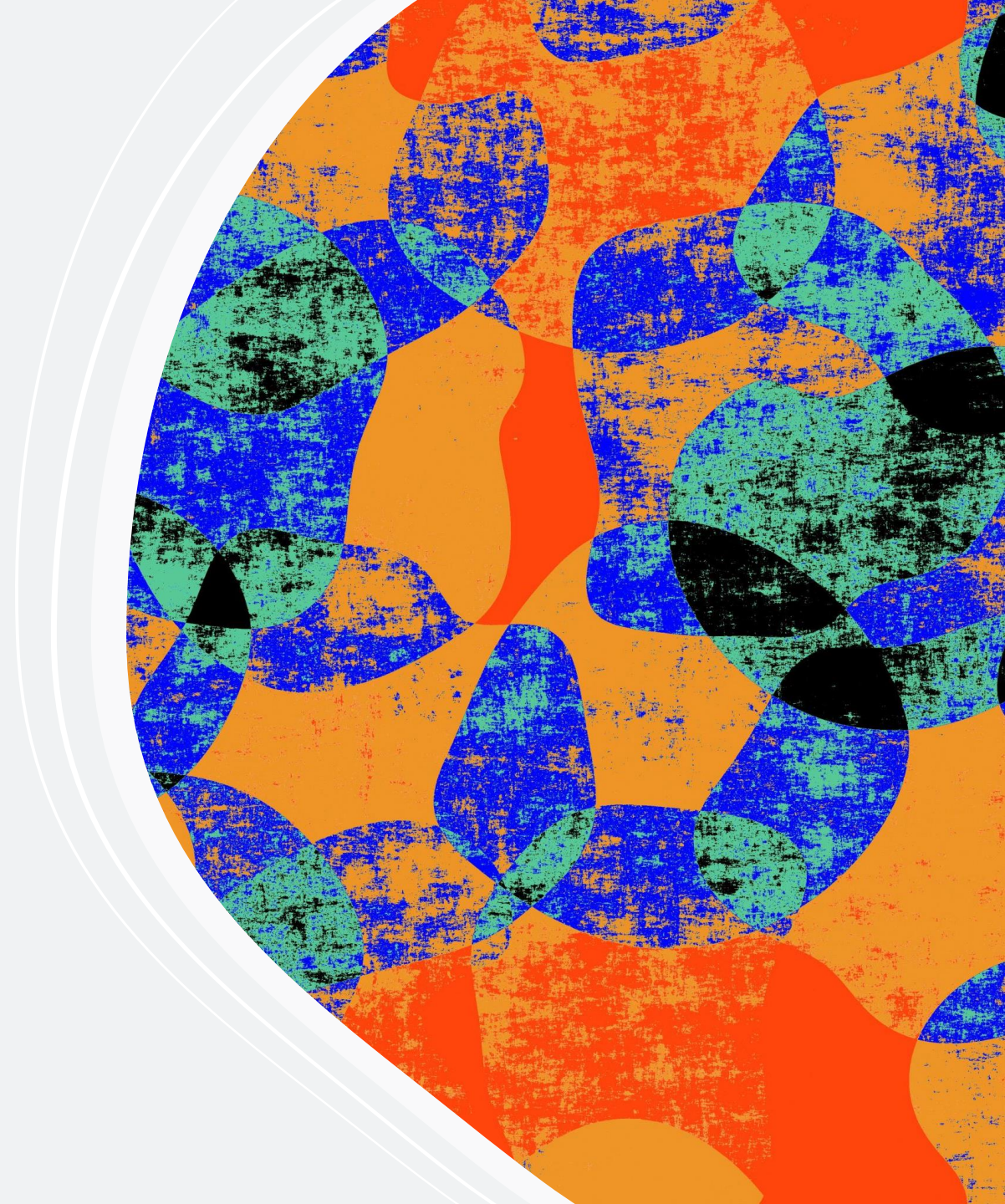


Introduction to Python Programming

A Quick Review



Using Python

- Python must be installed and configured prior to use

One of the items installed is the Python interpreter

- Python interpreter can be used in two modes:

Interactive mode: enter statements on keyboard

Script mode: save statements in Python script



Interactive Mode

- When you start Python in interactive mode, you will see a prompt

Indicates the interpreter is waiting for a Python statement to be typed

Prompt reappears after previous statement is executed

Error message displayed If you incorrectly type a statement

- Good way to learn new parts of Python

Writing Python Programs and Running Them in Script Mode

- Statements entered in interactive mode are not saved as a program
- To have a program use script mode

