# [301] JSON

Tyler Caraza-Harter

# Learning Objectives Today

JSON

- differences with Python syntax
- creating JSON files
- reading JSON files

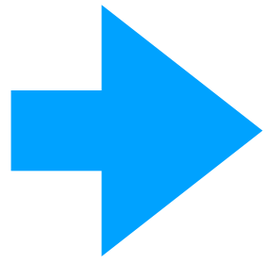
**Read: Sweigart Ch 14**
**https://automatetheboringstuff.com/chapter14/**

**“JSON and APIs” to the end**

# Python Data Structures and File Formats

**Python**

```
[
 [“name”, “x”, “y”],
 [“alice”, 100, 150],
 [“bob”, -10, 80]
]
```



**File**

?

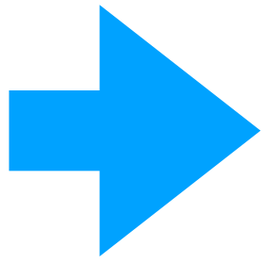
**list of lists**

# Python Data Structures and File Formats

**Python**

```
[
 [“name”, “x”, “y”],
 [“alice”, 100, 150],
 [“bob”, -10, 80]
]
```

**list of lists**



**File**

```
name,x,y
alice,100,150
bob,-10,80
```

**CSV file**

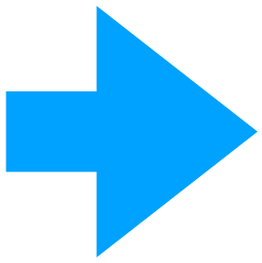
**We can use CSV files to store data we would want in lists of lists**

# Python Data Structures and File Formats

**Python**

```
[
 [“name”, “x”, “y”],
 [“alice”, 100, 150],
 [“bob”, -10, 80]
]
```

**list of lists**



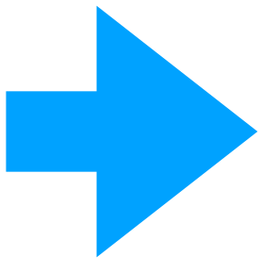
**File**

```
name,x,y
alice,100,150
bob,-10,80
```

**CSV file**

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

**dict of dicts**



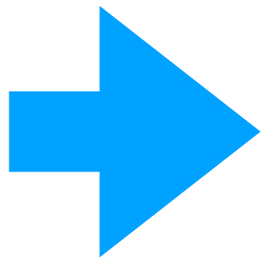
?

# Python Data Structures and File Formats

**Python**

```
[
 [“name”, “x”, “y”],
 [“alice”, 100, 150],
 [“bob”, -10, 80]
]
```

**list of lists**



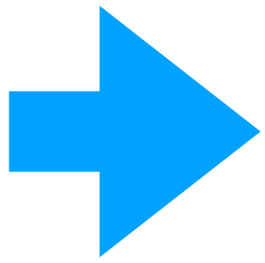
**File**

```
name,x,y
alice,100,150
bob,-10,80
```

**CSV file**

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

**dict of dicts**



```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

**JSON file**

# Python Data Structures and File Formats

Python

File

**JSON files look almost identical to Python code for data structures!**

```
[
  [“name”, “x”, “y”],
  [“alice”, 100, 150],
  [“bob”, -10, 80]
]
```

list of lists

```
name,x,y
alice,100,150
bob,-10,80
```

CSV file

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

dict of dicts

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

**JSON file**

# Python Data Structures and File Formats

Python

File

**JSON files look almost identical to Python code for data structures!**

```
[
  [“name”, “x”, “y”],
  [“alice”, 100, 150],
  [“bob”, -10, 80]
]
```

list of lists

```
name,x,y
alice,100,150
bob,-10,80
```

CSV file

**dicts use curly braces**

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

dict of dicts

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

**JSON file**

# Python Data Structures and File Formats

Python

```
[
  [“name”, “x”, “y”],
  [“alice”, 100, 150],
  [“bob”, -10, 80]
]
```

list of lists

JSON files look almost identical to Python code for data structures!

keys are separated from values with a colon

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

dict of dicts

File

```
name,x,y
alice,100,150
bob,-10,80
```

CSV file

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

JSON file

# Python Data Structures and File Formats

Python

File

**JSON files look almost identical to Python code for data structures!**

```
[
  [“name”, “x”, “y”],
  [“alice”, 100, 150],
  [“bob”, -10, 80]
]
```

list of lists

```
name,x,y
alice,100,150
bob,-10,80
```

CSV file

**lists use square brackets**

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

dict of dicts

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

**JSON file**

# Python Data Structures and File Formats

Python

File

**JSON files look almost identical to Python code for data structures!**

```
[
  [“name”, “x”, “y”],
  [“alice”, 100, 150],
  [“bob”, -10, 80]
]
```

list of lists

```
name,x,y
alice,100,150
bob,-10,80
```

CSV file

**strings are in quotes**

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

dict of dicts

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

JSON file

# Python Data Structures and File Formats

Python

File

**JSON files look almost identical to Python code for data structures!**

```
[
  [“name”, “x”, “y”],
  [“alice”, 100, 150],
  [“bob”, -10, 80]
]
```

list of lists

```
name,x,y
alice,100,150
bob,-10,80
```

CSV file

**integers look like integers**

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

dict of dicts

```
{
  “alice”: {
    “age”: 40,
    “scores”: [10,20,19]},
  “bob”: {
    “age”: 45,
    “scores”: [15,23,17,15]}
}
```

**JSON file**

# JSON

Stands for **JavaScript Object Notation**

- JavaScript is a language for web development
- JSON was developed as a way for JavaScript programs to store/share data
- JavaScript is similar to Python, which is why JSON looks like Python code

# JSON

Stands for **JavaScript Object Notation**

- JavaScript is a language for web development
- JSON was developed as a way for JavaScript programs to store/share data
- JavaScript is similar to Python, which is why JSON looks like Python code

Minor JavaScript vs. Python differences entail gotchas:

| | Python | JSON |
|---|---|---|
| **Booleans** | True, False | true, false |
| **No value** | None | null |
| **Quotes** | Single (‘) or double (“) | Only double (“) |
| **Commas** | Extra allowed: [1,2,] | No extra: [1,2] |
| **Keys** | Any type: {1: “one”} | Str only: {“1”: “one”} |

# Reading JSON Files

**JSON file saved somewhere**

