



INDEX

1) Language Fundamentals	1
2) Operators	32
3) Flow Control	56
4) String Data Type	72
5) List Data Structure	98
6) Tuple Data Structure	116
7) Set Data Structure	124
8) Dictionary Data Structure	131
9) Functions	142
10) Modules	162
11) Packages	173
12) 100 Pattern Programs	177



DETAILED INDEX

1) LANGUAGE FUNDAMENTALS 1

❖ Introduction	2
❖ Features of Python	4
1) Simple and Easy to Learn	
2) Freeware and Open Source	
3) High Level Programming Language	
4) Platform Independent	
5) Portability	
6) Dynamically Typed	
7) Both Procedure Oriented and Object Oriented	
8) Interpreted	
9) Extensible	
10) Embedded	
11) Extensive Library	
❖ Limitations of Python	5
❖ Flavors of Python	6
1) CPython	
2) Jython OR JPython	
3) IronPython	
4) PyPy	
5) RubyPython	
6) AnacondaPython	
❖ Python Versions	6
❖ Identifiers	7
❖ Reserved Words	9



❖ DATA TYPES	10
1) int Data Type	
• Decimal Form	
• Binary Form	
• Octal Form	
• Hexa Decimal Form	
2) Float Data Type	
3) Complex Data Type	
4) bool Data Type	
5) str Data Type	
6) bytes Data Type	
7) bytearray Data Type	
8) List Data Type	
9) Tuple Data Type	
10) Range Data Type	
11) Set Data Type	
12) frozenset Data Type	
13) dict Data Type	
14) None Data Type	
❖ Base Conversions	12
❖ Slicing of Strings	16
❖ TYPE CASTING	18
• int()	
• float()	
• complex()	
• bool()	
• str()	
❖ Fundamental Data Types vs Immutability	21
❖ Escape Characters	31
❖ Constants	31

2) OPERATORS 32

1) Arithmetic Operators	33
2) Relational Operators OR Comparison Operators	35
3) Equality Operators	36



4) Logical Operators	37
5) Bitwise Operators	38
• Bitwise Complement Operator (~)	
6) Shift Operators	39
• << Left Shift Operator	
• >> Right Shift Operator	
7) Assignment operators	40
8) Ternary Operator OR Conditional Operator	41
9) Special operators	42
• Identity Operators	
• Membership operators	
☯ Operator Precedence	44
☯ Mathematical Functions (math Module)	45
☯ Command Line Arguments	50
☯ Output Statements	52
3) FLOW CONTROL	56
☯ Conditional Statements	57
• if	
• if-elif	
• if-elif-else	
☯ Iterative Statements	62
• for	
• while	
☯ Transfer Statements	66
• break	
• continue	
• pass	
☯ Loops with else Block	68
☯ del Statement	70



☯ Difference between del and None 71

4) STRING DATA TYPE 72

☯ What is String?	73
☯ How to define multi-line String Literals?	73
☯ How to Access Characters of a String?	74
• Accessing Characters By using Index	
• Accessing Characters by using Slice Operator	
☯ Behaviour of Slice Operator	75
☯ Slice Operator Case Study	76
☯ Mathematical Operators for String	76
☯ len() in-built Function	77
☯ Checking Membership	78
☯ Comparison of Strings	78
☯ Removing Spaces from the String	79
• rstrip()	
• lstrip()	
• strip()	
☯ Finding Substrings	79
☯ Counting substring in the given String	81
☯ Replacing a String with another String	82
☯ Splitting of Strings	83
☯ Joining of Strings	83
☯ Changing Case of a String	84
☯ Checking Starting and Ending Part of the String	84
• s.startswith(substring)	
• s.endswith(substring)	
☯ To Check Type of Characters Present in a String	85
☯ Formatting the Strings	86



- ⚙ Important Programs regarding String Concept 87
- 1) Program to Reverse the given String
 - 2) Program to Reverse Order of Words
 - 3) Program to Reverse Internal Content of each Word
 - 4) Program to Print Characters at Odd Position and Even Position for the given String
 - 5) Program to Merge Characters of 2 Strings into a Single String by taking Characters alternatively
 - 6) Program to Sort the Characters of the String and First Alphabet Symbols followed by Numeric Values
 - 7) Program for the following Requirement (Input: a4b3c2, Output: aaaabbbcc)
 - 8) Program to perform the following Activity (Input: a4k3b2, Outpt: aeknbd)
 - 9) Program to Remove Duplicate Characters from the given Input String
 - 10) Program to find the Number of Occurrences of each Character present in the given String
 - 11) Program to perform the following Task
 - Input: 'one two three four five six seven'
 - Output: 'one owt three ruof five xis seven'

