GLS UNIVERSITY

Faculty of Computer Applications & Information Technology

Integrated MCA Programme

Semester V

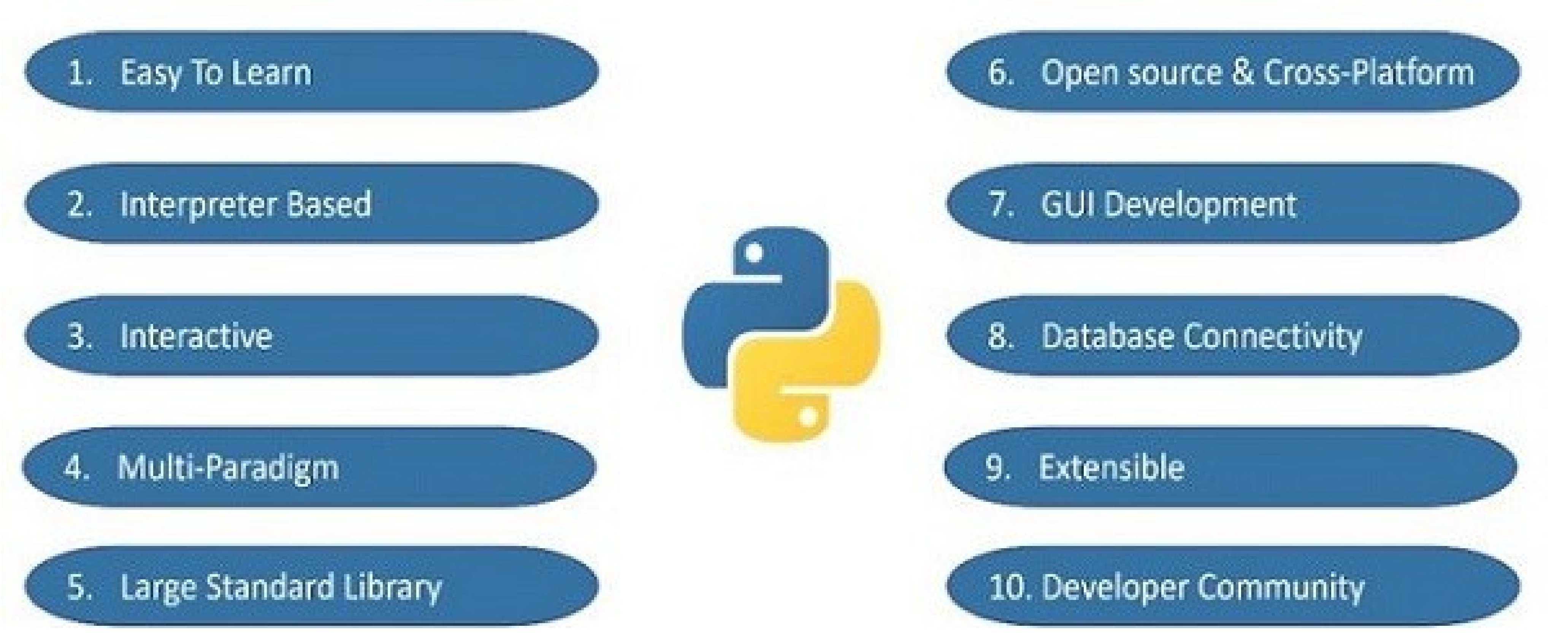
Introduction to Python

Unit 1: Introduction to Python

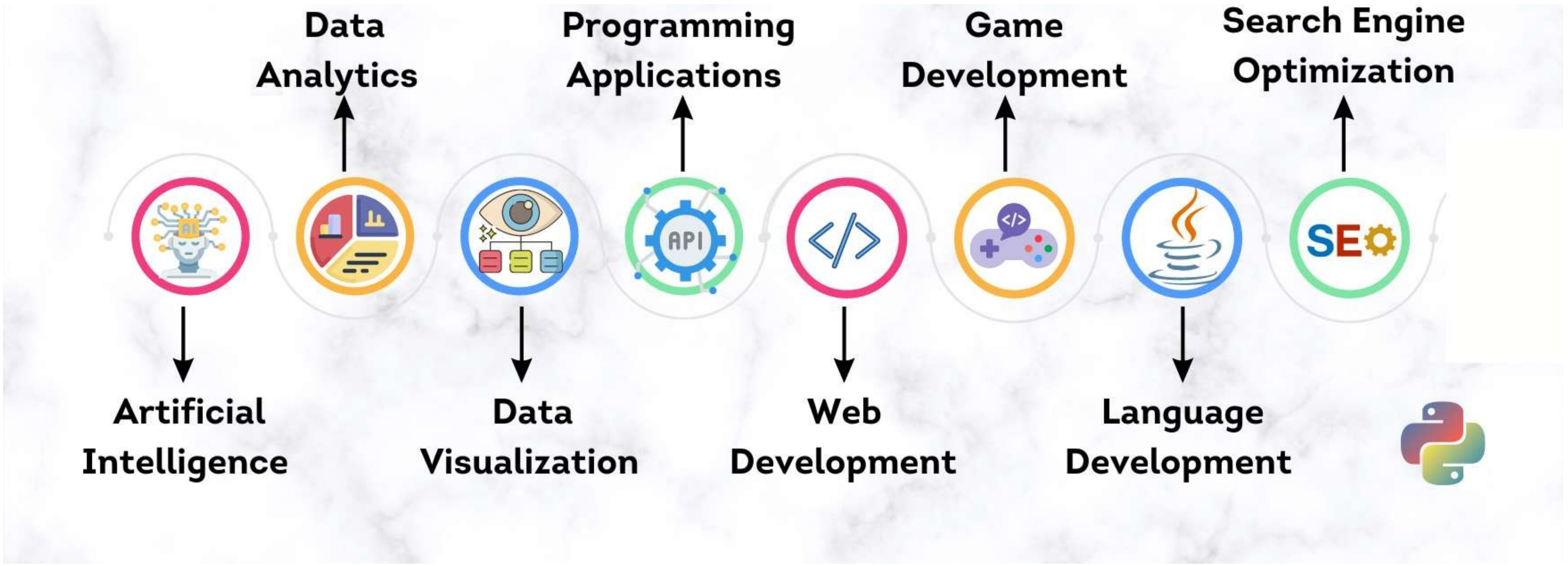
Introduction

- Python laid its foundation in the late 1980s.
- The implementation of Python was started in December 1989 by Guido Van Rossum at CWI in Netherland.
- In February 1991, Guido Van Rossum published the code (labeled version 0.9.0) to alt.sources.
- In 1994, Python 1.0 was released with new features like lambda, map, filter, and reduce.
- Python 2.0 added new features such as list comprehensions, garbage collection systems.
- On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify the fundamental flaw of the language.
- ABC programming language is said to be the predecessor of Python language, which was capable of Exception Handling and interfacing with the Amoeba Operating System.

Features of Python



1. Easy To Learn
2. Interpreter Based
3. Interactive
4. Multi-Paradigm
5. Large Standard Library
6. Open source & Cross-Platform
7. GUI Development
8. Database Connectivity
9. Extensible
10. Developer Community


Artificial Intelligence
Data Analytics
Data Visualization
Programming Applications
Web Development
Game Development
Language Development
Search Engine Optimization
AI
API
</>