```
{
  “alice”: 10,
  “bob”: 12,
  “cindy”: 15
}
```

# Reading JSON Files

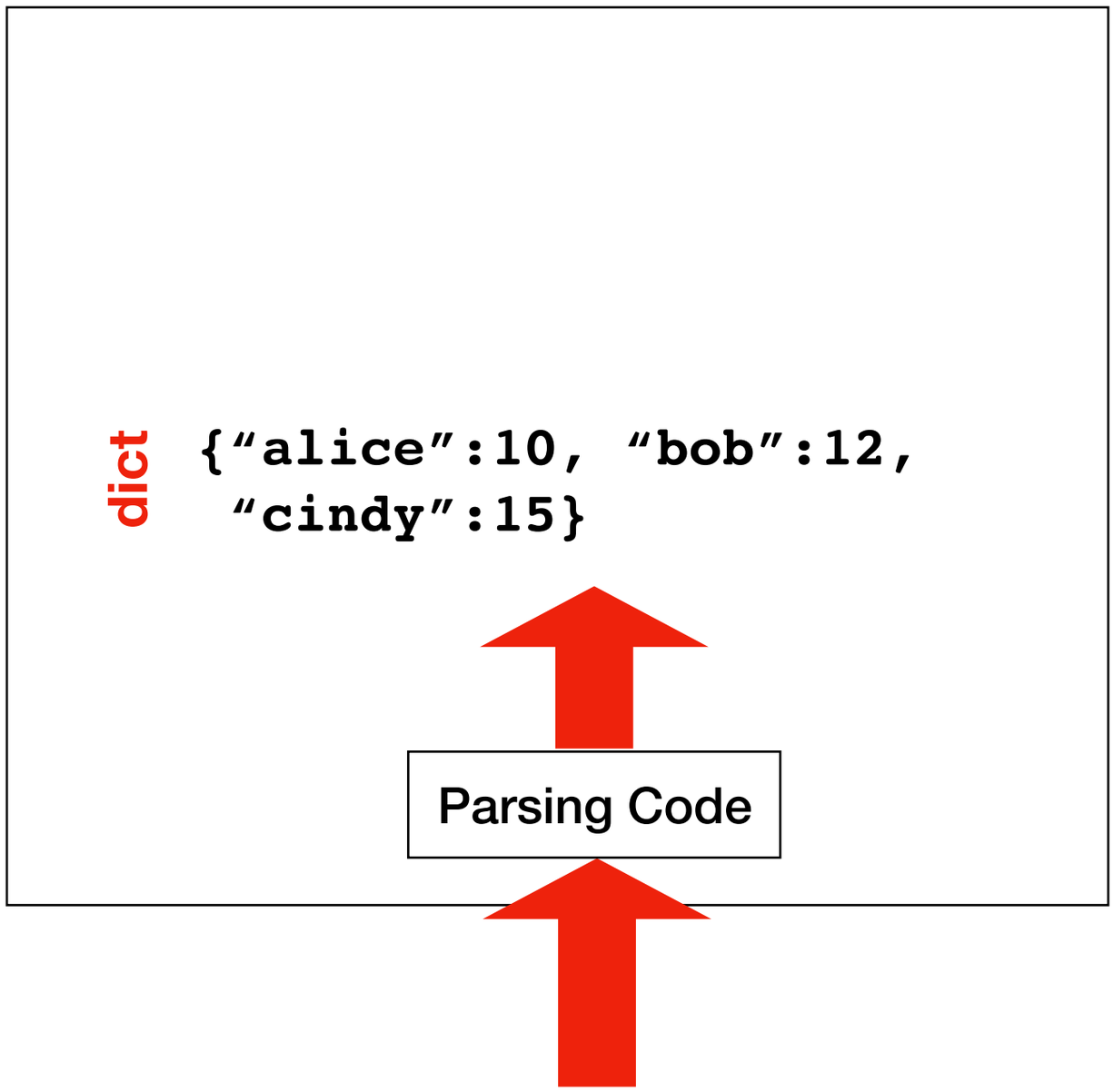
**Python Program**

Parsing Code

**JSON file saved somewhere**

```
{
  “alice”: 10,
  “bob”: 12,
  “cindy”: 15
}
```

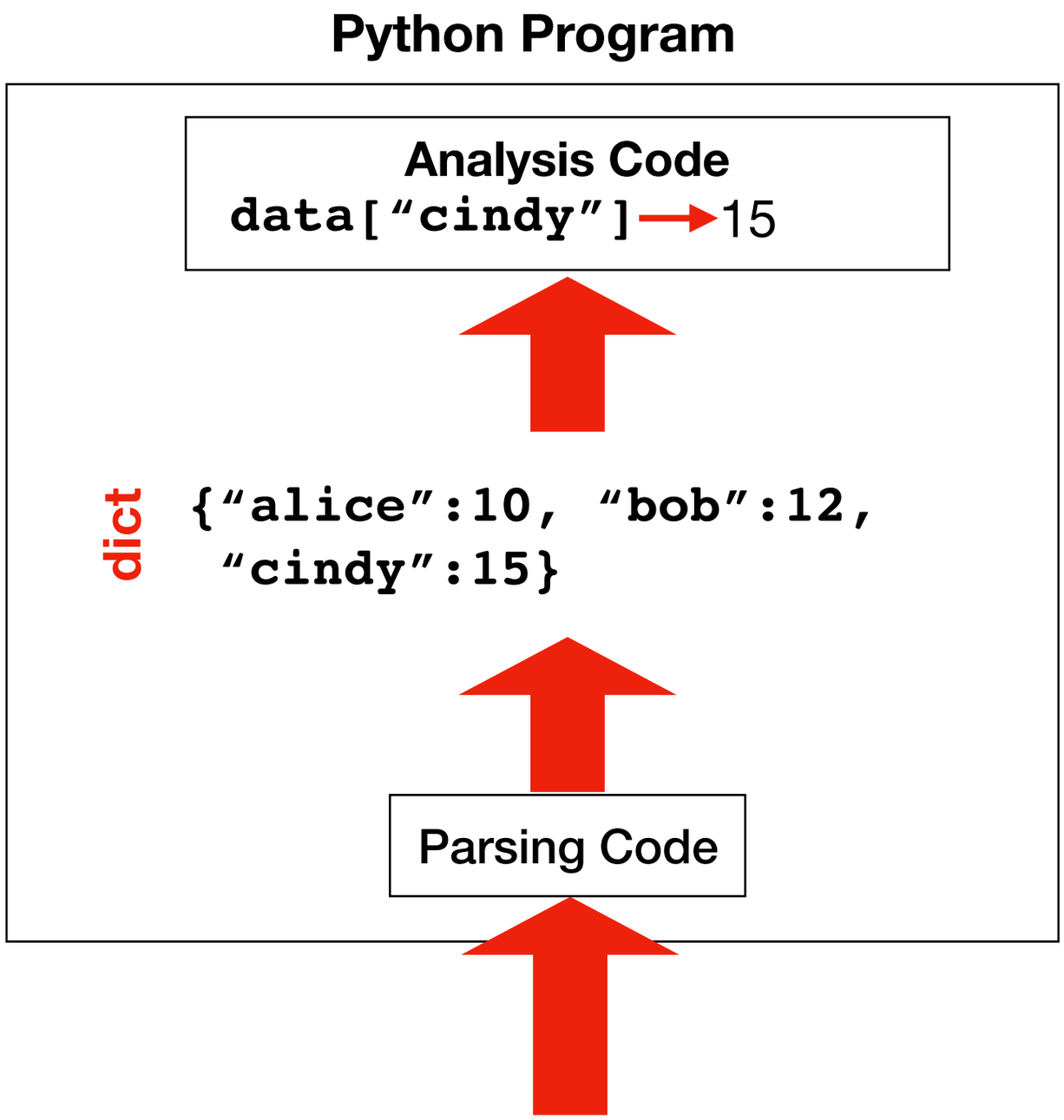
# Reading JSON Files




**JSON file saved somewhere**

