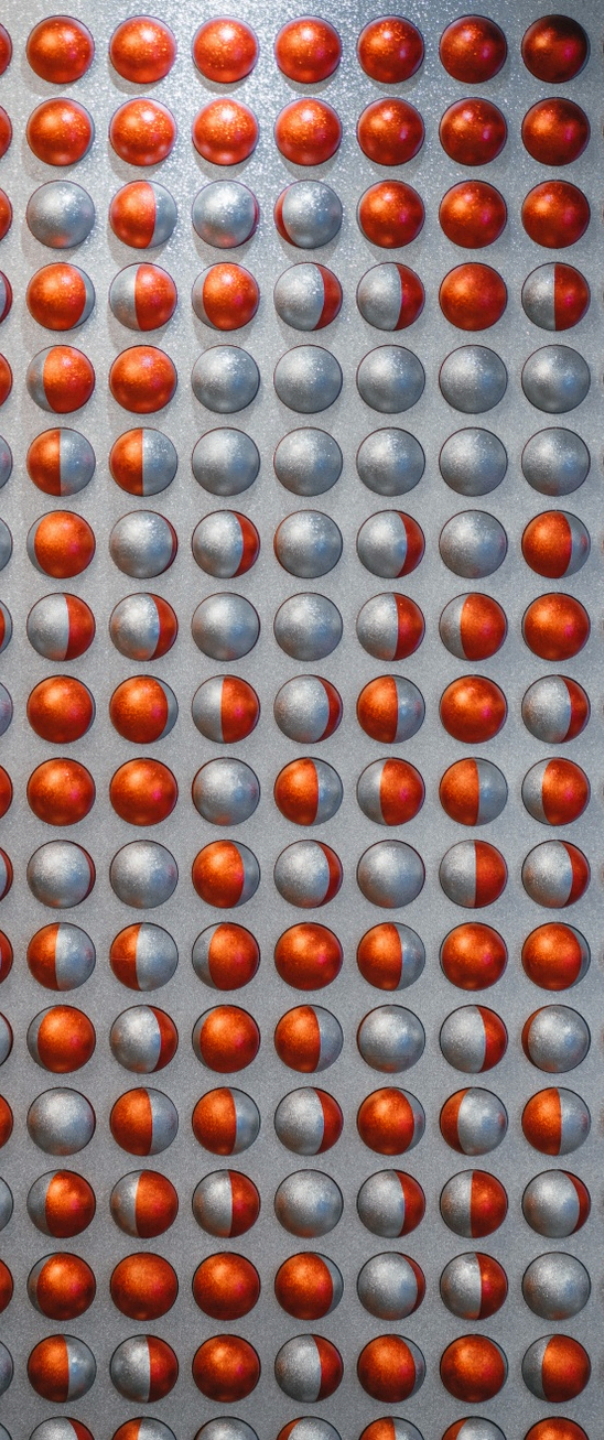
```
numbers = [1, 2, 3, 4, 5]
fruits = ["apple", "banana", "cherry"]
mixed = [1, "apple", True]
```

Ordered and mutable

Dynamic; can grow and shrink in size

Zero-indexed so you can access elements or slices

Versatile, built-in methods for even more flexibility



Sets

A *set* is an **unordered** collection of **unique** events

Sets do not allow duplicate values nor preserve order

```
set_1 = {1, 2, 3, 4}
set_2 = set([2, 3, 4, 5])  # Cast from a list
```

Sets are ideal for tasks that require uniqueness and membership

```
print(set_1 | set_2) # Output: {1, 2, 3, 4, 5}
print(set_1 & set_2) # Output: {2, 3, 4}
print(set_1 - set_2) # Output: {1}
```

Tuples

A *tuple* is an **ordered, immutable** collection of items

Tuples are typically used to represent fixed collections of related items

```
location = (30.2672, 97.7431)
rgb_color = tuple([191, 87, 0]) # Cast from list
```

You can access, slice, and perform operations on *tuples* just like *lists*

Tuples ensure data is not accidently modified

Tuples use less memory, making them useful for read-only data


FIND THE BUG

Dictionaries

A *dict* is an **unordered** collection of **key-value** pairs

```
{
    "name": "Hagen Fritz",
    "id": 12345,
    "is_student": False
}
```