Top python Frameworks & Libraries To Use

Django

Pyramid

Bottle

Tornado

Flask

Aiohttp

Pre-commit

Request

Numpy

Pandas

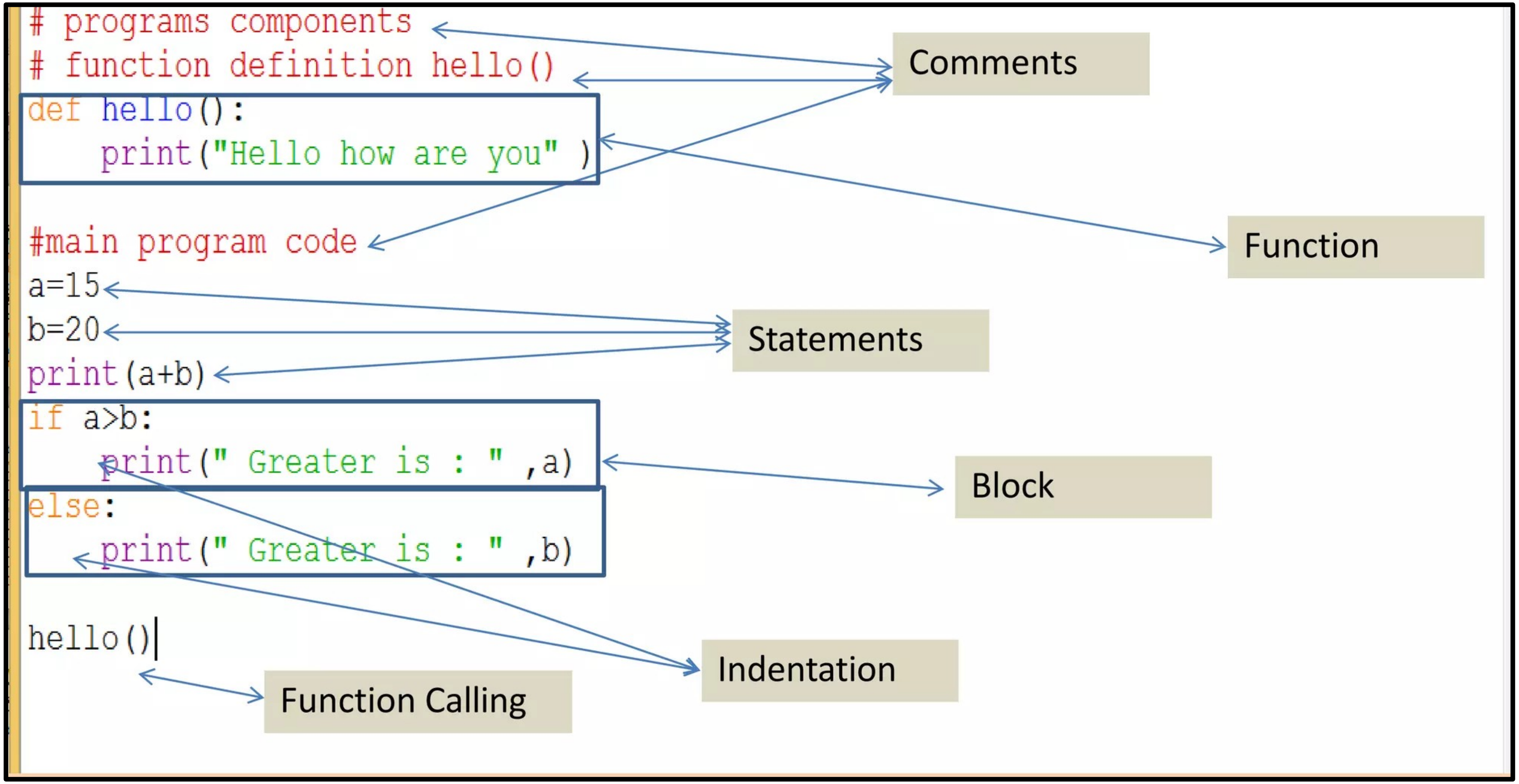
Pillow

Tensorflow

Scikit-learn

Pytorch

Opencv



```
# programs components
# function definition hello()
def hello():
    print("Hello how are you" )

#main program code
a=15
b=20
print(a+b)
if a>b:
    print(" Greater is : " ,a)
else:
    print(" Greater is : " ,b)

hello()
```

Python Programming Mode

Interactive Mode: Used when an user wants to run one single line or one block of code.