```
{
  “alice”: 10,
  “bob”: 12,
  “cindy”: 15
}
```

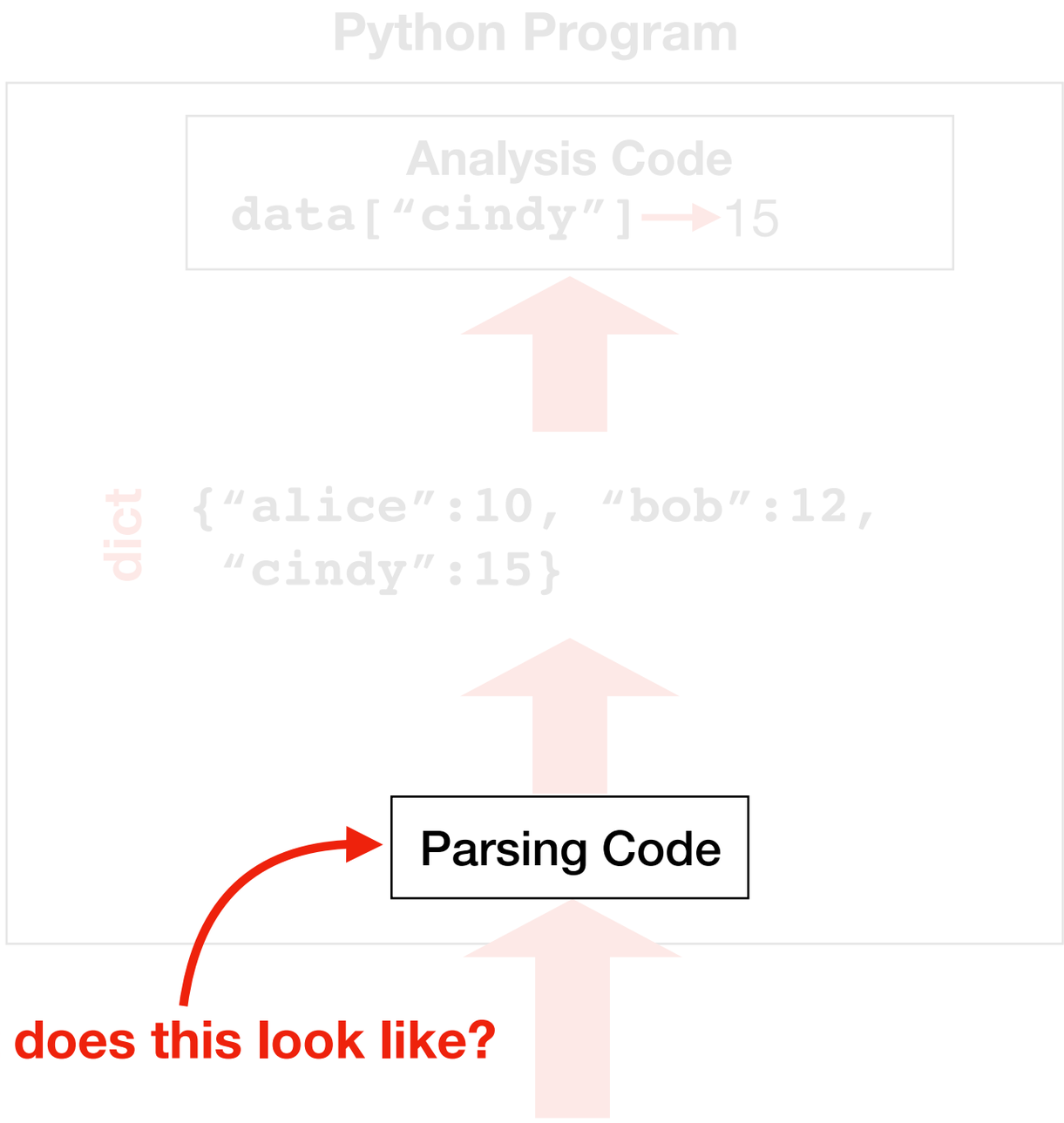
# Reading JSON Files




**JSON file saved somewhere**

```
{
  “alice”: 10,
  “bob”: 12,
  “cindy”: 15
}
```

# Reading JSON Files

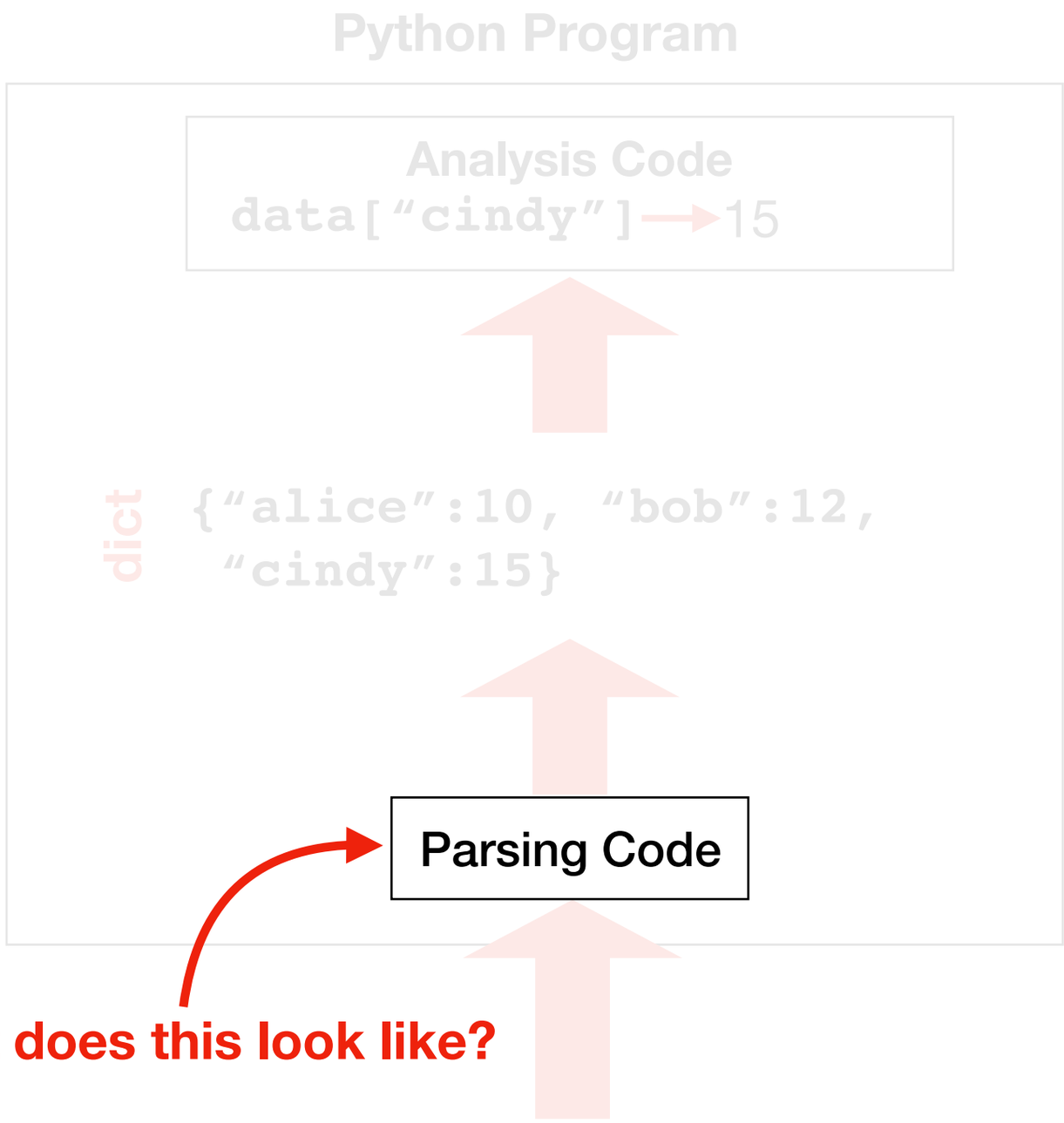



JSON file saved somewhere

