PYTHON
BEGINNER TO EXPERT

Agenda

- Basics of Python
  - Installation
  - Variables
  - Keywords
  - Data types
  - Operators/Operands
- Python Data Structures
  - Sets
  - Lists
  - Dictionaries
  - Tuples
- Python Loops, Functions, & File Handling
  - Loops
  - Functions
  - Lambda Functions
  - Map, Reduce, & Filter
  - File Handling
- Python Exception Handling
- Iterators & Generators

Introduction

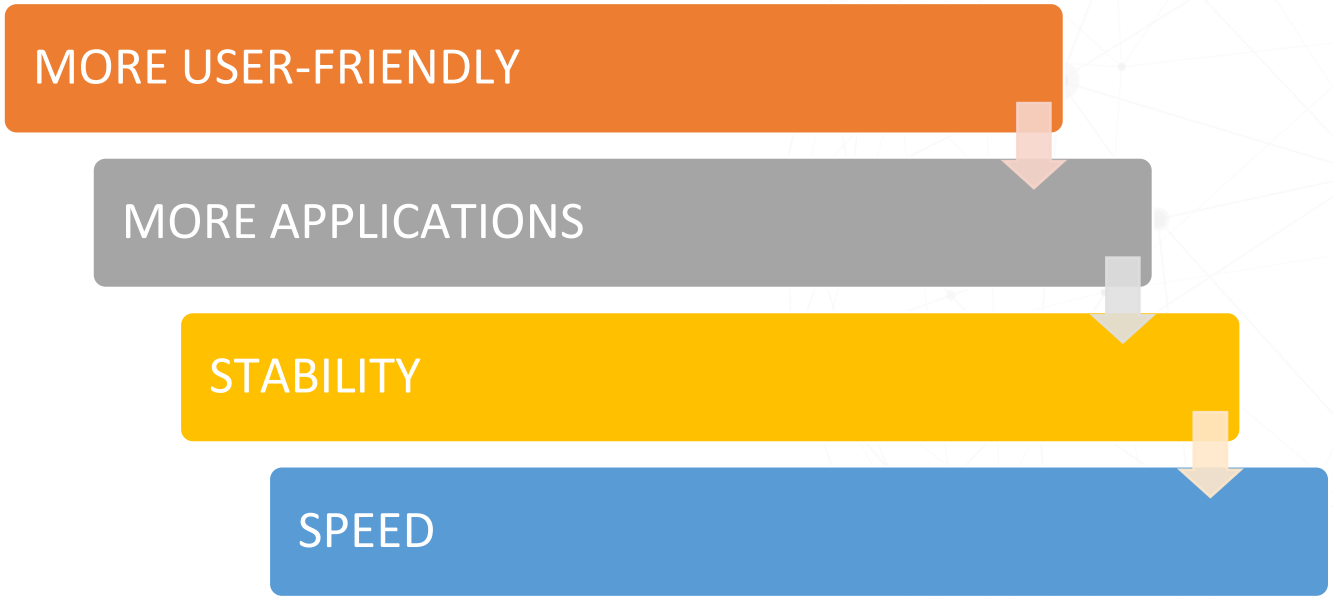
- Open Source general purpose programming language
- Object Oriented
- Programming as well as scripting language

"Python is a general-purpose programming language that is often applied in scripting roles"

Features

- Easy to learn & use
- Interpreted language
- Open Source
- Large Standard Library
- Large Community Support
- Extensible
- Cross-platform language

PYTHON vs OTHER PROGRAMMING LANGUAGES



1. Basics of Python

Let's start with the first chapter

Installation

Install Python on your machine.

- You can download from the official website
- *https://www.python.org/downloads*

Install Anaconda

- *https://www.anaconda.com/download*

```
(base) C:\Users\pattn>python --version
Python 3.7.3
```

Variables

You can consider a variable to be a temporary storage space where you can keep changing values.

Assigning values to a variable:

- To assign values to a variable in Python, we will use the assignment (=) operator.

  a = 10, a = "Welcome to the class"

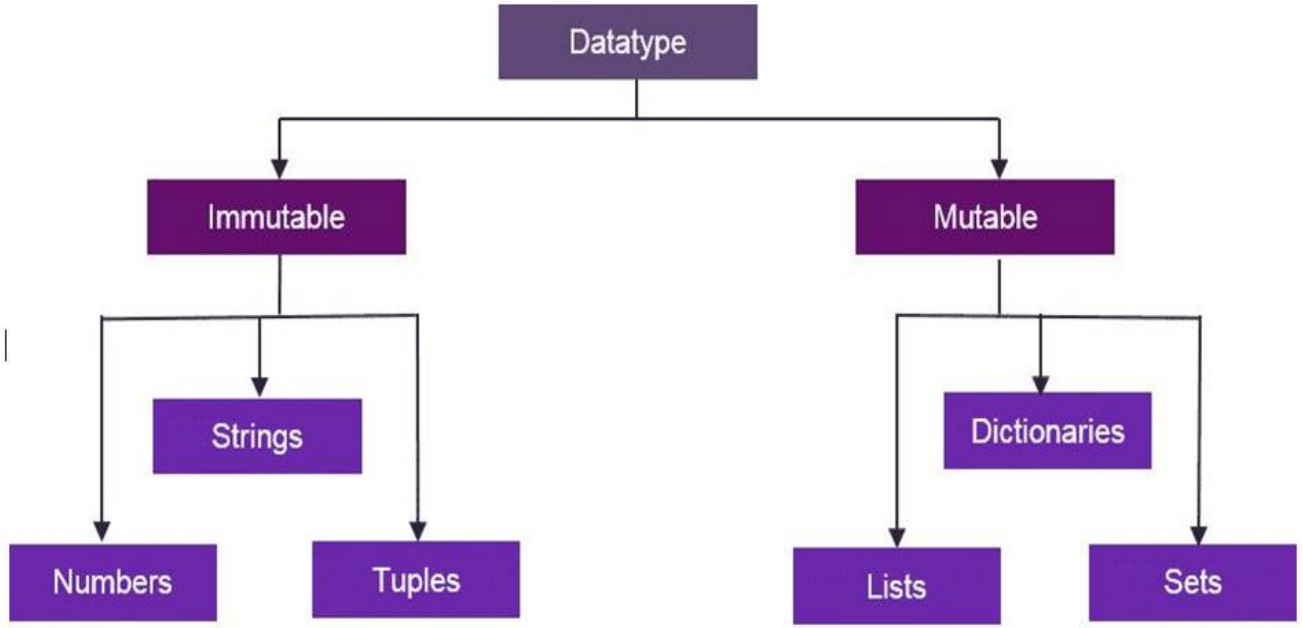
- No need to declare the datatype of the variables done in other programming languages

