

# 1. Definitions and Short Answers

1. What is a **comment** in a program and what is its purpose? **Help people understand the codes in order to reuse program.**
2. What is an **operator**? Give some examples of **arithmetic operators** in Python.  
**Operator performs operation on operand(s). + - \* // % \*\***
3. What is a **comparison operator**? What are possible results of a comparison?  
**Comparison operator compares two values. only boolean values: True, False**
4. What is a **logical operator**? What are possible results of a logical operation?  
**To combine conditional statements. Results**
5. What is 20 in **hexadecimal** representation? **14** in **octal** representation?**24**
6. Why does Python support two **division operators**? What is their difference?  
**// gives integer / gives floating point**
7. What is the difference between `'12'` and `12` in Python?  
**The former is a string. The latter is an integer.**
8. What is the difference between `x = 3` and `x == 3` in Python?  
**Integer 3 is assigned to variable x ; compare the value of x with integer 3**
9. Assuming the variable `y` has been assigned the integer value of 4, which of the following are legal in Python and what do they do? which are illegal in Python?
  - `y = 4` **legal integer 4 is assigned to variable y**
  - `4 = y` **illegal**
  - `y == 4` **legal compare the two values**
  - `4 == y` **legal compare the two values**
  - `'y' = y` **illegal**
  - `'y' == '4'` **illegal**
  - `'4' = y` **illegal**
10. Assume variable `x` has integer value 3, and variable `y` has integer value of 4. What is the result of the following operator expressions, if they are legal in Python? Which of the following are not legal?
  - `x * y` **legal 12**
  - `'x' * y` **legal 'xxxx'**
  - `x * 'y'` **legal 'yyy'**
  - `'x' * 'y'` **illegal**
  - `x + y` **legal 7**
  - `'x' + 'y'` **legal 'xy'**
  - `'x' + y` **illegal**
  - `x + 'y'` **illegal**
11. What is the data type of `['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat']`?  
**List**
12. if `L = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat']`, then what are the **values** of the following **expressions** if they are legal Python? Which are illegal?
  - `L[3]` **legal 'Wed'**
  - `L[1:5]` **legal ['Mon', 'Tue', 'Wed', 'Thu']**
  - `L[5:1]` **legal []**
  - `L(2:3)` **illegal**

- `L[1,2,3]` illegal
- `L{3}` illegal
- `L[1-5]` legal 'Wed'
- `L['3']` illegal

13. Assume `T = ('Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat')`, which of the following are allowed in Python, and what are their output or effect? Which are not allowed, for what reasons?

- `print(T[3])` legal Wed
- `print(T(3))` illegal, () is for parameter passing
- `print(T{3})` illegal, to define set or dictionary
- `T[3] = 'WED'` illegal, Tuple is unchangeable
- `T[3] == 'WED'` legal False
- `print(T[3:5])` legal ('Wed', 'Thu')
- `print(T[3, 5])` illegal, this statement
- `print(T['3'])` illegal, not supported

14. Assume `S = {'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'}`, which of the following are allowed in Python, and what are its output or effect? Which are not allowed, for what reasons?

- `print(S[3])` illegal
- `print(S(3))` illegal
- `print(S{3})` illegal
- `S[3] = 'WED'` illegal
- `S[3] == 'WED'` illegal
- `print(S[3:5])` illegal
- `print(S[3, 5])` illegal
- `print(S['3'])` illegal

15. Assume `D = {'Sun':0, 'Mon':1, 'Tue':2, 'Wed':3, 'Thu':4, 'Fri':5, 'Sat':6}`, which of the following are legal in Python, and what are their values?

- `D[3]` illegal
- `D['Thu']` legal, 4
- `D[0:3]` illegal
- `D[2, 6]` illegal
- `D{'Sun'}` illegal
- `D(0)` illegal
- `D{3}` illegal
- `D('Sun')` illegal

16. What is the value of `{ 2, 3, 4 } | { 3, 4, 5 }`? {2, 3, 4, 5}

17. What is the value of `{ 2, 3, 4 } & { 3, 4, 5 }`? {3, 4}

18. Suppose you have the following sequence of Python statements:

```
x = 3
y = 2
if x > y:
    print("x is bigger than y")
elif x == y:
```

```

    print("x and y are the same")
else:
    print("x is smaller than y")

```

What is printed? **X is bigger than y**

19. What is wrong with the following code, which is supposed to compute the total of a list of numbers?

```

L = [3, 2, 6, 5]
for i in L:
    total = total + i
print(total)

```

How can it be fixed?

**Forgot to initialize total.**

**Initialize total = 0 before the loop.**

20. What is the difference between

```

x = 0
while x < 100:
    x = x + 1
and
x = 0
if x < 100:
    x = x + 1
?

```

**while loop repeats. if doesn't**

21. What is an example of a **function** in Python? **print('Hello')** How do you **call** a function? **nameofFunction(parameters...)** What is a **parameter**?
22. What is an example of calling a function that **returns a value**? **input('enter your name')**
23. Python supports two kinds of **loops**. What are they? **for loop and while loop**
24. What is a **suite**? **indented block of statements associated with a control construct such as a loop or a conditional**
25. What does **import math** do? **Bring math module into the program.** How do you call the **cos** function (cosine) defined in the **math** module in Python? **math.cos(parameter)**
26. To read a file, it is common to see **fh = open('filename')**. What kind of data is **fh** called? **file object** Give an example of using **fh** for accessing (e.g., reading or writing) a file. **L = h.readlines()**

27. if **s = 'hello'**, Python supports two styles of "calls" (or "invocation"):

- **len(s)** is an example of a **function call**
- **s.upper()** is another form of call. What kind of call is it?

**It is to call a function of an object.**

28. How are **class** and **instance** related to each other?

29. Why is it incorrect to split the statement

```

f = a + b * 2 + c / 2 - 4 * d

```

onto two separate lines as the following

```

f = a + b * 2 + c / 2
- 4 * d

```

? **The two lines will be regarded as two statements.**

How can it be fixed so Python will accept it?

Add \: `f = a + b * 2 + c / 2 \`  
`- 4 * d`

30. If you want to **swap** the values of two variables `x` and `y`, why can't you just do

```
x = y
y = x
?
```

`x = y`: the value of `x` is replaced by `y`

`y = x`: `y` and `x` will get the same value

Give two different ways you can swap their values correctly in Python.

1. `(x, y) = (y, x)`

2. `Temp = x`

`x = y`

`y = temp`

31. What is a **keyword** in Python? **words with reserved meaning and cannot be reused for other purpose** Give some example keywords in Python. `False, None, True, and, as`

32. Which of the following are legal and illegal **identifiers** in Python?

- `myname` legal
- `my_name` legal
- `_myname` legal
- `MyName` legal
- `myname_` legal
- `my-name` illegal
- `my11name` legal
- `myname11` legal
- `11myName` illegal
- `my_11Name` legal
- `_11myName` legal
- `@myname` illegal
- `my@name` illegal
- `myname@` illegal
- `in` illegal
- `out` legal
- `_in` legal
- `_out` legal
- `IN` legal
- `OUT` legal
- `And` illegal
- `Or` illegal
- `but` legal
- `function` legal
- `integer` legal
- `number` legal
- `class` illegal
- `instance` legal
- `global` illegal

- local legal
- you+me illegal
- I\_love\_\$\$ illegal

33. What is an example of a **snake-case** identifier? a **camel-case** identifier?

snake-case: average\_midterm\_score

camel-case: averageMidtermScore