Save a set of Python statements in a file

The filename should have the .py extension

To run the file, or script, type

```
python filename
```

at the operating system command

line



The IDLE Programming Environment and Jupyter

- IDLE (Integrated Development Program): single program that provides tools to write, execute and test a program

Automatically installed when Python language is installed

Runs in interactive mode

Has built-in text editor with features designed to help write Python programs

<https://jupyter.org/>

Introduction to Jupyter

Data Types

- **Data types**: categorize value in memory
- e.g., int for integer, float for real number, str used for storing strings in memory
- **Numeric literal**: number written in a program
- No decimal point considered int, otherwise, considered float
- **Some operations behave differently depending on data type**



Reading Input from the Keyboard

- Most programs need to read input from the user
- Built-in input function reads input from keyboard
- Returns the data as a string
- Format: `variable = input(prompt)`
- prompt is typically a string instructing user to enter a value
- Does not automatically display a space after the prompt



Reading Numbers with the input Function

- **input function always returns a string**
- **Built-in functions convert between data types**
- `int(item)` converts *item* to an int
- `float(item)` converts *item* to a float
- Nested function call: general format: `function1(function2(argument))`
- value returned by function2 is passed to function1
- Type conversion only works if item is valid numeric value, otherwise, throws exception

The `if` Statement

Control structure: logical design that controls order in which set of statements execute

Sequence structure: set of statements that execute in the order they appear

Decision structure: specific action(s) performed only if a condition exists

Also known as selection structure

The `if` Statement (cont'd.)

Python syntax:

```
if condition:
```

```
    Statement
```

```
    Statement
```

First line known as the `if` clause

Includes the keyword `if` followed by condition

- *The condition can be true or false*
- *When the `if` statement executes, the condition is tested, and if it is true the block statements are executed. otherwise, block statements are skipped*

The `if-else` Statement (cont'd.)

If the condition is true, this block of statements is executed.

```
if condition:  
    statement  
    statement  
    etc.  
else:  
    statement  
    statement  
    etc.
```

Then, control jumps here, to the statement following the `if-else` statement.

```
if condition:  
    statement  
    statement  
    etc.  
else:  
    statement  
    statement  
    etc.
```

If the condition is false, this block of statements is executed.

Then, control jumps here, to the statement following the `if-else` statement.

The while Loop: a Condition-Controlled Loop

- while loop: while condition is true, do something
- Two parts:
- Condition tested for true or false value
- Statements repeated as long as condition is true
- In flow chart, line goes back to previous part
- General format:

while condition:

statements



The for Loop: a Count-Controlled Loop

- **Count-Controlled loop**: iterates a specific number of times

- Use a for statement to write count-controlled loop

- Designed to work with sequence of data items

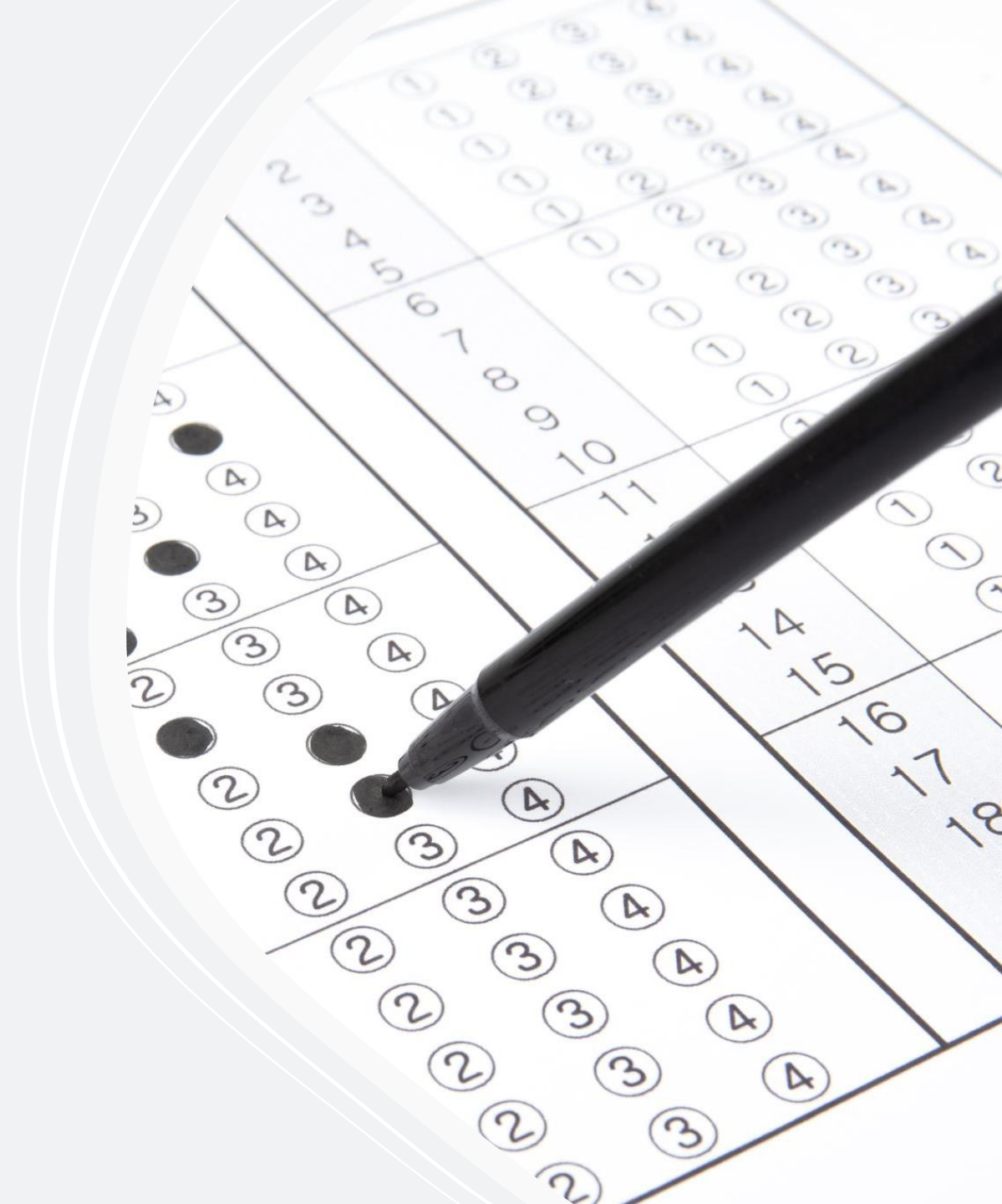
- Iterates once for each item in the sequence

- General format:

```
for variable in [val1, val2, etc]:
```

```
    statements
```

- Target variable: the variable which is the target of the assignment at the beginning of each iteration



Void Functions and Value-Returning Functions

- A void function:

Simply executes the statements it contains and then terminates.

- A value-returning function:

Executes the statements it contains, and then it returns a value back to the statement that called it.

- *The `input`, `int`, and `float` functions are examples of value-returning functions.*



Defining and Calling a Function (cont'd.)

Function name should be descriptive of the task carried out by the function

Often includes a verb

Function definition: specifies what function does

```
def function_name():
```

```
    statement
```

```
    statement
```

Introduction to File Input and Output

- For program to retain data between the times it is run, you must save the data

Data is saved to a file, typically on computer disk

Saved data can be retrieved and used at a later time

- "Writing data to": saving data on a file
- Output file: a file that data is written to



Opening a File

- open function: used to open a file

Creates a file object and associates it with a file on the disk

General format:

```
file_object = open(filename,  
mode)
```

- Mode: string specifying how the file will be opened

Example: reading only ('r'), writing ('w'), and appending ('a')