Keywords

- Special reserved words which convey a special meaning to interpreter or compiler.
- It can’t be used as a variable.
- Few of the example keywords are as follows:
    - def, else, if, class, continue, break, finally, from, return, lambda, except, import, None

Data Types

- Variables hold values of different data types.
- With the help of type() function you can check the type of variable used.



Operators & Operands

- Operators are special symbols that represent computations like addition and multiplication.
- Value of operators is applied to are called operands.
- Operator : -, +, /,*,**

20+32, hour-1, hour*60+minute 5**2

- Order of Precedence : PEMD (Parenthesis, Exponentiation, Multiplication and Operators)

2. Python Data Structure

Let's start with the second chapter

Lists

- List is a sequence of values of any type.
- Values in lists are called as elements or items.

  [10, 20, 'Class']
- A list within another list is nested.
- Lists are mutable.
- It has variable length.
- Lists are accessed similarly like arrays. First element will be stored at 0^{th} index.
- Let's discuss function used in lists.

Tuples

- Tuples are similar like lists, having a sequence of values of any type and enclosed within parentheses.
- Tuples are immutable.
- It has fixed length.
- tup_1 = ('a', 1, 'df', 'b')
- Let's discuss about the functions of tuples.

List vs Tuples

List

1. Items surrounded in square brackets []
2. Lists are mutable in nature
3. There are more than 40 available methods in Lists.
4. If content is not fixed, and keeps on changing then we should go for lists.
5. List objects cannot be used as keys for dictionaries because keys should be Hash table and immutable.

Tuple

1. Items surrounded in round brackets ()
2. Tuples are immutable in nature
3. There are almost 30-35 available methods in Tuples.
4. If content is fixed, and never changes then we should go for Tuples.
5. Tuple objects can be used as keys for dictionaries because keys should be Hash table and immutable.

Sets

- Unordered collection of items is known as set.
- Items of set can not be duplicate.

Colors = { ‘red’, ‘blue’ , ‘green’ }

- Let’s see how to use, Union, Intersection methods in set.
- Let’s discuss about the other functions of Sets.