```
{
  “alice”: 10,
  “bob”: 12,
  “cindy”: 15
}
```

# Reading JSON Files

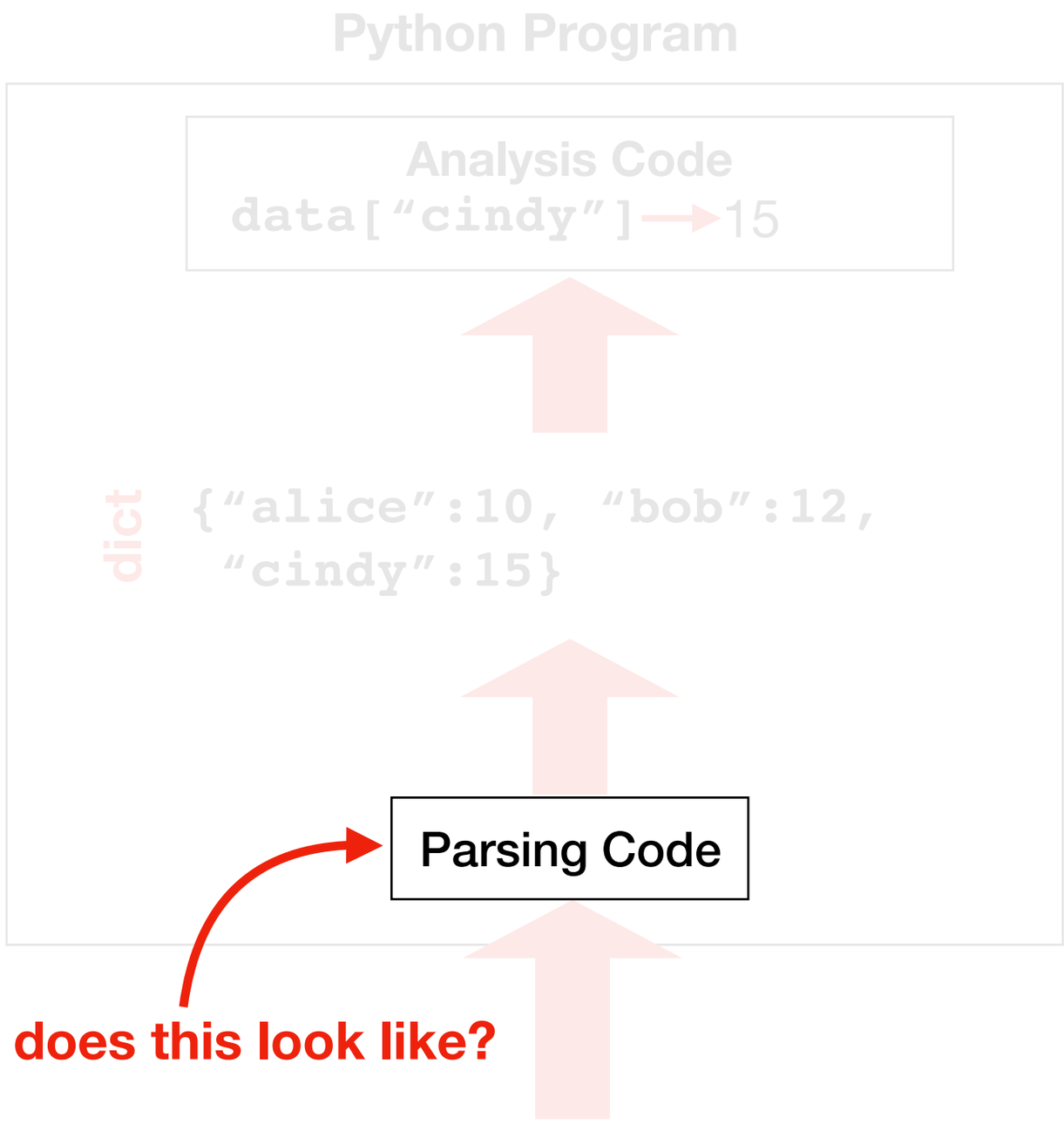
```
import json
def read_json(path):
    with open(path) as f:
        return json.load(f)
```



# Reading JSON Files