Script Mode: Used when an user wants to run multiple line of codes in a text file then save it with a .py extension.

Comments

- Comments can be used to explain Python code.
- Comments can be used to make the code more readable.
- Comments can be used to prevent execution when testing code.
- #

Variables

- Variables are containers for storing data values.
- Variable declaration: variable_name=value
- Eg: x = 5

  a1 = “IMCA”

- Variables do not need to be declared with any particular type, and can even change type after they have been set.

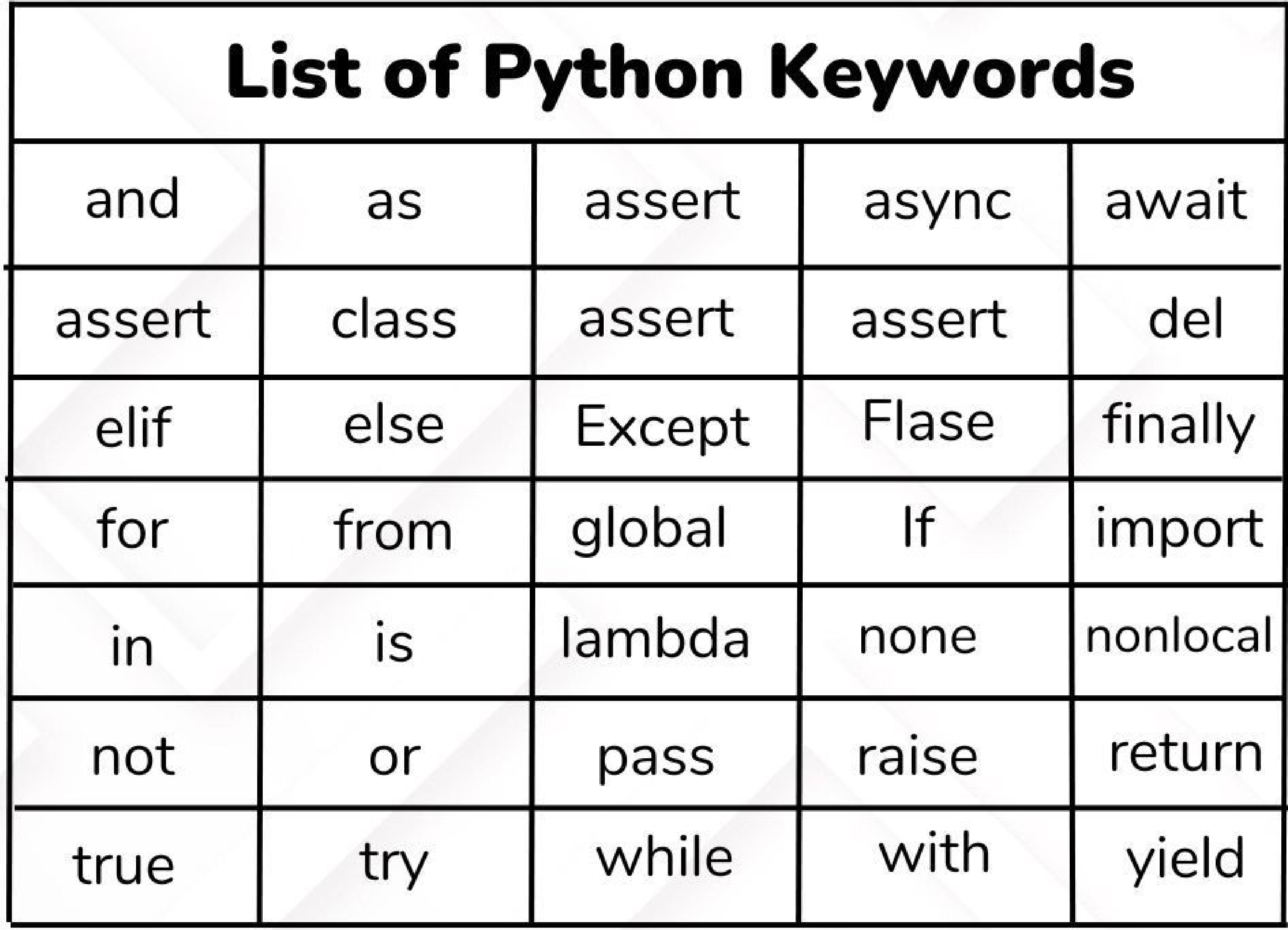
Variables

- A variable can have a short name (like a, b) or a more descriptive name (employee, student_name, total_marks).
- <u>Rules for Python variables:</u>
  - A variable name must start with a letter or the underscore character
  - A variable name cannot start with a number
  - A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
  - Variable names are case-sensitive (age, Age and AGE are three different variables)
  - A variable name cannot be any of the Python keywords.

Casting Datatype in Variables

- If you want to specify the data type of a variable, this can be done with casting.
  - x = str(5)
  - y = int(5)
  - z = float(5)
- You can get the data type of a variable with the type() function.
  - x = 5
  - y = "John"
  - print(type(x))
  - print(type(y))

Keywords



List of Python Keywords

and	as	assert	async	await
assert	class	assert	assert	del
elif	else	Except	Flase	finally
for	from	global	If	import
in	is	lambda	none	nonlocal
not	or	pass	raise	return
true	try	while	with	yield

Constants

- A Constant is a variable whose value cannot be changed throughout the program.
- Rules for Python constants:
  - Python Constants and variable names should contain a combination of lowercase (a-z) or capital (A-Z) characters, numbers (0-9), or an underscore ( ).
  - When using a Constant name, always use UPPERCASE, For example, CONSTANT = 50.
  - The Constant names should not begin with digits.
  - Except for underscore(_), no additional special character (!, #, ∧, @, $) is utilized when declaring a constant.

Literals

- The data which is being assigned to the variables are called as Literal.