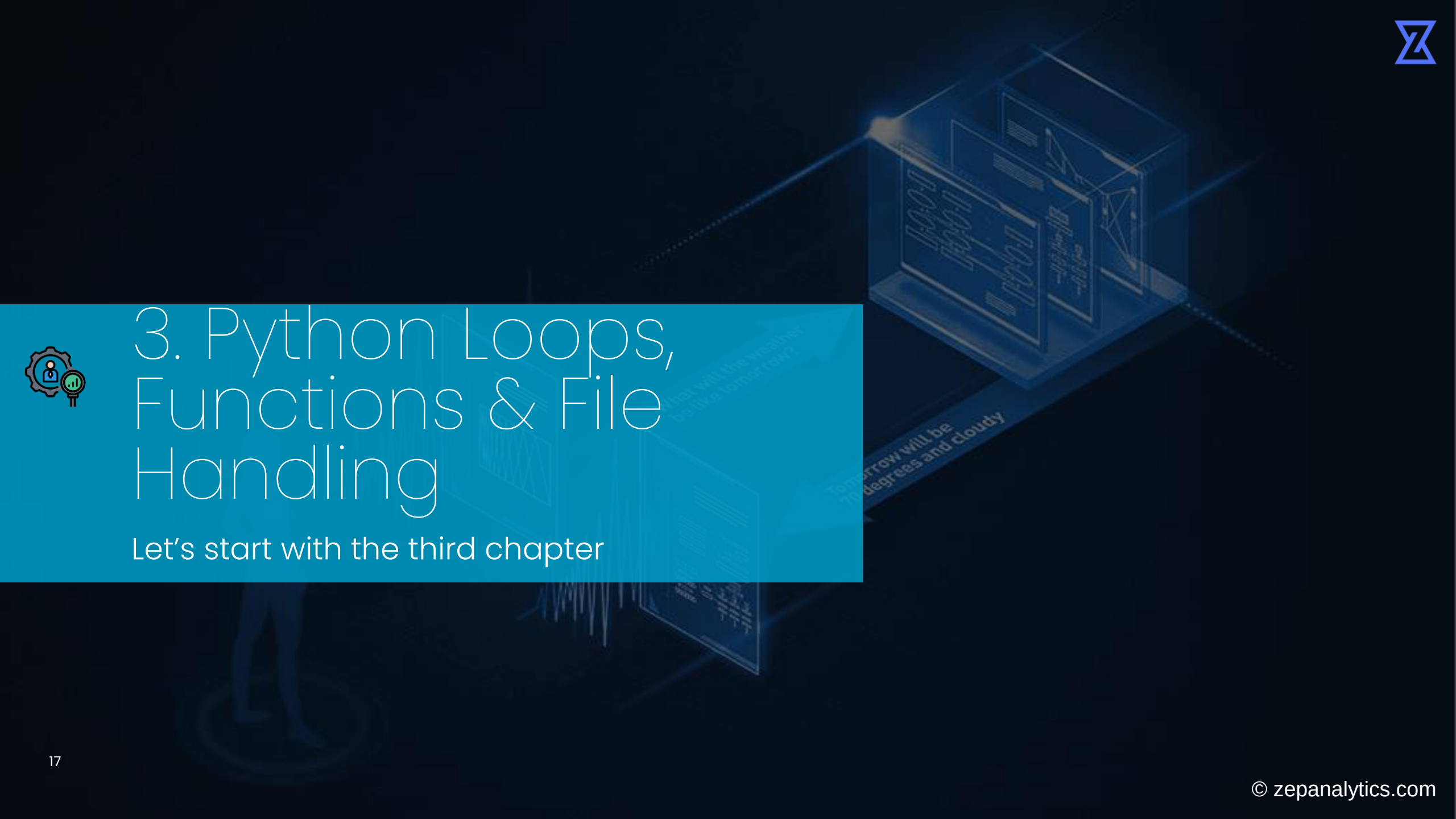
Dictionary

- Unordered collection of data.
- Data in dictionary is stored as key:value pair.
- Key should not be mutable and value can be of any type.
- Dict = { “name”: “begindatum”, ‘age’:10 }

Keys: name and age

Values: begindatum and 10

- Let’s see how to access the keys and values in dictionary along with different functions used.



3. Python Loops, Functions & File Handling

Let's start with the third chapter

Loops

- For loop

```
for index in sequence:
        statements
for i in range(1,10):
        print(i)

for i in range(0,5):
    print(i)
else:print("for loop completely exhausted, since there is no break.");
```

Once for loop is executed, else block is also executed.

Functions

- Function is a named-sequence of statements that performs some operations.

  Example: type(32)

  Here, name of function is type and expression in parenthesis is called argument of the function.

- Type Conversion Function : Convert values from one type to another. Int can convert floating-point values to integers and vice-versa

  int(3.4545)

  3

Lambda Function

- Lambda Function is a small anonymous function which makes the developer's life easier
- It can take any number of arguments, but can only have one expression.

Syntax

```
lambda arguments : expression
```

The expression is executed and the result is returned:

Map, Reduce & Filter

Map → Utility function, maps a collection to another collection object based on certain functionality.

map(function, iterable object)

For example: If we have list of people like:

firstname = ["Ram", "Shyam", "Vinay", "Gopal"]

- Map the list to obtain the names in upper case
- list(map(lambda x:x.upper(), firstname))

Map, Reduce & Filter

Filter→ Similar function, but it requires the function to look for a condition and then returns only those elements from the collection that satisfies the condition.

Reduce → An operation that breaks down the entire process into pair-wise operations and uses the result from each operation, with the successive element.

Filter→

Data = [d1,d2,.....,dn]

filter(f, data)

Reduce →

Function = f(x,y)

reduce(f,data):

Step 1: val1 = f(d1,d2)

Step 2: val2 = f(val1, d3)

Step n-1: val(n-1) = f(val(n-2), dn)

File Handling

- Let's see how to read and write with the files in Python. We will use basic functions and methods like open(), read(), close() to perform the file manipulation.
- file object = open(file_name [, access_mode][, buffering])

Fileopen = open("filename", "r")

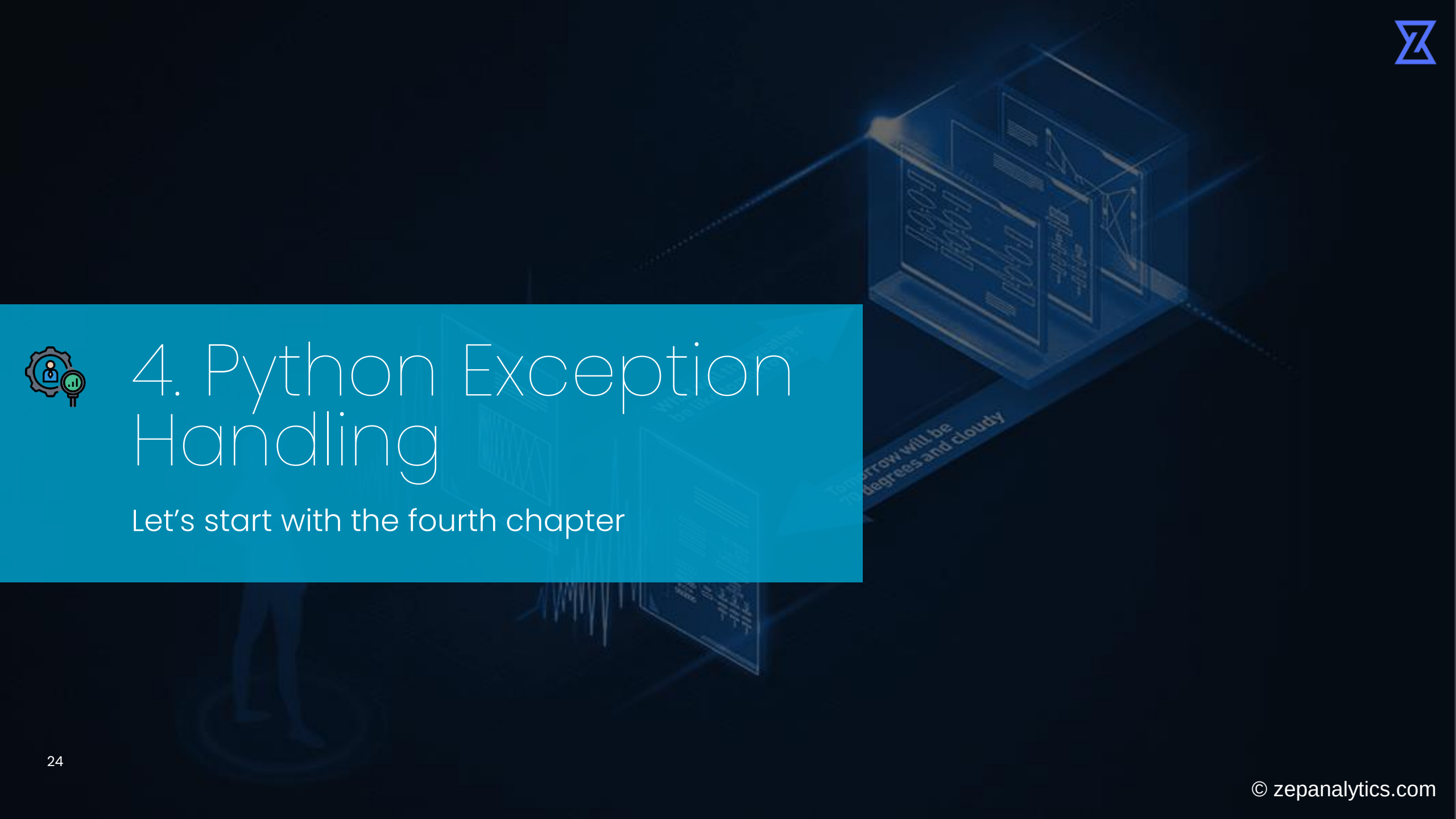
Fileopen.close()

