

BASIC PYTHON COMMANDS

LIST = [] : list(X)

.append(1 'obj')	Appends to END
.clear()	Clears List
.copy()	Make Copy
.count('obj')	Counts Instances
.extend(1 'obj')	Takes in List and EXTENDS
.index('obj')	Returns ONLY FIRST INDEX
.insert(#, 'obj')	Insert (INDEX, OBJECT)
.pop(INDEX)	DELETES BY INDEX Position
.remove('NAME')	DELETES BY NAME
.reverse()	SORT Descending order
.sort()	SORT Ascending order

DICTIONARY = {key:value} : dict(X)

Add new key-value pair: dict["Orange"] = 6
Overwrite key-value pair: dict["Orange"] = 8

.clear()	Clears Dict
.copy()	Make Copy
.get('key', 'default val')	Get Value Of Key
.items()	Get All Keys & Values
.keys()	Get All Keys
.pop('key')	Deletes Key
.popitem()	Deletes Last Key
.setdefault(key, default='x')	Returns value=default
.values()	Get all values
.fromkeys(iterable, value)	Create a new dict with keys from iterable and values.

Update dict 1 from dict 2

```
dict_1 = {'one':1, 'two':3}
dict_2 = {'two':2}
dict_1.update(dict_2) → 'one':1, 'two':2
```

TUPLE COMMANDS = ('a', 'b') : tuple()

.count('value')	Return occurrences of value
.index('item')	Returns index position
tupA + tupB	Concatenation

6/10 Escape Characters In Python

\b	Backspace	\e	Escape
\n	Newline	\s	Space
\t	Tab	\v	Vertical Tab

%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

33 Python Reserved Words By Example

- **from ... import ... as**
- **False / True**
- **def function: return**
- **if (and / or / not / in / == / !=): elif: else:**
- **while ... : continue: pass: break:**
- **try: ... except: else: ... finally:**
 - try**-lets you test code for errors.
 - except**-lets you handle the error.
 - else**-lets you execute code when there is no error.
 - finally**-lets you execute code, regardless of result
- **assert ... :**
- **lambda arguments : expression** x = lambda a : a + 10
- **del** - delete variable
- **with** - with open('file_path', 'w') as file:
file.write('hello world !')
- **Is - Points To Same Address**
a = 100
b = a
result = a is b
print(result) # True
- **None, nonlocal, global, yield, raise, class, in**

SET = set(['a', 'b', 'c']) : set(X)

.add	ADD to set
.clear	Clears all
.copy	make copy
.difference	Finds A-B
.discard	Remove specified item
.union	U
.intersection	∩
.isdisjoint	$A \cap B = \phi$
.issubset	$A \subseteq B$, A is subset of B
.issuperset	$A \supseteq B$
.pop	Remove from end
.remove	Remove at index position
.symmetric_difference:	$A \Delta B = (A \cup B) - (A \cap B)$
.intersection_update	
.symmetric_difference_update	
.difference_update	
.update	

String Methods ; str(X)

- **find, rfind, startswith, endswith, rindex, index**
- **capitalize, lower, upper, title**
- **isalnum, isalpha, isascii, isdecimal, isdigit, islower, isnumeric, isspace, istitle, isupper**
- **strip, lstrip, rstrip,**
- **split, rsplit, splitlines,**
- **count, format, join, replace, zfill**