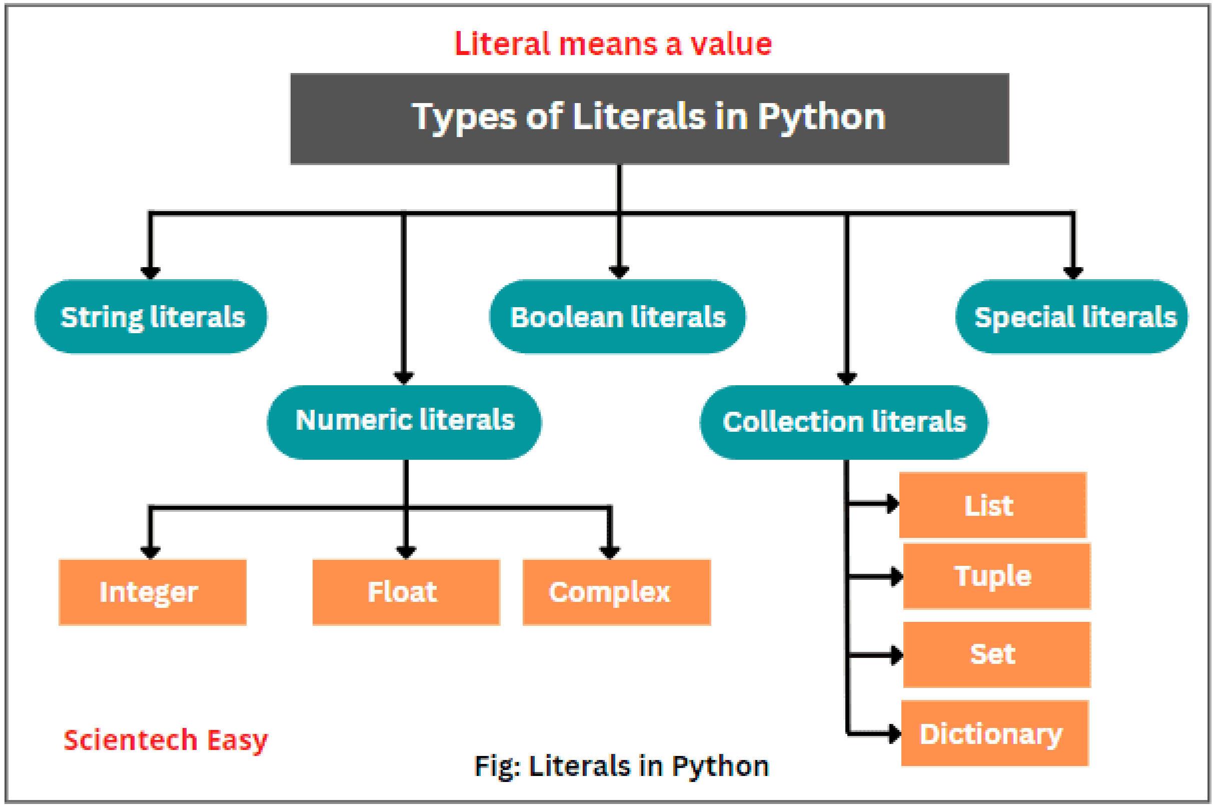