Fileopen.read()

Fileopen.readline() // Read lines of the file

Fileopen.write("Welcome to the class")

Let's see the programs for file operations.



4. Python Exception Handling

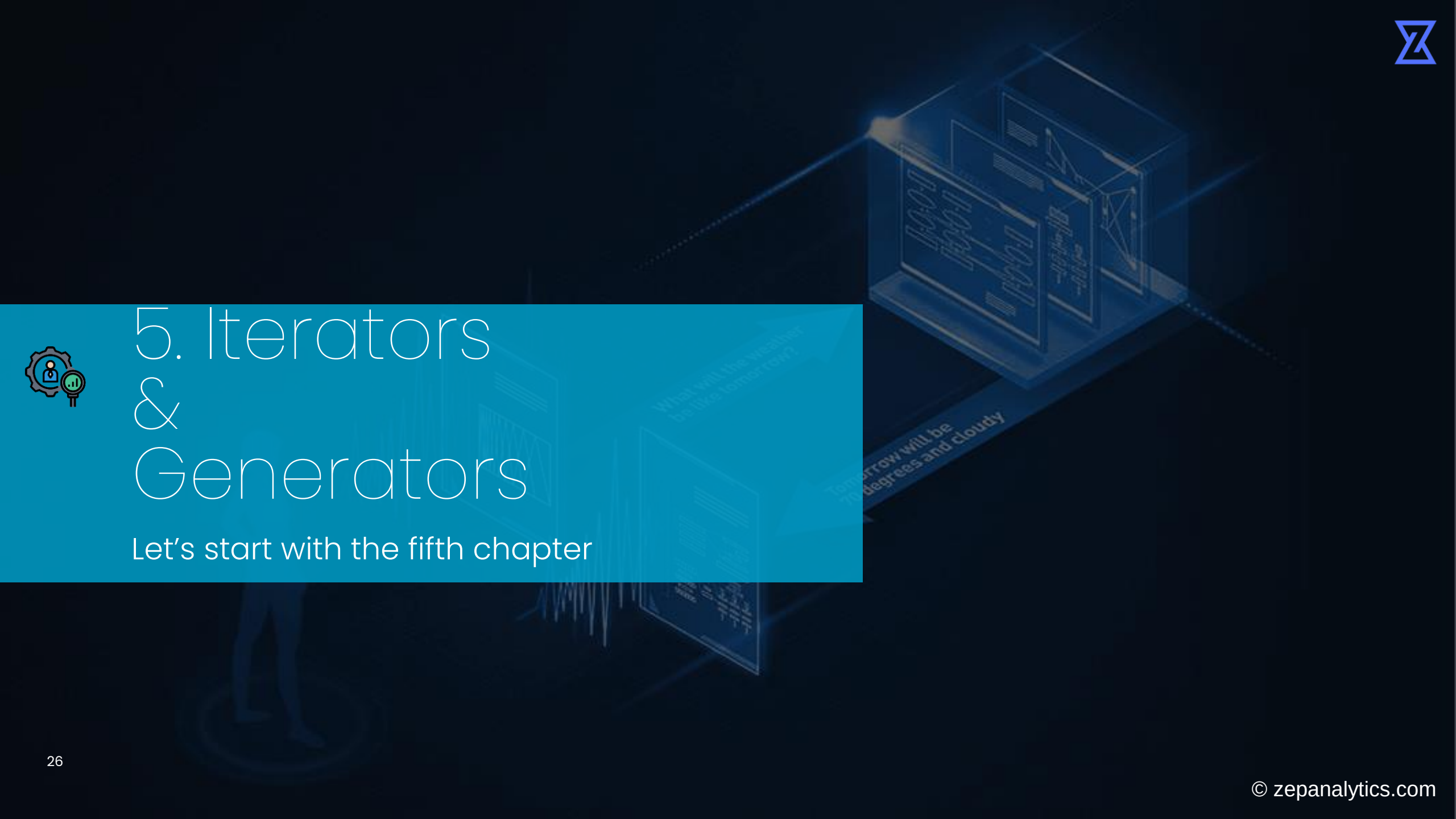
Let's start with the fourth chapter

Exception Handling

- An exception is an abnormal condition or error that occurs during the execution of program.
- Some common exceptions are as follows:

- NameError
- ZeroDivisionError
- Indentation Error
- IO Error

Let's see how to use try, catch, except and finally to handle the exceptions.



