p

|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| **PYTHON**  BASIC syntax |
| |  |  |  | | --- | --- | --- | | [Author name] | 11/14/23 | **JAHANGIR HUSSEN1** | |

|  |  |
| --- | --- |
| **Python Data type:** | |
| # Integer  num = 42  print(num) | 42 |
| # Float  float\_num = 3.14  print(float\_num) | 3.14 |
| # String  text = "Hello, World!"  print(text) | Hello, World! |
| # Boolean  is\_true = True  print(is\_true) | True |
| # List  my\_list = [1, 2, 3]  print(my\_list) | [1, 2, 3] |
| # Tuple  my\_tuple = (1, 2, 3)  print(my\_tuple) | (1, 2, 3) |
| # Set  my\_set = {1, 2, 3}  print(my\_set) | {1, 2, 3} |
| # Dictionary  my\_dict = {"key": "value", "num": 42}  print(my\_dict) | {'key': 'value', 'num': 42} |
| # NoneType  none\_var = None  print(none\_var) | None |
| # Complex  complex\_var = 3 + 4j  print(complex\_var) | (3+4j) |
| # Bytes  byte\_data = b'hello'  print(byte\_data) | b'hello' |
| # Bytearray  bytearray\_data = bytearray([65, 66, 67])  print(bytearray\_data) | bytearray(b'ABC') |
| # Range  range\_var = range(5)  print(list(range\_var)) | [0, 1, 2, 3, 4] |
| # FrozenSet  frozen\_set = frozenset({1, 2, 3})  print(frozen\_set) | frozenset({1, 2, 3}) |
| # Bytes (with Unicode)  unicode\_bytes = b'hello'.decode('utf-8')  print(unicode\_bytes) | hello |
|  |  |

|  |  |
| --- | --- |
| **Python operator:** | |
| # Addition  result\_addition = 10 + 5  print(result\_addition) | 15 |
| # Subtraction  result\_subtraction = 10 - 5  print(result\_subtraction) | 5 |
| # Multiplication  result\_multiplication = 10 \* 5  print(result\_multiplication) | 50 |
| # Division  result\_division = 10 / 5  print(result\_division) | 2.0 |
| # Modulus (Remainder)  result\_modulus = 10 % 3  print(result\_modulus) | 1 |
| # Equal to  result\_equal = (10 == 5)  print(result\_equal) | False |
| # Not equal to  result\_not\_equal = (10 != 5)  print(result\_not\_equal) | True |
| # Greater than  result\_greater\_than = (10 > 5)  print(result\_greater\_than) | True |
| # Less than  result\_less\_than = (10 < 5)  print(result\_less\_than) | False |
| # Greater than or equal to  result\_greater\_equal = (10 >= 5)  print(result\_greater\_equal) | True |
| # Less than or equal to  result\_less\_equal = (10 <= 5)  print(result\_less\_equal) | False |
| # Logical AND  result\_logical\_and = True and False  print(result\_logical\_and) | False |
| # Logical OR  result\_logical\_or = True or False  print(result\_logical\_or) | True |
| # Logical NOT  result\_logical\_not = not True  print(result\_logical\_not) | False |

|  |  |  |
| --- | --- | --- |
| **Python conditional Statements:** | | |
| # If statement  x = 10  if x > 5:      print("x is greater than 5") | | x is greater than 5 |
| # If-else statement  y = 3  if y % 2 == 0:      print("y is even") | else:      print("y is odd") | y is odd |
| # If-elif-else statement  z = 15  if z > 10:      print("z is greater than 10")  elif z == 10:      print("z is equal to 10")  else:      print("z is less than 10") | | z is greater than 10 |

|  |  |
| --- | --- |
| **Python loop statements:** | |
| # For loop with a list  fruits = ["apple", "banana", "cherry"]  for fruit in fruits:      print(fruit) | apple  banana  cherry |
| # For loop  for j in range(3):      print(j) | 0  1  2 |
| # While loop  i = 0  while i < 5:      print(i)      i += 1 | 0  1  2  3    4 |

|  |  |
| --- | --- |
| **Python input/output:** | |
| # Input from the user  user\_input = input("Enter a number: ")  print("You entered:", user\_input) | 42 |
| # Taking multiple inputs from the user  name = input("Enter your name: ")  age = input("Enter your age: ")  print(f"Hello, {name}! You are {age} years old.") | Enter your name:Alice  Enter your age: 25  Hello, Alice! You are 25 years old. |
| # Printing output  print("Hello, World!") | # Output: Hello, World! |
| # Formatted Output  name = "Alice"  age = 30  print(f"My name is {name} and I am {age} years old.") | My name is Alice and I am 30 years old. |