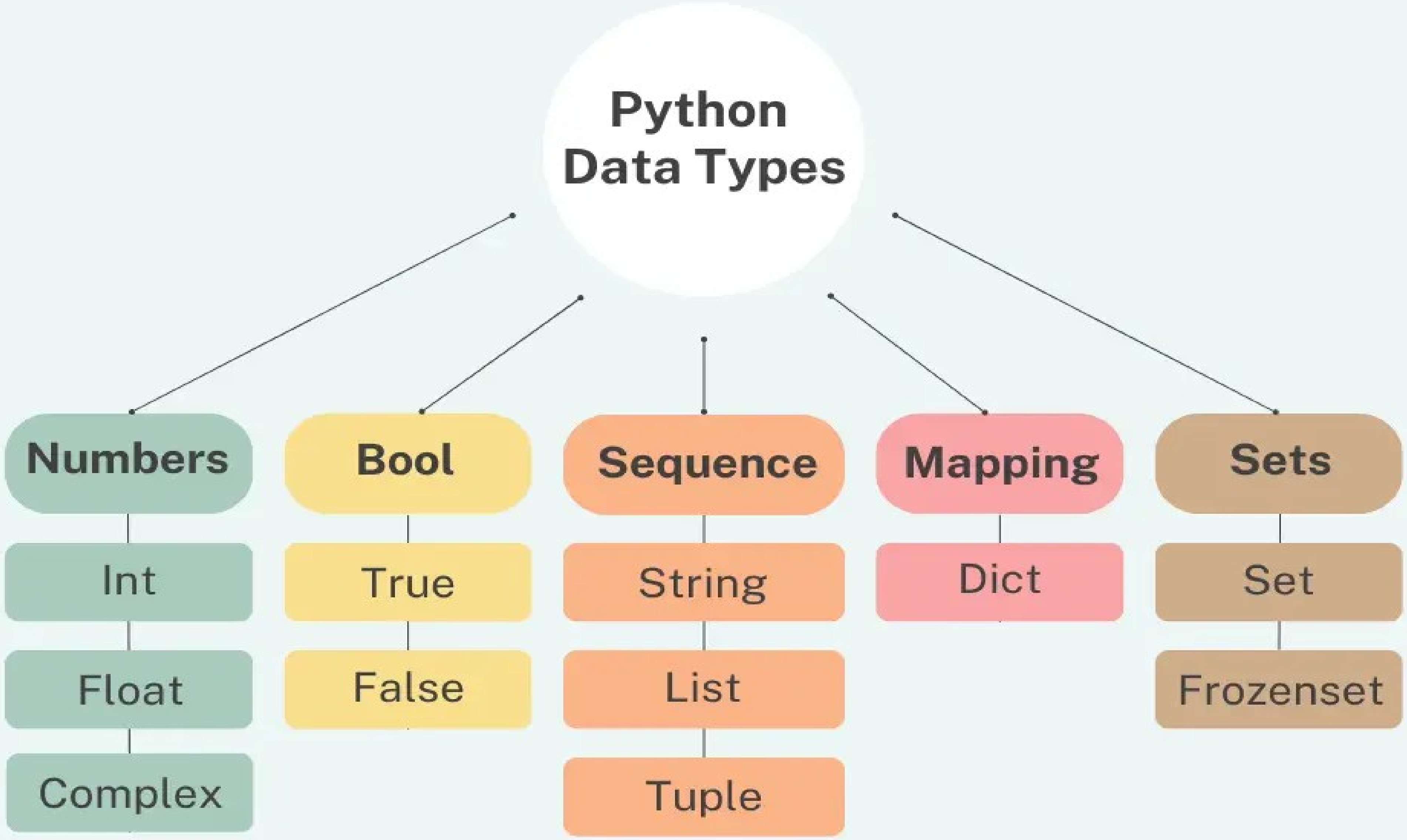