5. Iterators & Generators

Let's start with the fifth chapter

Iterators

- Iterator in Python is simply an object that can be iterated upon. An object which will return data, one element at a time.
- It implement two special methods, __iter__() and __next__(), together called the iterator protocol.
- Let's understand more in detail with help of the program.

Generators

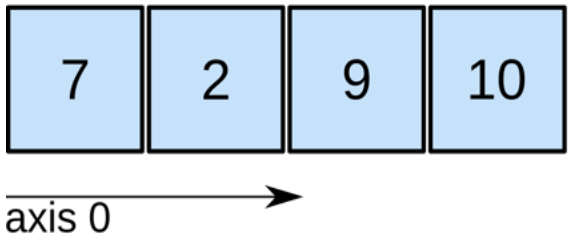
- Python Generators are simple way of creating iterators. Basically, it is a function that returns an object (iterator) which we can iterate over (one value at a time).
- If a function contains at least one yield statement (it may contain yield or return statements), it becomes a generator function.
- The difference, is that, while a return statement terminates a function entirely, yield statement pauses the function saving all its states and later continues from there on successive calls.
- Let's understand in more details with help of program

6. Python for Data Science

Let’s start with the sixth chapter

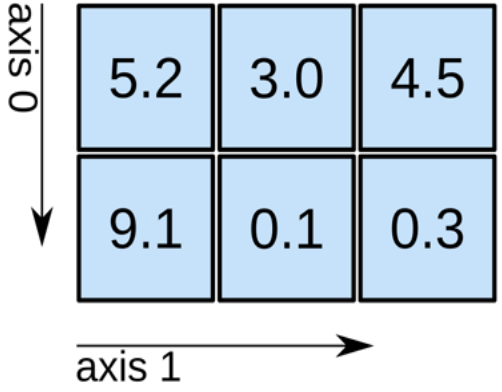
NumPy

1D array



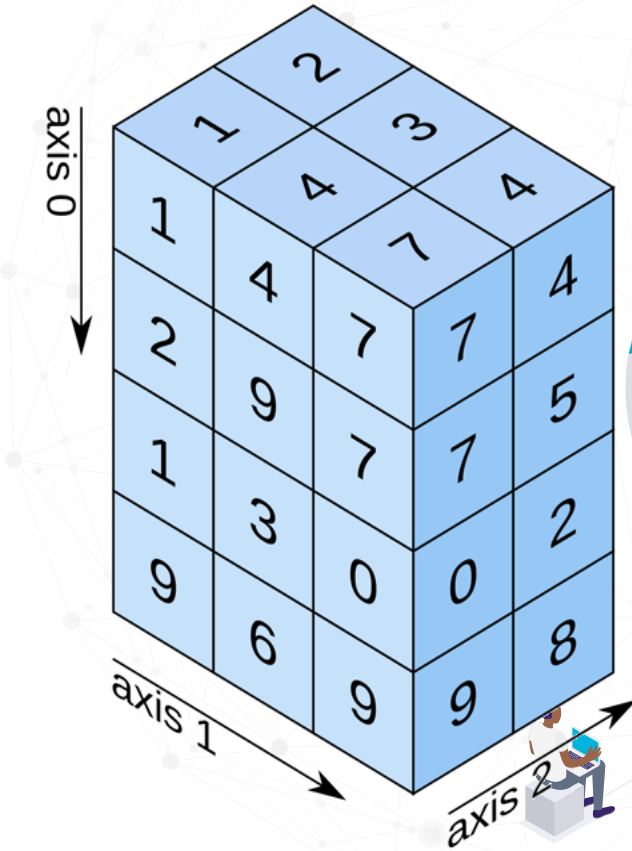
shape: (4,)

2D array



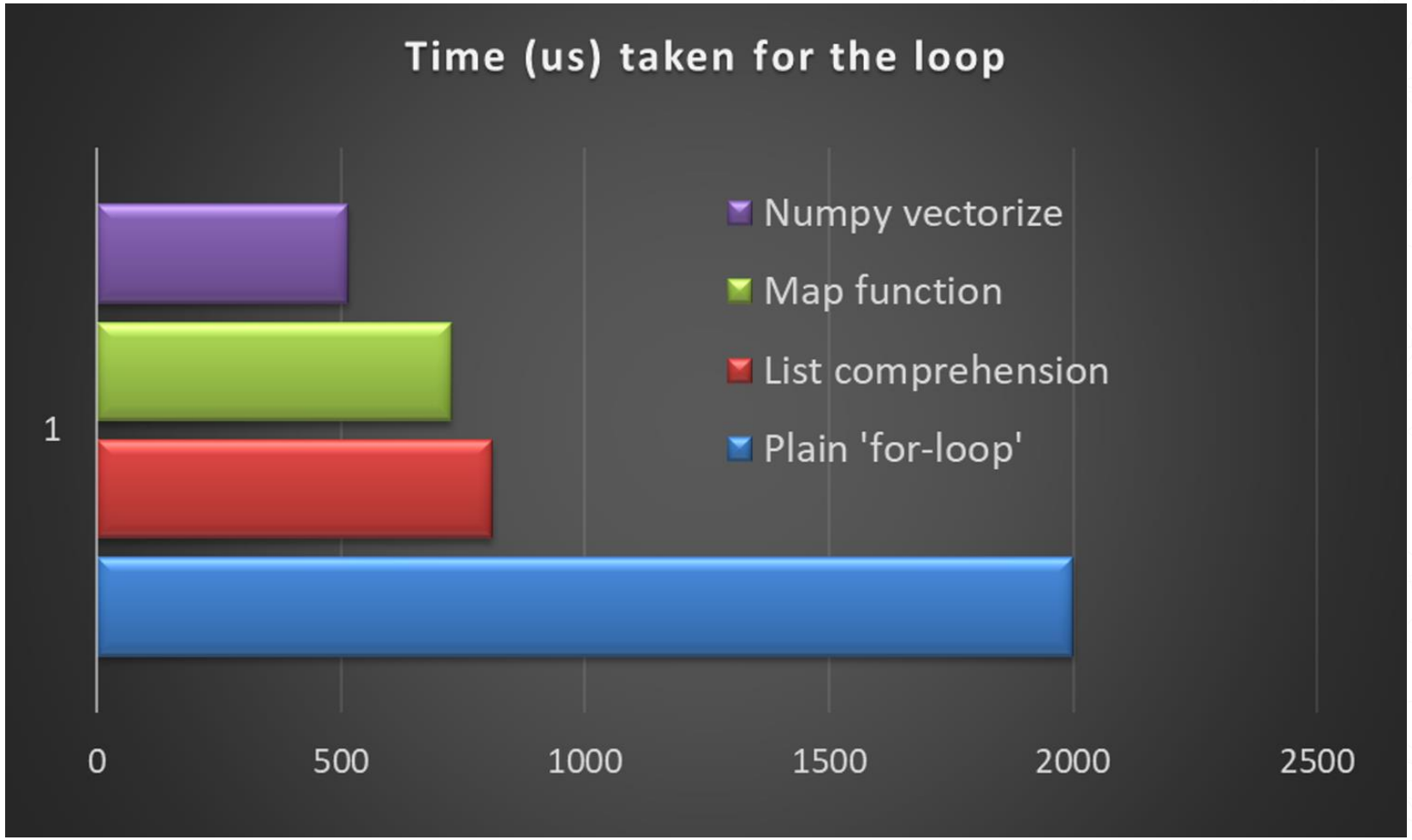
shape: (2, 3)