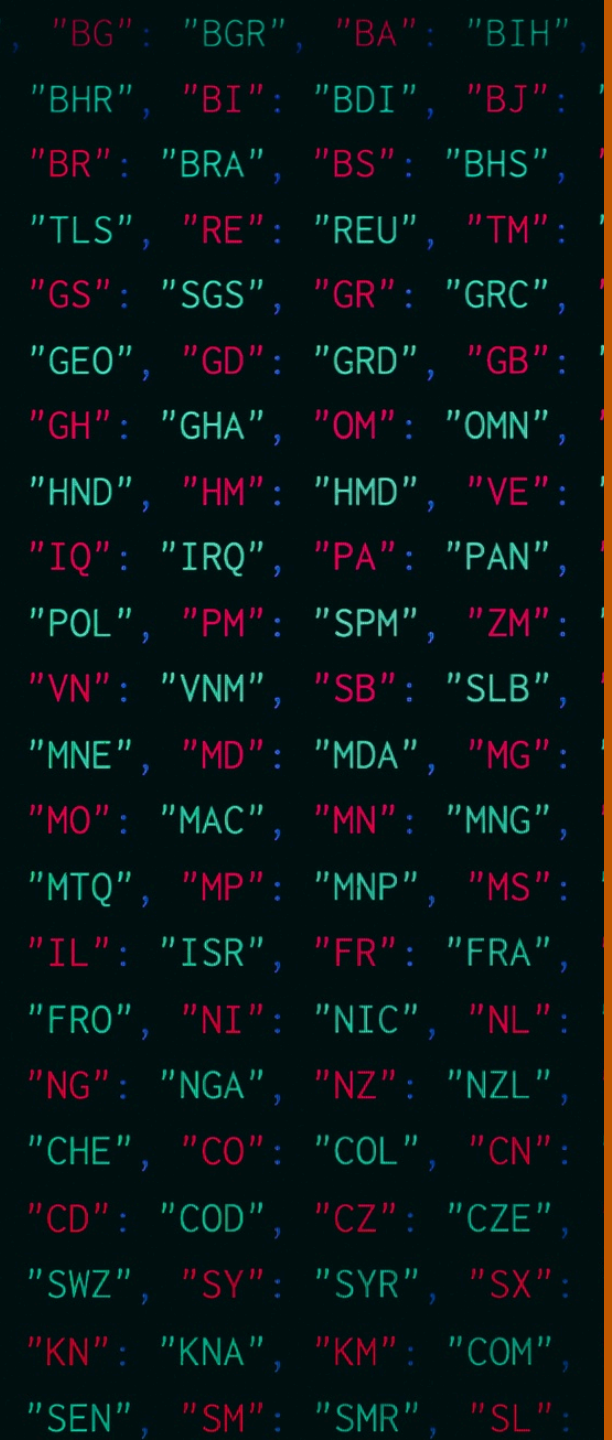
Create using curly braces or the *dict* constructor

```
person = {"name": "Hagen", "age": 31}
person = dict(name="Hagen", age=31)
```

Nested Dictionaries

Dictionaries can have multiple, nested levels

```
{
    "name": "Hagen Fritz",
    "id": 12345,
    "is_student": False,
    "employment": {
        "University of Texas at Austin": {
            "job_title": "Lecturer",
            "start_year": 2025
        },
        "Rogers-O'Brien Construction": {
            "job_title": "Software Engineer",
            "start_year": 2022
        }
    }
}
```



Why Dictionaries?

Great for organizing data where you need to access values by **keys** rather than indices

```
person = {"name": "Hagen", "age": 31}
print(person["name"]) # Output: "Hagen"
```

A *dict* is very similar to **JSON** (JavaScript Object Notation)

Commonly used for storing and exchanging data in web applications

Summary

A *list* is an **ordered**, **mutable** collection of items

Uses brackets: []

A *tuple* is an **ordered**, **immutable** collection of items

Elements are enclosed in parenthesis: ()

A *set* is an **unordered** collection of **unique** events

Uses curly brackets with single elements: {}

A *dict* is an **unordered** collection of **key-value pairs**

Create with curly brackets and key-value pairs separated by a colon