```
import json
def read_json(path):
    with open(path) as f:
        return json.load(f)
```

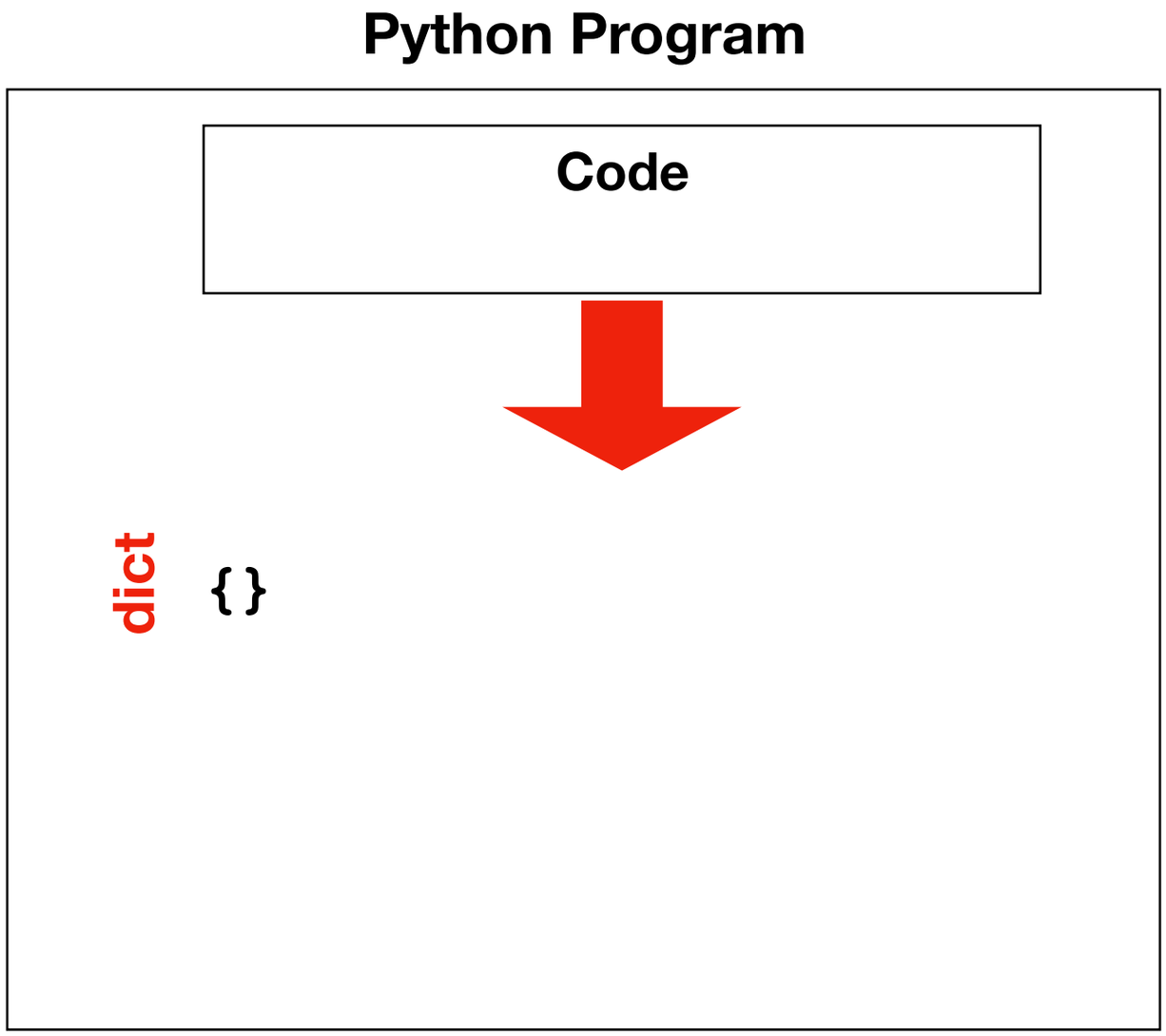
**what about writing?**



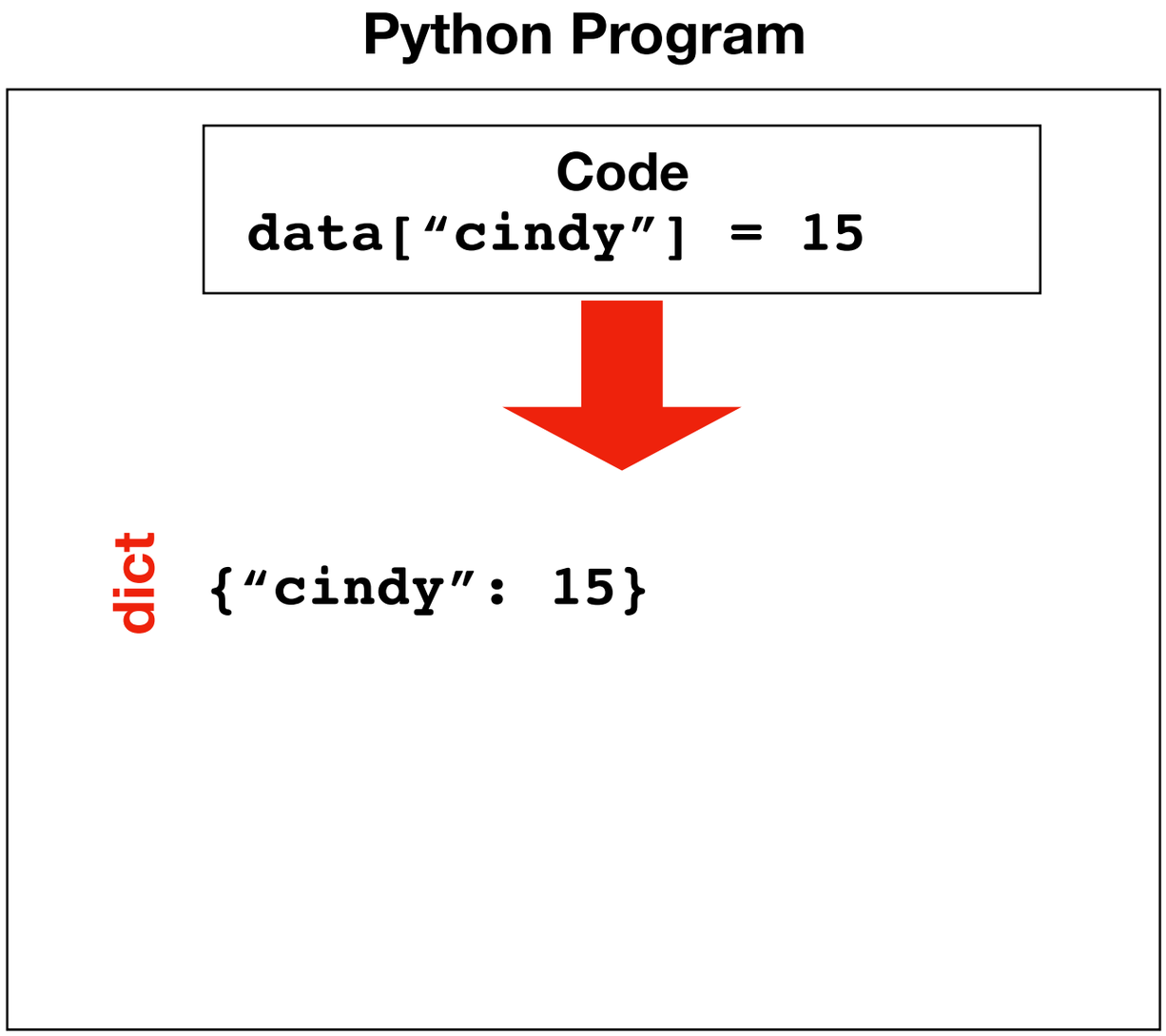

JSON file saved somewhere