3D array



shape: (4, 3, 2)

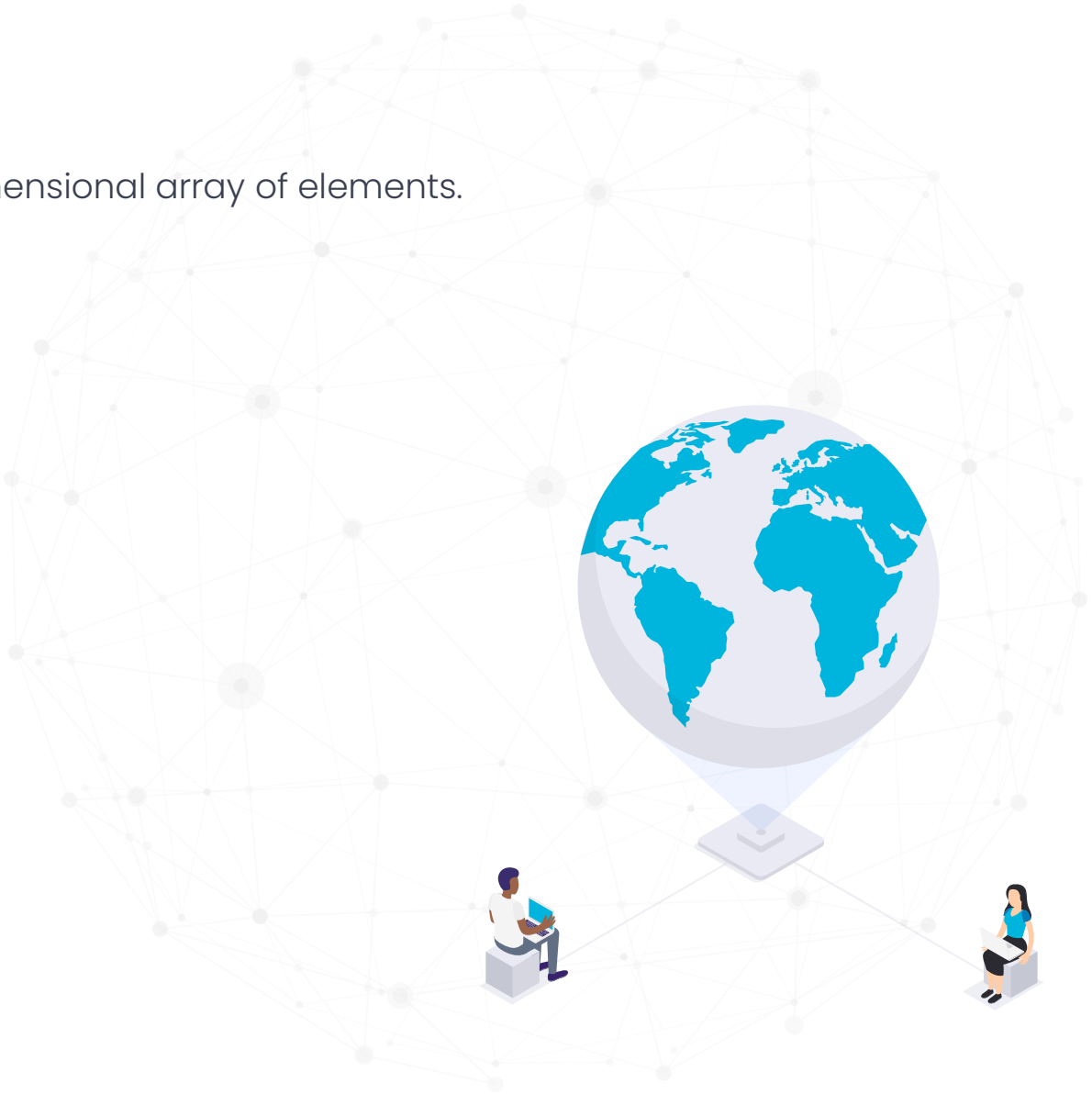
NumPy



NumPy

- It is used for computation and processing of single and multi dimensional array of elements.
- Smaller memory consumption and better runtime behavior.
  - To install the module: **pip install numpy**
  - To use NumPy : import numpy as np
  - values = [20.1, 20.4, 12.3, 43.5, 54.4, 23.5]
  - Convert = np.array(values)
  - print(Convert)
  - print(Convert+20)

Let's see more examples on this.