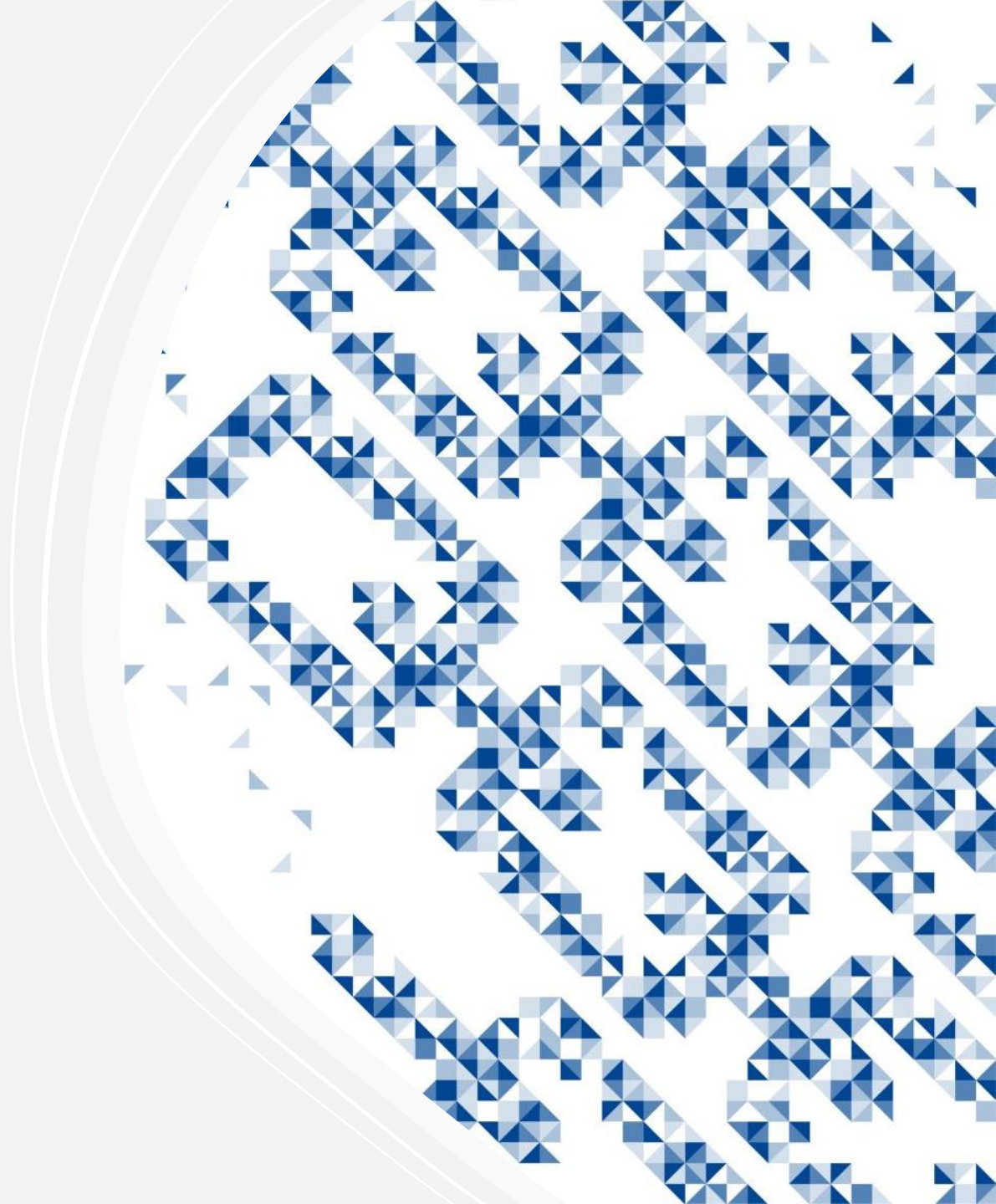
Using Python's `for` Loop to Read Lines

- Python allows the programmer to write a `for` loop that automatically reads lines in a file and stops when end of file is reached

Format: `for line in file_object:`

`statements`

The loop iterates once over each line in the file



Exceptions

- Exception: error that occurs while a program is running

Usually causes program to abruptly halt

- Traceback: error message that gives information regarding line numbers that caused the exception

Indicates the type of exception and brief description of the error that caused exception to be raised

Introduction to Lists

- **List:** an object that contains multiple data items
- **Element:** An item in a list
- Format: `list = [item1, item2, etc.]`
- Can hold items of different types
- **print function** can be used to display an entire list
- **list()** function can convert certain types of objects to lists

Basic String Operations

- Many types of programs perform operations on strings
- In Python, many tools for examining and manipulating strings

Strings are sequences, so many of the tools that work with sequences work with strings

<https://www.programiz.com/python-programming/methods/string>

Dictionaries

- Dictionary: object that stores a collection of data

Each element consists of a *key* and a *value*

- *Often referred to as mapping of key to value*
- *Key must be an immutable object*

To retrieve a specific value, use the key associated with it

Format for creating a dictionary

```
dictionary =  
  
    {key1:val1, key2:val2}
```

<https://www.programiz.com/python-programming/methods/dictionary>



SQL Database

Example

Class Activity

Exercises to be conducted on the various topics discussed in this lesson