Week-2, Practice Assignment (theory)

Problem-1

Question

Which of the following are valid names for variables in Python? It is a Multiple Select Question (MSQ).

(a) `a_`

(b) `_a`

(c) `1a`

(d) `a variable`

(e) `a_variable`

Answer

(a), (b), (e)

Solution

Rules for variable name in python:-

- A variable name must be started with a letter or the underscore character.
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters and underscores (A-Z, a-z, 0-9, and _).
- A variable name can not contain any blank space in it.

Hence, (a), (b), and (e) are correct according to the rules.

Problem-2

Question

Consider the following code-block. `word1` and `word2` are strings that have already been defined:

```
1 # word1, word2 are two strings
2 print(word1 + word2)
3 word = word1
4 word1 = word2
5 word2 = word
6 print(word2 + word1)
```

Which of the following statements are true about the two lines in the output?

(a) The two lines are always the same.

(b) The two lines are always different.

(c) The two lines are the same if and only if `word1` and `word2` are equal.

(d) None of the above

Answer

(a)

Solution

- Assume that `word1='Hello'` and `word2='Python'`.
- In line 2, `print(word1 + word2)` print the concatenation of word1 and word2 (`HelloPython`) in output.
- Line 3 to 5, swap the value of `word1` and `word2`.
- In line 6, `print(word2 + word1)` print the concatenation of word1 and word1 (`HelloPython`) in output which will be the same as the previous print statement output.

Hence, (a) is correct .

Problem-3

Question

Assume that `a`, `b` and `c` are three distinct integers. The following code runs without any error.

```
x, y, z = a, b, c
x = y = z
```

Which of the following statements evaluate to `True` at the end of execution of the code given above? It is a Multiple Select Question (MSQ).

(a) `x == y == z`

(b) `x == y == z == a`

(c) `x == y == z == b`

(d) `x == y == z == c`

Answer

(a), (d)

Solution

- In the first line, we are using multiple assignments in one line. So, after the first line of execution, x = a, y = b and z = c.
- In the second line, we Know that `=` operator has the right to left associativity. So, the value of z which is c, will be assigned to y and the value of y which is now c, will be assigned to x. So finally, all variables will contain the same value c.

Hence, (a) and (d) are correct .

Problem-4

Question

`x` is a variable of type `float` and is of the form `a.bcd`, where `a, b, c, d` are all positive integers less than 10. What is the output of the following snippet of code?

```
print(int(-x))
```

(a) `a`

(b) `a + 1`

(c) `-a`

(d) `-a - 1`

(e) `-a + 1`

Answer

(c)

Solution

`int(-x)` converts the float value to integer value and returns only the integer part(sign remains same) and discards the decimal part.

Hence, (c) is correct .

Problem-5

Question

Consider the following code-blocks. `E` is a Boolean expression.

`Block-1`:

```
if E:
    print('good')
else:
    print('bad')
```

`Block-2`:

```
if E:
    print('good')
print('bad')
```

`Block-3`:

```
print('good')
print('bad')
```

Which of the following statements are true? It is a Multiple Select Question (MSQ).

(a) All three blocks are equivalent to each other.

(b) Exactly two of these three blocks are equivalent to each other.

(c) Blocks 1 and 2 print the same output when `E` evaluates to `True`.

(d) Blocks 1 and 2 print the same output when `E` evaluates to `False`.

(e) Blocks 2 and 3 print the same output when `E` evaluates to `True`.

Answer

(d), (e)

Solution

For Block-1 possible cases:-

- If `E = True`, output will be `good`
- If `E = False`, output will be `bad`

For Block-2 possible cases:-

- If `E = True`, output will be `good` and `bad` both.
- If `E = False`, output will be `bad`

For Block-3 possible cases:-

- Block-3 code is not dependent on `E` so the output always will be `good` and `bad` .

Hence, (d) and (e) are correct according to the above observation .

Problem-6

Question

`bool_var` is a variable of type `bool`. `x` is a variable of type `int`. Assume that both these variables have already been defined. Now, consider the following code-block:

```
1 if bool_var:
2     x = x + 1
3     bool_var = not bool_var
4     if bool_var:
5         x = x + 1
6     else:
7         x = x - 1
8 print(x)
```

Which of the following statements are true at the end of execution of the code-block given above? It is a Multiple Select Question (MSQ).

(a) The value of variable `x` is independent of the value of `bool_var`.

(b) The value of variable `x` is dependent on the value of `boon_var`.

(c) Line-5 is never executed.

(d) The variable `x` is updated exactly two times.

Answer

(a), (c)

Solution

If `bool_var = True`

- Inside the `if` block, In line 2, `x` will be incremented by 1
- In line 3, the value of `bool_var` becomes `False` so in line 6, `else` block will be executed in which `x` will be decremented by 1. So finally `x` value will be printed which remains the same as the initial. We can see that here line 4 `if` block will never execute.

If `bool_var = False`

- Just print the value of `x`

Hence, (a) and (c) are correct.