Pandas

Pandas is fast, powerful, flexible and easy to use open source data analysis and manipulation tool.

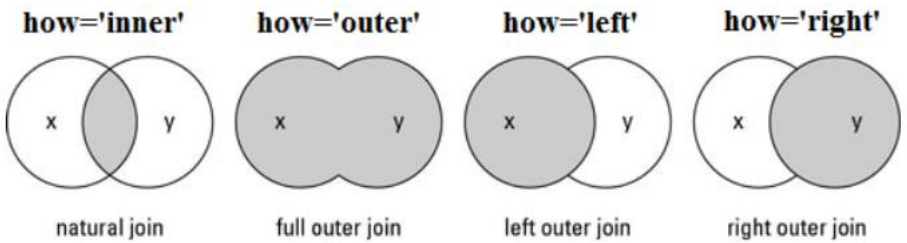
To install pandas: **pip install pandas**

- Used to handle the data for single and multi-dimensional data structure.
- Creates the powerful data frame.
- Data frame is two-dimensional array used to store data whereas Series is one-dimensional array.
- Let's see more practical use of Pandas.

Pandas

Combining data frames

- Merge:merge(left_df, right_df, on='Customer_id', how='inner')



- Concat:

```
frames = [df1, df2]
result = pd.concat(frames)
```

Matplotlib

- Graphical Representation of the values
- To install: **pip install matplotlib**
- To use in program: import matplotlib
- Let's say if we have to populate our previous lists, we can use:

Import matplotlib.pyplot as plt

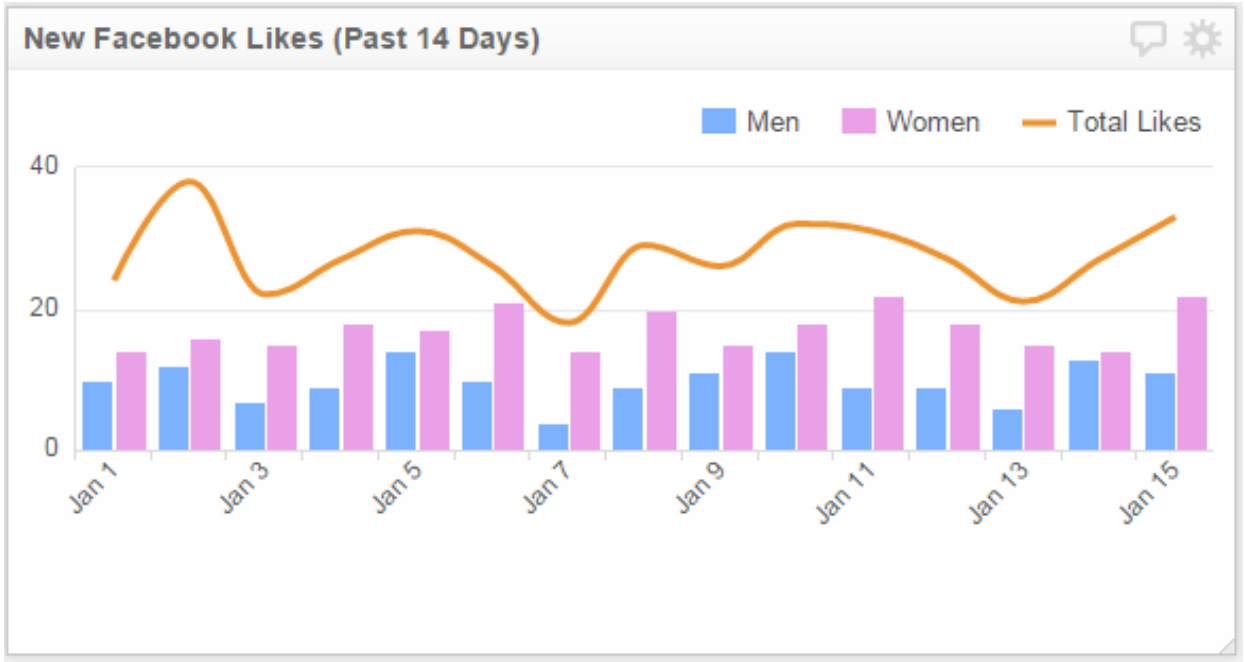
Plt.plot(Convert)

Plt.show()