```
{
  “alice”: 10,
  “bob”: 12,
  “cindy”: 15
}
```

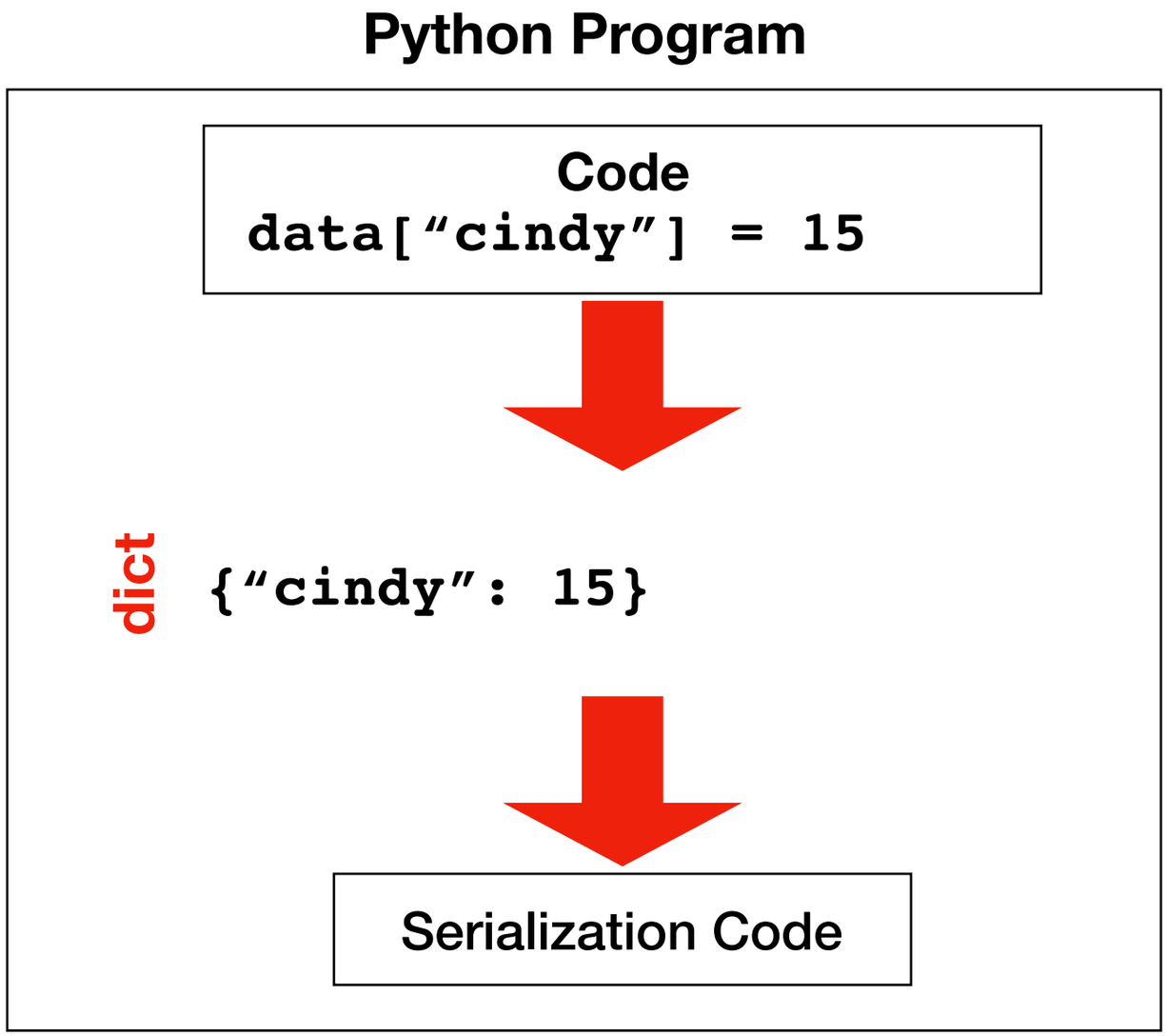
# Writing JSON Files



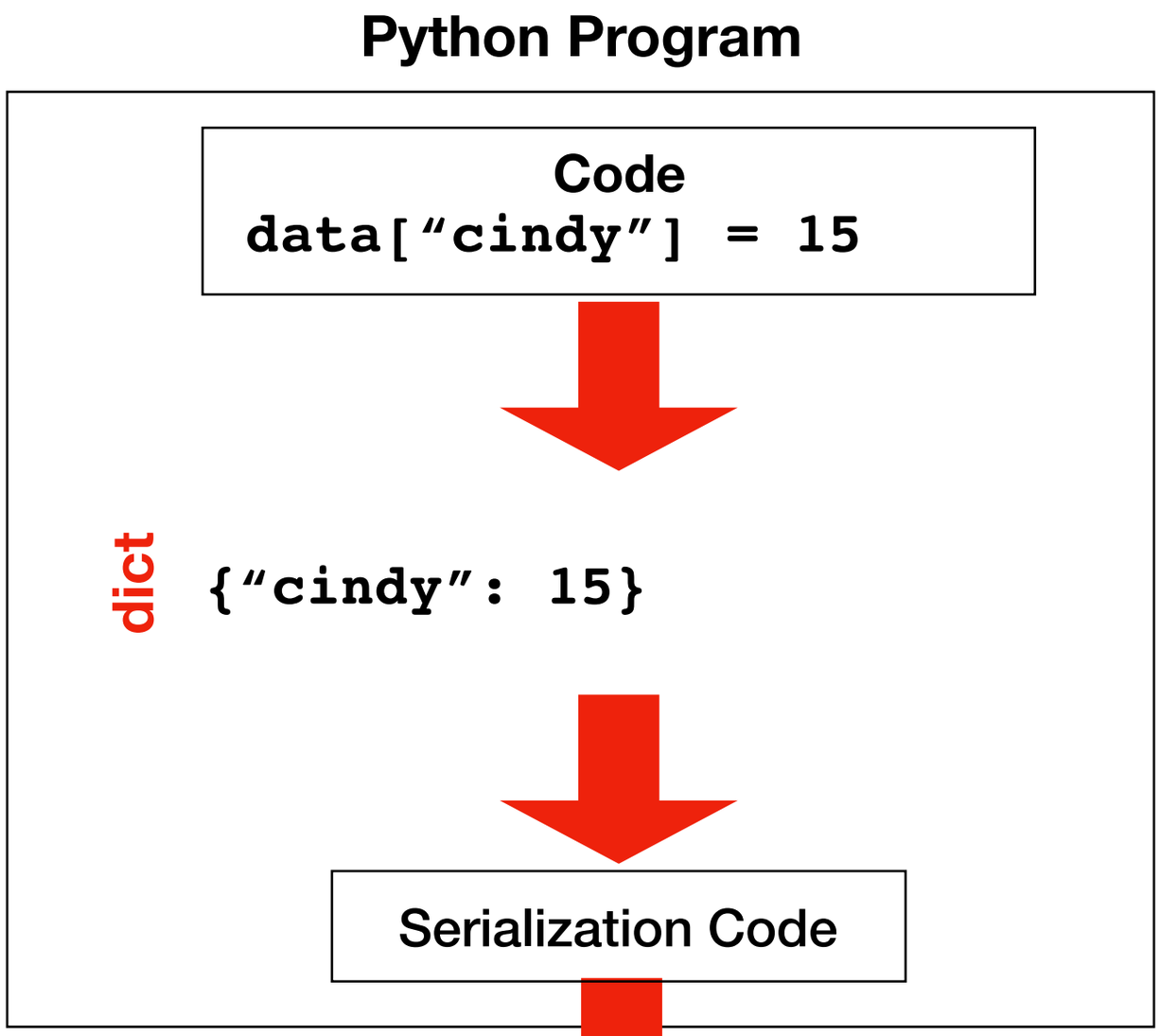