Fig: Literals in Python


Python Data Types
Numbers
Int
Float
Complex
Bool
True
False
Sequence
String
List
Tuple
Mapping
Dict
Sets
Set
Frozenset

Strings

- Strings in python are surrounded by either single quotation marks, or double quotation marks.
- Eg. 'imca' is the same as "imca"
- Individual characters in the strings can be accessed using their index.
- Eg. str=”Gls University”

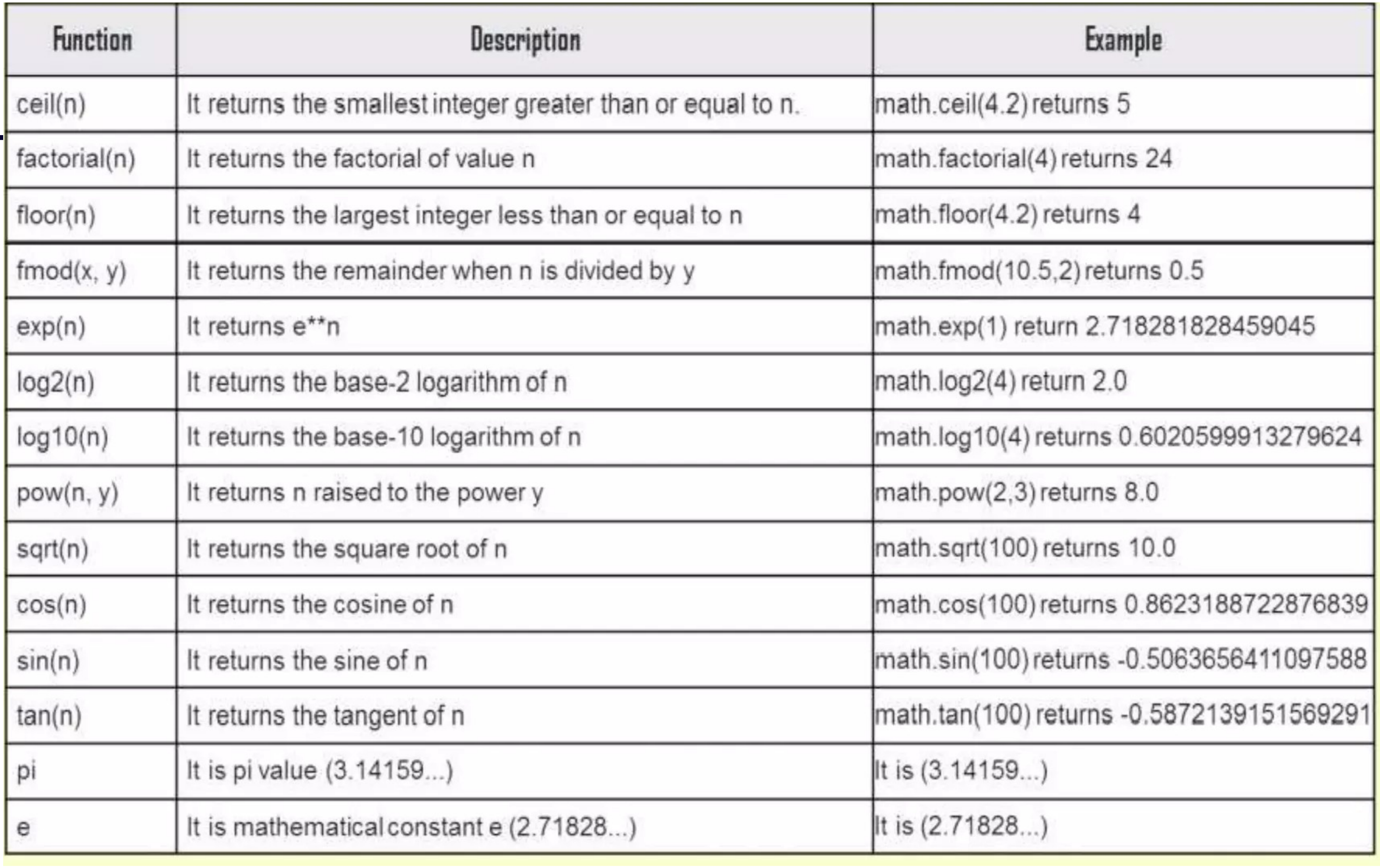
  print str[0] O/p: G

  print str[4] O/p: U

Strings Methods

Method	Description
strip()	removes any whitespace from the beginning or the end
lower()	Returns a string in lower case characters
upper()	Returns a string in uppercase characters
replace()	Replaces a string with another string
split()	Splits the string into sub strings
capitalize()	Capitalizes the first character in the string
count()	Returns no. of occurrences in the string
index()	Returns the index of the character
find()	Gives the index value of the string specified
isalpha()	Returns true if the string has only alphabets
isalnum()	Returns true if the string has both alphabets and numbers
isdigit()	Returns true if the string has only numbers
islower()	Returns true if the string has only lower case characters