Common data for problems 7 and 8

Consider the following code-block. `E_1, E_2` and `E_3` are all Boolean variables that have already been defined. `x` is a variable that has NOT been defined before.

```
if E_1:
    x = 1
if E_2:
    x = 2
if E_3:
    x = 3
print(x)
```

Problem-7

Question

When will this code throw an error?

(a) When all three Boolean variables are `True`.

(b) When all three Boolean variables are `False`.

(c) When at least one of the three Boolean variables is `True`.

(d) When at least one of the three Boolean variables is `False`.

(e) This code will never throw an error.

Answer

(b)

Solution

All possible cases for `E_1`, `E_2` and `E_3`:-

E_1	E_2	E_3	Output
`True`	`True`	`True`	3
`True`	`True`	`False`	2
`True`	`False`	`True`	3
`True`	`False`	`False`	1
`False`	`True`	`True`	3
`False`	`True`	`False`	2
`False`	`False`	`True`	3
`False`	`False`	`False`	`NameError: name 'x' is not defined`

We can see that here if any of the Boolean variables is True so x value will be initialized in `if` block otherwise x will not be initialized. So when all three Boolean variables are `False` then name 'x' is not defined error comes. Hence, option (b) is correct.

Problem-8

Question

If the code throws an error, in which line will it occur? Enter an integer between 1 and 7, both endpoints included.

Answer

7

Solution

All possible cases for `E_1`, `E_2` and `E_3`:-

E_1	E_2	E_3	Output
`True`	`True`	`True`	3
`True`	`True`	`False`	2
`True`	`False`	`True`	3
`True`	`False`	`False`	1
`False`	`True`	`True`	3
`False`	`True`	`False`	2
`False`	`False`	`True`	3
`False`	`False`	`False`	`NameError: name 'x' is not defined`

We can see that here if any of the Boolean variables is True so x value will be initialized in `if` block otherwise x will not be initialized. So when all three Boolean variables are `False` line number 7 gives error name 'x' is not defined . Hence, the answer is 7.

Problem-9

Question

Chose the correct matching option for operator and related function in string .

Operator	Function related to string
1. +	A. Replicates same string multiple times
2. *	B. Membership check
3. []	C. Escape character
4. [:]	D. Concatenates two string
5. in	E. Range slice
6. \	F. Character of the string using indexing

Chose the correct option for given code.

(a) 1-D, 2-A, 3-F, 4-E, 5-B, 6-C

(b) 1-D, 2-A, 3-F, 4-E, 5-C, 6-B

(c) 1-D, 2-A, 3-E, 4-F, 5-B, 6-C

(d) 1-A, 2-D, 3-E, 4-F, 5-B, 6-C

Answer

(a) 1-D, 2-A, 3-F, 4-E, 5-B, 6-C

Solution

Correct answer is option (a)

Operator	Function related to string
1. +	D. Concatenates two string
2. *	A. Replicates same string multiple times
3. []	F. Character of the string using indexing
4. [:]	E. Range slice
5. in	B. Membership check
6. \	C. Escape character

Problem-10

Third index was not introduced in the lectures, student has to search through the various resources available on internet and answer the following questions below.

Question

```
s = "abcdefghijklmnopqrstuvwxyz"
a = int(input())
b = int(input())
c = int(input())
d = int(input())
e = int(input())
print(s[-a:-len(s):-3])
print(s[::-b])
print(s[c:0:-3])
print(s[len(s):-d:-3])
print(s[:e:-3])
```

For what user input of `a`, `b`, `c`, `d` and `e` (where `0 <= a,b,c,d,e <= len(s)`) does the above code-snippet print following output ? It is a Numerical Answer Type (NAT) question. Enter your input value separated by `,` without any space like: `7,15,0,4,0`

```
zwtqnkheb
zwtqnkheb
zwtqnkheb
zwtqnkheb
zwtqnkheb
```

Answer

1,3,26,26,0

or

1,3,25,26,0

or

1,3,26,0,0

or

1,3,25,0,0

Solution

`s[start : end : step]`

`start` is start index.

`end` is last index+1(because end is not inclusive).

`step` define interval of indexing in range (`start , start+step , start+step+step,.....end-1`).

For traverse string left to right:-

- If `start`, `end` and `step` is not defined in slicing then by default `start` will be first index of string, `end` will be one more then from last index of element and `step` will be 1.
- If we are visiting elements from left to right ,we must remember that the `start` index must be at the left side from the `end` index and the `step` value should be positive.

For traverse string right to left:-

- `step` value must be mention and it should be negative
- If we are visiting elements from right to left ,we must remember that the `start` index must be at the right side from the `end` index and the `step` value should be negative.
- If `start` and `end` index is not defined and `step` value is negative so by default `start` index will be the last index of string and `end` index will be one less then from first index of string.

We can see that in the output, characters are printed in reverse order from last character to first character with an interval of 3. So all possible combinations given in the answer will print the same output `zwtqnkheb` 5 times.