# Writing JSON Files



# Writing JSON Files



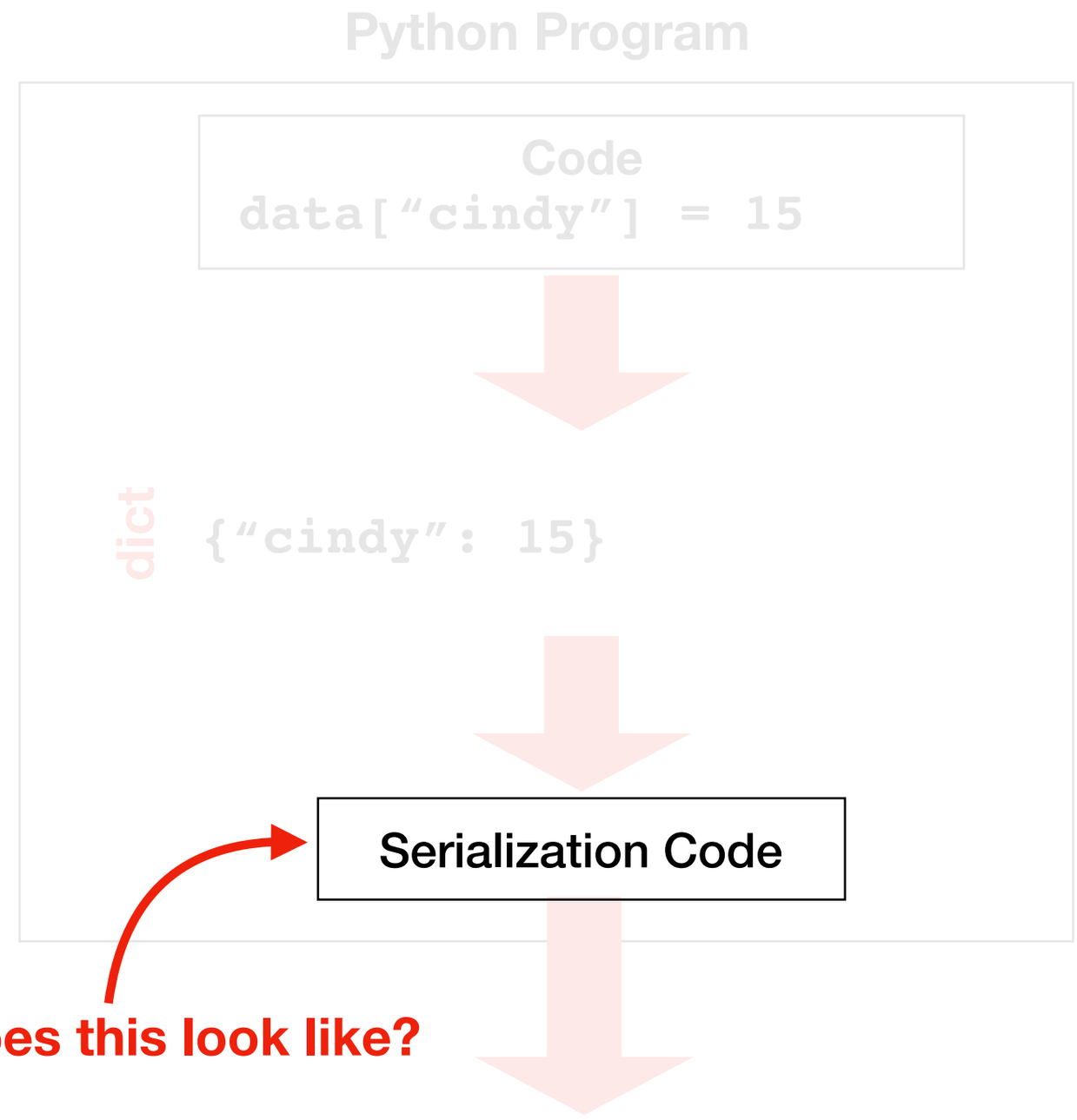
# Writing JSON Files



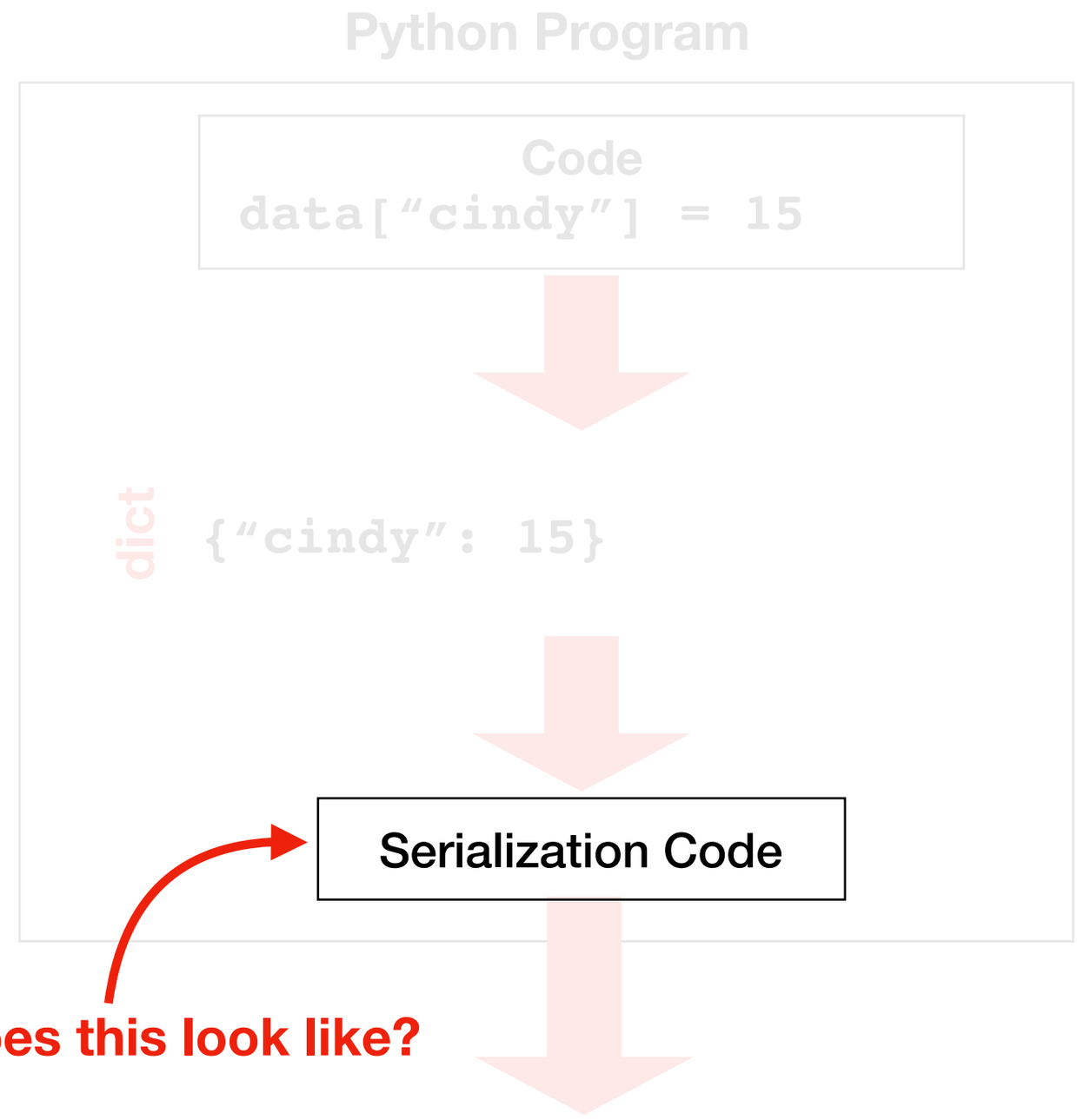

**JSON file saved somewhere**