|  |  |
| --- | --- |
| **Python strings:** | |
| # Slicing  message = "Hello, World!"  print(message[7:]) | World! |
| # Length of a String  text = "Hello, World!"  length = len(text)  print(length) | 13 |
| # Lowercase  text = "Hello, World!"  lower\_text = text.lower()  print(lower\_text) | hello, world! |
| # Uppercase  text = "Hello, World!"  upper\_text = text.upper()  print(upper\_text) | HELLO, WORLD! |
| # Capitalize  text = "hello, world!"  capitalized\_text = text.capitalize()  print(capitalized\_text) | Hello, world! |
| # Find  text = "Hello, World!"  index\_world = text.find("World")  print(index\_world)  # Replace  text = "Hello, World!"  new\_text = text.replace("World", "Universe")  print(new\_text) | 7  Hello, Universe! |
| # Strip  text = "   Hello, World!   "  stripped\_text = text.strip()  print(stripped\_text) | Hello, World! |
| # Split  text = "apple,orange,banana"  fruits\_list = text.split(",")  print(fruits\_list) | ['apple', 'orange', 'banana'] |
| # Startswith  text = "Hello, World!"  starts\_with\_hello = text.startswith("Hello")  print(starts\_with\_hello) | True |
| # Endswith  text = "Hello, World!"  ends\_with\_world = text.endswith("World!")  print(ends\_with\_world) | True |
| # isalpha  text = "Hello"  is\_alpha = text.isalpha()  print(is\_alpha) | True |
| # isdigit  number = "123"  is\_digit = number.isdigit()  print(is\_digit) | True |
| # isspace  whitespace = "   "  is\_space = whitespace.isspace()  print(is\_space) | True |
| # Join  fruits = ["apple", "orange", "banana"]  joined\_fruits = ", ".join(fruits)  print(joined\_fruits) | apple, orange, banana |
| # Center  text = "Hello"  centered\_text = text.center(10, "\*")  print(centered\_text) | \*\*Hello\*\*\* |
| # Left Strip and Right Strip  text = "   Hello, World!   "  left\_stripped\_text = text.lstrip()  right\_stripped\_text = text.rstrip()  print(left\_stripped\_text  print(right\_stripped\_text) | Hello, World!     Hello, World! |
| # Swapcase  text = "Hello, World!"  swapped\_case\_text = text.swapcase()  print(swapped\_case\_text) | hELLO, wORLD! |
| # Format  name = "Alice"  age = 30  formatted\_text = "My name is {} and I am {} years old.".format(name, age)  print(formatted\_text) | My name is Alice and I am 30 years old. |
| # Sorting  text = "programming"  sorted\_chars = sorted(text)  print(sorted\_chars) | ['a', 'g', 'g', 'i', 'm', 'm', 'n', 'o', 'p', 'r'] |
| # Min and Max  text = "programming"  min\_char = min(text)  max\_char = max(text)  print(min\_char)  print(max\_char) | a  r |
| # In  text = "PythonProgramming"  contains\_python = "Python" in text  print(contains\_python) | True |

|  |  |
| --- | --- |
| **Python math method/function:=import math** | |
| # Square Root  num\_sqrt = math.sqrt(25)  print("Square Root of 25:", num\_sqrt) | Square Root of 25: 5.0 |
| # Power  num\_power = math.pow(2, 3)  print("2 raised to the power of 3:", num\_power) | 2 raised to the power of 3: 8.0 |
| # Absolute Value  absolute\_value = math.fabs(-10.5)  print("Absolute Value of -10.5:", absolute\_value) | Absolute Value of -10.5: 10.5 |
| # Ceil (smallest integer greater than or equal to x)  ceil\_value = math.ceil(4.2)  print("Ceil of 4.2:", ceil\_value) | Ceil of 4.2: 5 |
| # Floor (largest integer less than or equal to x)  floor\_value = math.floor(4.8)  print("Floor of 4.8:", floor\_value) | Floor of 4.8: 4 |
| # Trigonometric Functions (in radians)  sin\_value = math.sin(math.radians(30))  print("Sin of 30 degrees:", sin\_value)  cos\_value = math.cos(math.radians(45))  print("Cos of 45 degrees:", cos\_value)  tan\_value = math.tan(math.radians(60))  print("Tan of 60 degrees:", tan\_value) | Sin of 30 degrees: 0.49999999999999994  Cos of 45 degrees: 0.7071067811865475  Tan of 60 degrees: 1.7320508075688772 |
| # Logarithmic Functions  log\_value = math.log(100, 10)  print("Log base 10 of 100:", log\_value) | Log base 10 of 100: 2.0 |
| # Constants  print("Value of pi:", math.pi)  print("Value of e (Euler's number):", math.e) | Value of pi: 3.141592653589793  Value of e (Euler's number): 2.718281828459045 |
|  |  |