isupper()	Returns true if the string has only uppercase characters

Math Functions



Function	Description	Example
ceil(n)	It returns the smallest integer greater than or equal to n.	math.ceil(4.2) returns 5
factorial(n)	It returns the factorial of value n	math.factorial(4) returns 24
floor(n)	It returns the largest integer less than or equal to n	math.floor(4.2) returns 4
fmod(x, y)	It returns the remainder when n is divided by y	math.fmod(10.5,2) returns 0.5
exp(n)	It returns e**n	math.exp(1) return 2.718281828459045
log2(n)	It returns the base-2 logarithm of n	math.log2(4) return 2.0
log10(n)	It returns the base-10 logarithm of n	math.log10(4) returns 0.6020599913279624
pow(n, y)	It returns n raised to the power y	math.pow(2,3) returns 8.0
sqrt(n)	It returns the square root of n	math.sqrt(100) returns 10.0
cos(n)	It returns the cosine of n	math.cos(100) returns 0.8623188722876839
sin(n)	It returns the sine of n	math.sin(100) returns -0.5063656411097588
tan(n)	It returns the tangent of n	math.tan(100) returns -0.5872139151569291
pi	It is pi value (3.14159...)	It is (3.14159...)
e	It is mathematical constant e (2.71828...)	It is (2.71828...)