Matplotlib

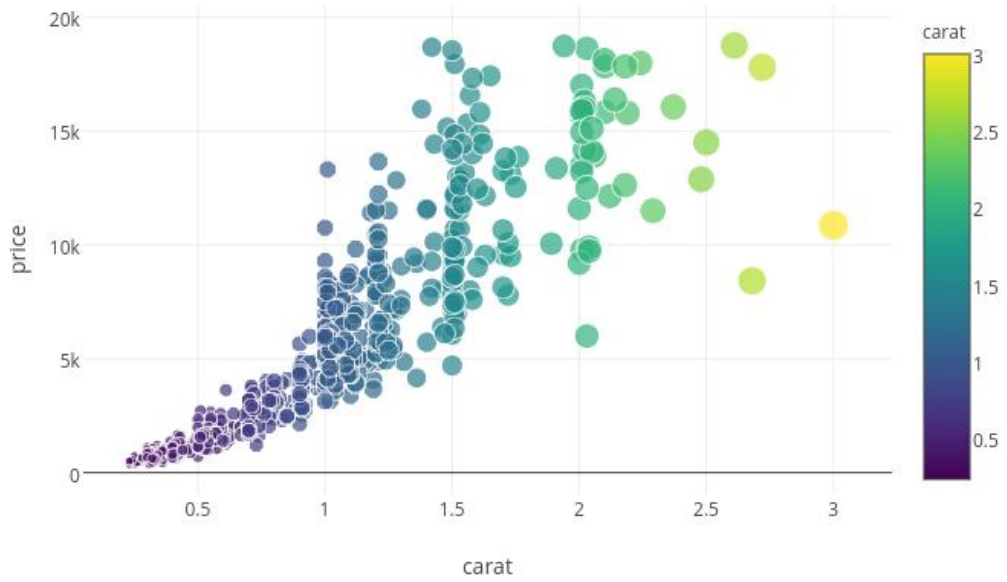
Bar Graph

- Helps to visualize a numeric feature



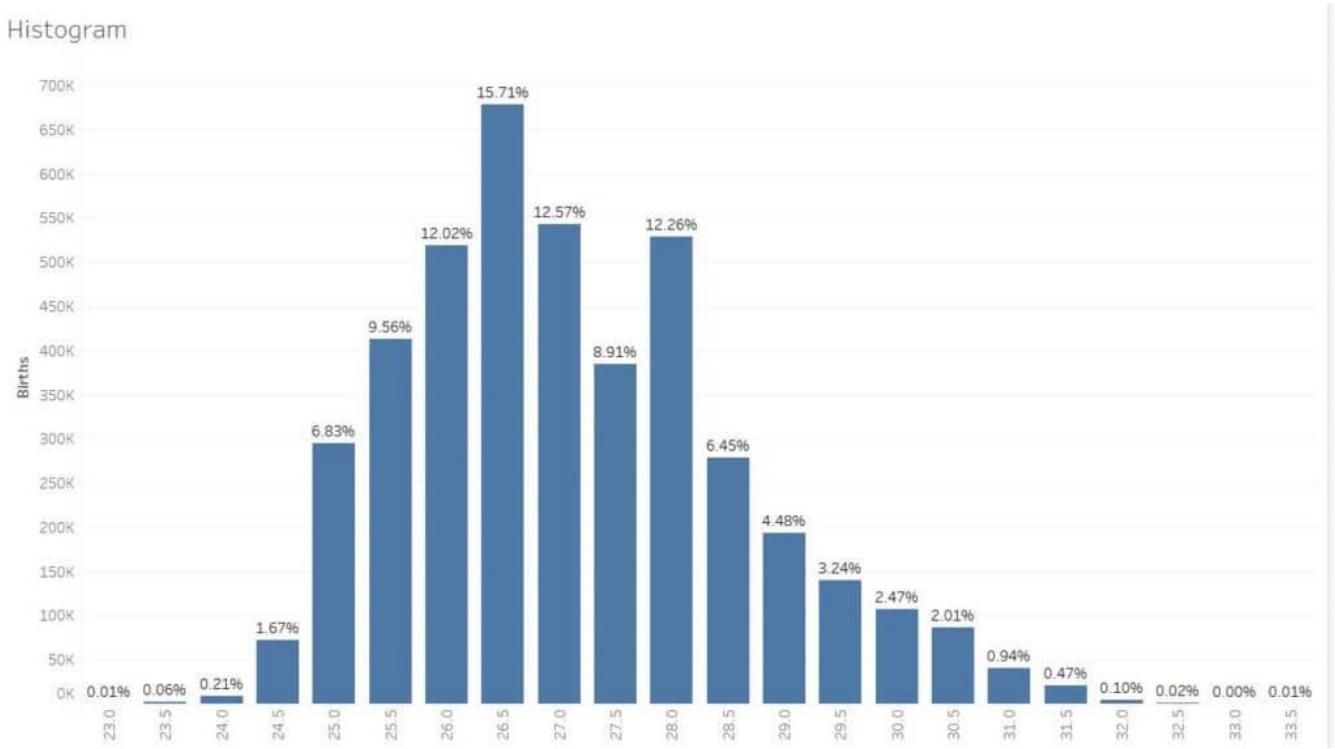
Matplotlib

Scatter Plot



Matplotlib

Histogram

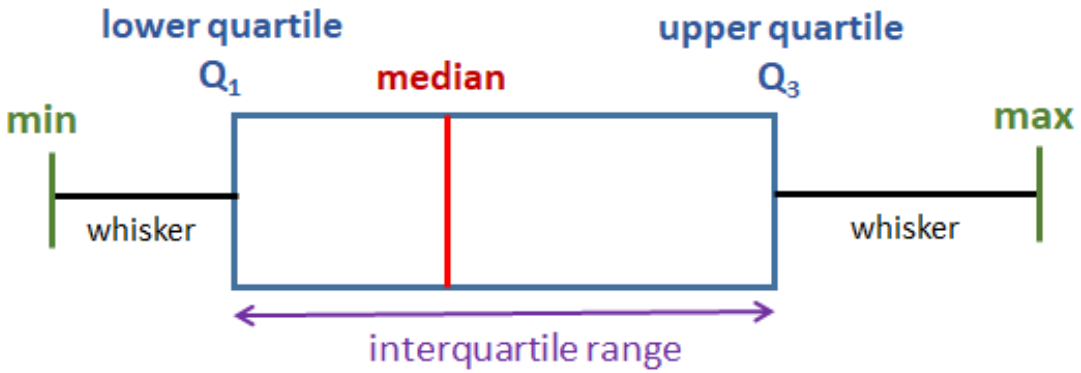


Matplotlib

Box Plot

Box and Whisker Plot

A box and whisker plot (also called a box plot) shows the five-number summary of a set of data: minimum, lower quartile, median, upper quartile, and maximum.



Thank you

39