- ⚙ Formatting the Strings 92

5) LIST DATA STRUCTURE 98

- ⚙ Creation of List Objects 99
- ⚙ Accessing Elements of List 100
- By using Index
 - By using Slice Operator
- ⚙ List vs Mutability 102
- ⚙ Traversing the Elements of List 102
- By using while Loop
 - By using for Loop
 - To display only Even Numbers
 - To display Elements by Index wise



⚽ Important Functions of List	104
☕ To get Information about List	
• len()	
• count()	
• index()	
☕ Manipulating Elements of List	
• append()	
• insert()	
• extend()	
• remove()	
• pop()	
☕ Ordering Elements of List	
• reverse()	
• sort()	
⚽ Using Mathematical Operators for List Objects	111
• Concatenation Operator (+)	
• Repetition Operator (*)	
⚽ Comparing List Objects	111
⚽ Membership Operators	112
• in Operator	
• not in Operator	
⚽ clear() Function	112
⚽ Nested Lists	113
⚽ Nested List as Matrix	113
⚽ List Comprehensions	114



6) TUPLE DATA STRUCTURE 116

- ⊗ Tuple Creation 118
- ⊗ Accessing Elements of Tuple
 - By using Index
 - By using Slice Operator
- ⊗ Tuple vs Immutability 119
- ⊗ Mathematical Operators for Tuple 119
 - Concatenation Operator (+)
 - Multiplication Operator OR Repetition Operator (*)
- ⊗ Important Functions of Tuple 120
 - len()
 - count()
 - index()
 - sorted()
 - min() And max()
 - cmp()
- ⊗ Tuple Packing and Unpacking 121
- ⊗ Tuple Comprehension 122
- ⊗ Differences between List and Tuple 123

7) SET DATA STRUCTURE 124

- ⊗ Creation of Set Objects 125
- ⊗ Important Functions of Set 126
 - add(x)
 - update(x,y,z)
 - copy()
 - pop()
 - remove(x)
 - discard(x)
 - clear()



⚙ Mathematical Operations on the Set	128
• union()	
• intersection()	
• difference()	
• symmetric_difference()	
⚙ Membership Operators: (in, not in)	129
⚙ Set Comprehension	129

8) DICTIONARY DATA STRUCTURE 131

⚙ How to Create Dictionary?	132
⚙ How to Access Data from the Dictionary?	132
⚙ How to Update Dictionaries?	134
⚙ How to Delete Elements from Dictionary?	134
• del d[key]	
• d.clear()	
• del d	
⚙ Important Functions of Dictionary	135
• dict()	
• len()	
• clear()	
• get()	
• pop()	
• popitem()	
• keys()	
• values()	
• items()	
• copy()	
• setdefault()	
• update()	
⚙ Dictionary Comprehension	141



9) FUNCTIONS 142

- ⊗ Built in Functions 143
- ⊗ User Defined Functions 143
- ⊗ Parameters 144
- ⊗ Return Statement 144
- ⊗ Returning Multiple Values from a Function 146

- ⊗ Types of Arguments 147
 - Positional Arguments
 - Keyword Arguments
 - Default Arguments
 - Variable Length Arguments

- ⊗ Case Study 151

- ⊗ Types of Variables 152
 - Global Variables
 - Local Variables

- ⊗ global Keyword 153
- ⊗ Recursive Functions 154
- ⊗ Anonymous Functions 155
- ⊗ Normal Function 155
- ⊗ Lambda Function 155
- ⊗ filter() Function 156
- ⊗ map() Function 156
- ⊗ reduce() Function 158

- ⊗ Everything is an Object 159
- ⊗ Function Aliasing 159
- ⊗ Nested Functions 160



10) MODULES 162

- ⊗ Renaming a Module at the time of import (Module Aliasing) 164
- ⊗ from ... import 164
- ⊗ Various Possibilities of import 164
- ⊗ Member Aliasing 165
- ⊗ Reloading a Module 165
- ⊗ Finding Members of Module by using dir() Function 166
- ⊗ The Special Variable __name__ 168
- ⊗ Working with math Module 169
- ⊗ Working with random Module 169
 - random() Function
 - randint() Function
 - uniform() Function
 - randrange ([start], stop, [step])
 - choice() Function

11) PACKAGES 173

12) 100 PATTERN PROGRAMS 177