Basic Python Cheat Sheet

Marina Hermaningsih

Basic String Method

Method	Purpose	Variabel	Example	
			In[ ]	Out[ ]
len()	Count number of string character	country = 'Indonesia'	print(len(country))	9
Notation[]	Know string character by specified index		print(country[0])	I
			print(country[2])	d
Concatenation	To combine two or more strings with + operator	province_1 = 'east' province_2 = 'java'	full_province = province_1 + ' ' + province_2 print(full_province)	east java
.upper()	Converts a string into upper case	province_1 = 'east' province_2 = 'java'	print(province_1.upper())	EAST
.lower()	Converts a string into lower case	city = 'SURABAYA'	print(city.lower())	surabaya
.tittle()	Converts the first character of each word to upper case	film = 'infinity war'	print(film.title())	Infinity War
.strip() **.lstrip()** **.rstrip()**	Remove spaces from left and right side of character	fruit = ' Apel '	print(fruit.strip())	Apel
.find()	Know sequence of character	province_1 = 'east' province_2 = 'java'	print(province_1.find('e'))	0
.replace (' ', ' ')	Change specified character with new specified character	string_1 = 'haha'	print(string_1.replace('a','i'))	hihi

Arithmetic Operation

Operation		Algebraic Expression	Example		Level of Operator Precedence	Rule of Operator Precedence
			In[ ]	Out[ ]		
Addition	+	x + y	7+5	12	1. Parentheses 2. Exponentiation 3. Multiplication, division, modulo 4. Addition and subraction	1. Expression in parentheses evaluate first. 2. Expression contains several (**). Python applies them from right to left. 3. Expression contains several (*),(/),(//),(%). Python applies them from left to right. 4. Expression contains several (+), (-). Python applies them from left to right.
Subtraction	-	x - y	7-5	2		
Multiplication	*	x * y	7*5	35		
Exponentiation	**	x ** y	2**3	8		
			9**(1/2)	3.0		
True Division	/	x/y or x:y	10/4	2.5		
Floor Division	//	[x/y] or [x:y]	10//4	2		
Modulo	%	x mod y	17%5	2		

Assignment Operation

Operation		Example	
		Arithmetic	Assignment
Addition	+=	In[] x=10 x=x%4 print(x) Out[] 2	In[ ] x=10 x%=4 print(x) Out[] 2
Subtraction	-=		
Multiplication	*=		
Exponentiation	**=		
True Division	/=		
Floor Division	//=		
Modulo	%=		

Comparison Operation

Algebraic Operator	Python Condition	Example	
		In[]	Out[]
>	x > y	#1 x = 9%3==0 x	#1 True
<	x < y		
≥	x >= y		
≤	x <= y	#2 y = 10<=2 y	#2 False
=	x == y		
≠	x != y		

Basic Set Method

Method	Purpose	Variabel	Example	
			In[ ]	Out[ ]
.add()	Add new item in set	object={('black','white'),'Indonesia', 21}	object.add(('bag','shoes')) object	{('bag', 'shoes'), ('black', 'white'), 21, 'Indonesia'}
.update()	Update set		object.update([5,97],('orange',97)) object	{('bag', 'shoes'), ('black', 'white'), 21, 5, 97, 'Indonesia', 'orange'}
.remove()	Delete specified item of set		object.remove('orange') object	{('bag', 'shoes'), ('black', 'white'), 21, 5, 97, 'Indonesia'}
.copy()	Copy all item of set		object_1= object.copy() object_1	{('bag', 'shoes'), ('black', 'white'), 21, 5, 97, 'Indonesia'}
.clear()	Delete all item in set		object.clear() object	set()

Basic List Method

Method	Purpose	Variabel	Example	
			In[ ]	Out[ ]
.insert()	Add new item at specified index	color=['yellow', 'orange','black']	color.insert(0,'pink') color	['pink','yellow','orange','black']
.append()	Add new item to the end of list		color.append('blue') color	['pink','yellow','orange','black','blue']
.extend()	Add new item of another sequence to the end of list		color.extend(['violet','green'] color	['pink','yellow','orange','black','blue', 'violet','green']
.remove()	Delete specified item of list		color.remove('green') color	['pink','yellow','orange','black','blue', 'violet']
.sort()	Sort the list		color.sort() color	['black','blue','orange','pink','violet', 'yellow']
.reverse()	Reverse the order of list		color.reverse() color	['yellow','violet','pink','orange', 'blue',black']
.notation[]	Modify item in list with new item/ know item with specified index		color.[2]='gray' color	['yellow','violet','gray','orange', 'blue',black']
.in	Check whether an item exist in list		'yellow' in color	True
.not in	Check whether an item not exist in list		'brown' not in color	True
.copy()	Copy all item of list		color_1 = color.copy() color_1	['yellow','violet','gray','orange', 'blue',black']
.pop()	Remove item at specified index		color_1.pop(1) color_1	['yellow','gray','orange','blue',black']
.clear()	Delete all item in list		color.clear() color	[]

Basic Dictionary Method

Method	Purpose	Variabel	Example	
			In[ ]	Out[ ]
.count()	Count the number of occurrence of specified item	fruit=('apel','orange', 'grape','guava','grape', 'apel','banana','apel')	print(fruit.count('strawberry')) print(fruit.count('apel')) print(fruit.count('grape'))	0 3 2
.index()	Searches index of specified item		print(fruit.index('apel')) print(fruit.index('orange'))	0 1

Basic Tuple Method

Purpose	Variabel	Example	
		In[]	Out[ ]
Accesing the value associated with a key	fruit_qty = {'apel' : 2, 'orange' : 5,'grape' : 7}	fruit_qty['grape']	7
Updating the value of an existing key-value pair		fruit_qty['grape'] = 10 fruit_qty	{'apel' : 2, 'orange' : 5, 'grape' : 10}
Adding a new key-value pair		fruit-qty['guava'] = 3 fruit_qty	{'apel' : 2, 'orange' : 5, 'grape' : 10, 'guava' : 3}
Removing a key-value pair		del fruit_qty['orange'] fruit_qty	{'apel' : 2, 'grape' : 10, 'guava' : 3}
		fruit_qty.pop('apel') fruit_qty	{'grape' : 10, 'guava' : 3}
Accessing nonexistent key		fruit_qty.get('orange','orange not in dictionary')	'orange not in dictionary'

Mathematic Set Operation

Method	Operator	Assignment Operator	Definition	Variabel	Example	
					In[ ]	Out[ ]
Union	|	|=	Consist of all the unique elements from both sets	score1 = {70,80,90} score2 = {60,70,90,100}	score1 | score2	{60, 70, 80, 90, 100}
Intersection	&	&=	Consist of all the unique elements that the two sets have in common		score1 & score2	{70, 90}
Difference	-	-=	Consist of different element between left and right operand		print(score1 - score2) print(score2 - score1)	{80} {100, 60}
Symmetric Difference	^	^=	Consist of different all element between left and right operand		score1 ^ score2	{60, 80, 100}
Disjoint	.isdisjoint	-	If they do not have any common elements		score1.isdisjoint(score2)	false