```
{
  “cindy”: 15
}
```

# Writing JSON Files



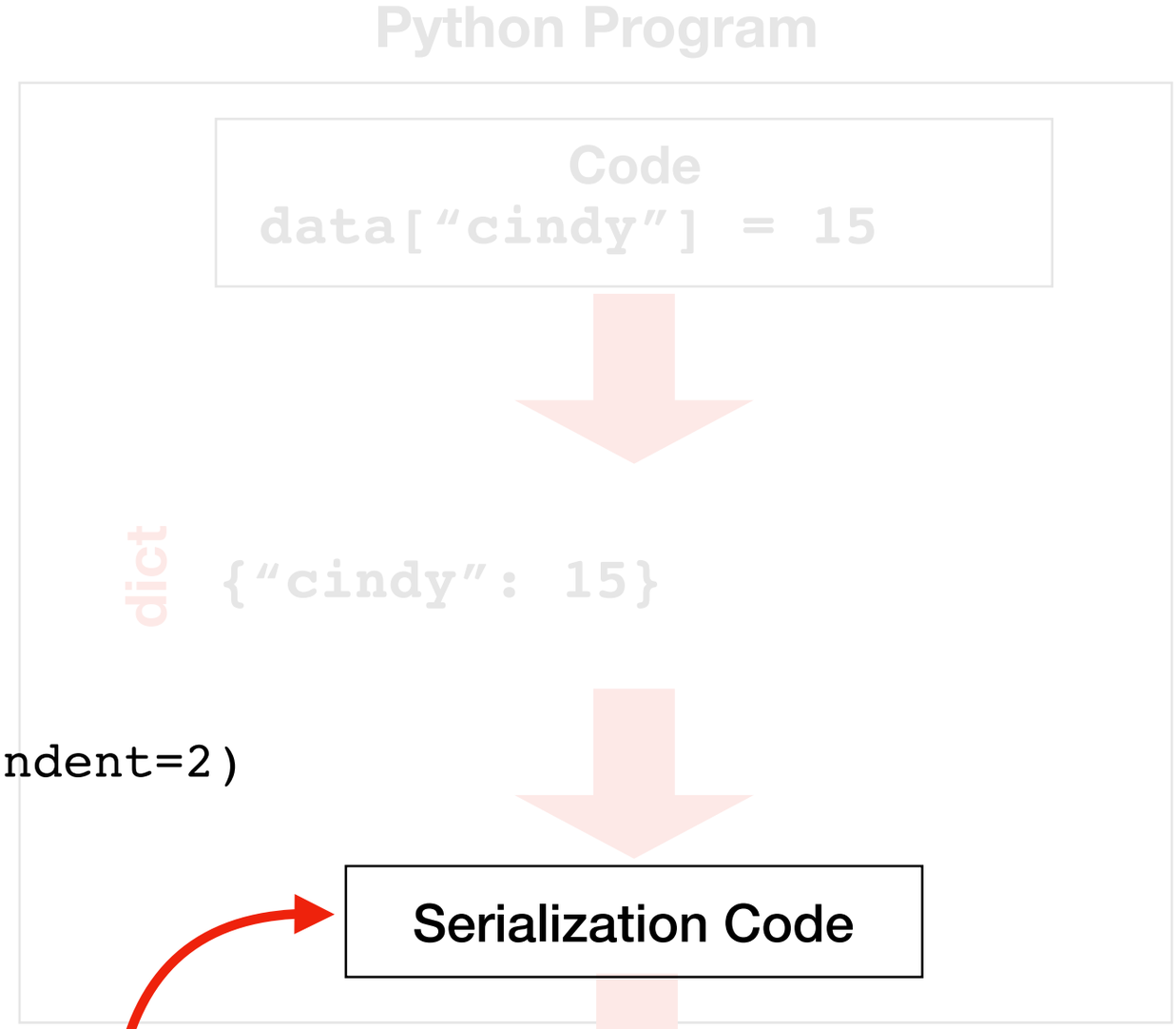
# Writing JSON Files



# Writing JSON Files

```
import json


def write_json(path, data):
    with open(path, 'w') as f:
        return json.dump(data, f, indent=2)
```




**What does this look like?**

JSON file saved somewhere

```
{
  “cindy”: 15
}
```

# Demo: Scores over Time

Goal: record scores (save across runs) and print average

**Input**:

- A **name** and a **score** to record

**Output**:

- Running average for that person

**Example**:

prompt> **python record.py alice 10**
Alice Avg: 10
prompt> **python record.py alice 20**
Alice Avg: 15
prompt> **python record.py bob 13**
Bob Avg: 13