

Summary of Python Basic

1.	Data Types	<p>1. Numbers</p> <ul style="list-style-type: none"> a) Integers b) Floating Point c) Complex Numbers <p>2. Strings</p> <p>3. List – sequence enclosed in []</p> <p>4. Tuples – sequence enclosed in ()</p> <p>5. Dictionary- sequence (contains key:value pair) enclosed in { }</p>
2.	Mutable Data Types	List , Dictionary
	Immutable Data Types	Integers, Float, Booleans, strings, tuples
3	TOKENS (individual unit of	<p>1. IDENTIFIERS (name given to memory location)</p> <p>2. LITERALS (constant value that can not be changed)</p> <ul style="list-style-type: none"> a. "xyz" – String literal b. 123- int literal c. 34.5 – float literal d. True/ False – Boolean literal e. 3+6j – Complex literal <p>3. OPERATORS (helps to operates upon operands)</p> <p>4. KEYWORDS (reserved words with specific purpose)</p> <p>5. PUNCTUATORS (, : " etc)</p>
4.	Operators	<p>1. Arithmetic (+,-,/,*,//,%,**)</p> <ul style="list-style-type: none"> a. Unary vs Binary b. Augmented Assignment Operators(+=,-=,*=,/=,//=,%=,**=) <p>2. Relational Operators (<,<=,>,>=,==,!=)</p> <p>3. Identity Operators (is, is not)</p> <p>= = vs is operator</p> <p>4. Logical Operators(and,or,not)</p> <p>5. Bitwise Operator(& , , ^ , ~)</p>

5	Operator Precedence	Evaluation order of Expressions																												
	<table border="1"> <thead> <tr> <th data-bbox="321 260 625 338">Operators</th> <th data-bbox="625 260 1377 338">Meaning</th> </tr> </thead> <tbody> <tr> <td data-bbox="321 338 625 415">()</td> <td data-bbox="625 338 1377 415">Parentheses</td> </tr> <tr> <td data-bbox="321 415 625 493">**</td> <td data-bbox="625 415 1377 493">Exponent</td> </tr> <tr> <td data-bbox="321 493 625 571">+x, -x, ~x</td> <td data-bbox="625 493 1377 571">Unary plus, Unary minus, Bitwise NOT</td> </tr> <tr> <td data-bbox="321 571 625 648">*, /, //, %</td> <td data-bbox="625 571 1377 648">Multiplication, Division, Floor division, Modulus</td> </tr> <tr> <td data-bbox="321 648 625 726">+, -</td> <td data-bbox="625 648 1377 726">Addition, Subtraction</td> </tr> <tr> <td data-bbox="321 726 625 804"><<, >></td> <td data-bbox="625 726 1377 804">Bitwise shift operators</td> </tr> <tr> <td data-bbox="321 804 625 882">&</td> <td data-bbox="625 804 1377 882">Bitwise AND</td> </tr> <tr> <td data-bbox="321 882 625 959">^</td> <td data-bbox="625 882 1377 959">Bitwise XOR</td> </tr> <tr> <td data-bbox="321 959 625 1037"> </td> <td data-bbox="625 959 1377 1037">Bitwise OR</td> </tr> <tr> <td data-bbox="321 1037 625 1115">==, !=, >, >=, <, <=, is, is</td> <td data-bbox="625 1037 1377 1115">Comparisons, Identity, Membership operators</td> </tr> <tr> <td data-bbox="321 1115 625 1192">not</td> <td data-bbox="625 1115 1377 1192">Logical NOT</td> </tr> <tr> <td data-bbox="321 1192 625 1270">and</td> <td data-bbox="625 1192 1377 1270">Logical AND</td> </tr> <tr> <td data-bbox="321 1270 625 1318">or</td> <td data-bbox="625 1270 1377 1318">Logical OR</td> </tr> </tbody> </table>	Operators	Meaning	()	Parentheses	**	Exponent	+x, -x, ~x	Unary plus, Unary minus, Bitwise NOT	*, /, //, %	Multiplication, Division, Floor division, Modulus	+, -	Addition, Subtraction	<<, >>	Bitwise shift operators	&	Bitwise AND	^	Bitwise XOR		Bitwise OR	==, !=, >, >=, <, <=, is, is	Comparisons, Identity, Membership operators	not	Logical NOT	and	Logical AND	or	Logical OR	
Operators	Meaning																													
()	Parentheses																													
**	Exponent																													
+x, -x, ~x	Unary plus, Unary minus, Bitwise NOT																													
*, /, //, %	Multiplication, Division, Floor division, Modulus																													
+, -	Addition, Subtraction																													
<<, >>	Bitwise shift operators																													
&	Bitwise AND																													
^	Bitwise XOR																													
	Bitwise OR																													
==, !=, >, >=, <, <=, is, is	Comparisons, Identity, Membership operators																													
not	Logical NOT																													
and	Logical AND																													
or	Logical OR																													
6.	Conversion Functions (convert one data type to another)	Int(), float(), complex(), str(), bool()																												
	Other important functions	id() – to find address of object type() - to check data type of an object len() – to find length of on object (total characters in object)																												

MATHEMATICAL FUNCTIONS

Function	Returns (description)
abs(x)	The absolute value of x: the (positive) distance between x and zero.
ceil(x)	The ceiling of x: the smallest integer not less than x
cmp(x, y)	-1 if $x < y$, 0 if $x == y$, or 1 if $x > y$
exp(x)	The exponential of x: e^x
fabs(x)	The absolute value of x.
floor(x)	The floor of x: the largest integer not greater than x
log(x)	The natural logarithm of x, for $x > 0$
log10(x)	The base-10 logarithm of x for $x > 0$
max(x1, x2,...)	The largest of its arguments: the value closest to positive infinity
min(x1, x2,...)	The smallest of its arguments: the value closest to negative infinity
modf(x)	The fractional and integer parts of x in a two-item tuple. Both parts have the same sign as x. The integer part is returned as a float.
pow(x, y)	The value of x^y .
round(x [,n])	x rounded to n digits from the decimal point. Python rounds away from zero as a tie-breaker: round(0.5) is 1.0 and round(-0.5) is -1.0.
sqrt(x)	The square root of x for $x > 0$

7.	Expressions	Python expression is any part of the python code that results to a value
8.	Python Statements (Conditional & ITERATIVE Statements)	<ol style="list-style-type: none"> 1. Empty Statement : Pass 2. Simple Statements (Single line of statements) 3. Compound Statements (block of statements)
9.	Statement flow control	<ol style="list-style-type: none"> 1. Sequence 2. Selection <ol style="list-style-type: none"> a. If b. If else c. If elif d. Nested if 3. Iteration statement <ol style="list-style-type: none"> a. While b. For
10.	Program logic development tool	<ol style="list-style-type: none"> 1. Algorithm 2. Flowchart 3. Pseudo code 4. Decision Tree