Contenido creado por: Isabel Maniega

Question 1

What is the result of the following code?

```
x = 7
y = x % 2
y += 1
print(y)
```

A. `1`

B. `5`

C. `3`

D. `2`

Solution 1

In [2]:
```
x = 7
y = x % 2    # y = 7 % 2   =>   y = 1
y += 1       # y = y + 1   =>   y = 2
print(y)                      # 2
```

```
2
```

In [3]:
```
# Solución
# D
```

Question 2 (Data Aggregates)

Take a look at the snippet and choose the true statement:

```
nums = [1, 2, 3]
vals = nums
del vals[:]
```

A. nums is longer than vals

B. nums and vals have the same length

C. the snippet will cause a runtime error

D. vals is longer than nums

Solution 2

In [5]:
```
nums = [1, 2, 3]
print('nums:', nums)
vals = nums
print('hemos hecho vals=nums')
del vals[:]
print('hemos hecho del vals[:]')

print("\n")
print('nums:', nums, 'vals:', vals)
print('len(nums):', len(nums), ' y  len(vals):', len(vals))

# los cambios hechos en uno se hacen también para el otro
```

```
nums: [1, 2, 3]
hemos hecho vals=nums
hemos hecho del vals[:]


nums: [] vals: []
len(nums): 0  y  len(vals): 0
```

In [6]:
```
# Solución
# B
```

Question 3 (Operators)

The expression:

```
'mike' > 'Mike'
```

is

A. `erroneous`

B. `False`

C. `True`

Solution 3

In [8]:
```
ord('m'), ord('M'), ord('m') > ord('M')
```

Out[8]:
```
(109, 77, True)
```

In [9]:
```
'mike' > 'Mike'
```

Out[9]: True

In [10]:
```
# Solución
# C
```

Question 4 (Control Flow)

What is the expected output of the following code?

```
x = (1, 4, 7, 9, 10, 11)
y = {2: 'A', 4: 'B', 6: 'C', 8: 'D', 10: 'E', 12: 'F'}
res = 1
for z in x:
    if z in y:
        res += z
print(res)
```

A. `14`

B. `6`

C. `15`

D. `22`

Solution 4

In [12]:
```
x = (1, 4, 7, 9, 10, 11)

y = {2: 'A',
     4: 'B',
     6: 'C',
     8: 'D',
     10: 'E',
     12: 'F'}

res = 1
for z in x:
    if z in y:
        res += z    # res = res + z

print(res)
```

15

In [13]:
```
# Solución
# C
```

Question 5 (Data Types)

What is the expected output of the following code?

```
print(not 0)
print(not 23)
print(not '')
print(not 'Peter')
print(not None)
```

A.

```
True
False
True
False
False
```

B.

```
False
False
True
False
True
```

C.

```
True
False
True
False
True
```

D.

```
True
False
False
False
True
```

Solution 5

In [15]:
```
print(not 0)          # 1
print(not 23)         # 0
print(not '')         # 1
print(not 'Peter')    # 0
print(not None)       # 1
```

```
True
False
True
False
True
```

In [16]:
```
# Solución
# C
```

Question 6 (Basics)

What is machine code?

A.

A low-level programming language consisting of binary digits/bit that the computer reads and understands

B.

A low-level programming language consisting of hexadecimal digits that make up high-level language instructions

C.

A high-level programming language consisting of instruction lists that humans can read and understand

D.

A medium-level programming language consisting of the assembly code designed for the computer processor

Solution 6

In [18]:
```
# Solución
# A
```

Question 7 (Functions)

What is the expected behavior of the following snippet?

```
x = 1

def a(x):
    return 2 * x
```

```
x = 2 + a(x)      # Line 8
print(a(x))       # Line 9
```

It will:

A. cause a runtime exception on Line 9

B. print `4`

C. print `6`

D. print `8`

E. cause a runtime exception on Line 8

Solution 7

In [20]:
```
x = 1


def a(x):
    return 2 * x


x = 2 + a(x)      # Line 8
# x = 2 + a(1)  (x=1)
# x = 2 + 2*1  => x = 4
print(a(x))       # Line 9
# print(a(4))
# 2 * 4 => 8
```

8

In [21]:
```
# Solución
# D
```

Question 8 (Operators)

Evaluate the following Python arithmetic expression:

```
(3 * (1 + 2) ** 2 - (2 ** 2) * 3)
```

What is the result?

A. `13`

B. `3`

C. `69`

D. 15

Solution 8

In [23]:
```
(3 * (1 + 2) ** 2 - (2 ** 2) * 3)
# 3 *  (3    ** 2) -    4     * 3
# 3 * 9 - 12
# 27 - 12 = 15

# 15
```

Out[23]: 15

In [24]:
```
# Solución
# D
```

Question 9 (Data Aggregates)

How many elements does the my_list list contain?

```
my_list = [0 for i in range(1, 3)]
```

A. two

B. three

C. one

Solution 9

In [26]:
```
my_list = [0 for i in range(1, 3)] # 1,2
print(my_list)                     # [0, 0]
print(len(my_list))                # 2
```

```
[0, 0]
2
```

In [27]:
```
# Solución
# A
```

Question 10 (Data Types)

What is the expected output of the following code if the user enters 3 and 2?

```
x = int(input())
y = int(input())
x = x % y
x = x % y
y = y % x
print(y)
```

A. `3`

B. `0`

C. `1`

D. `2`

Solution 10

In [29]:
```
# 3 and 2

x = int(input())  # x = 3
y = int(input())  # y = 2
x = x % y         # 3 % 2 => x = 1
x = x % y         # x = 1 % 2 => x = 1
y = y % x         # y = 2 % 1 = > y = 0
print(y)          # 0
```

```
3
2
0
```

In [30]:
```
# Solución
# B
```

Question 11 (Operators)

An operator able to check whether two values are equal, is coded as:

A. `is`

B. `==`

C. `===`

D. `=`

Solution 11

In [32]:
```
a = 40
b = 41
```

```
a == b
```

Out[32]: False

In [33]:
```
# Solución
# B
```

Question 12 (Operators)

What is the expected output of the following code?

```
print(1 // 2)
```

A. None of the above

B. `0.0`

C. `0`

D. `0.5`

Solution 12

In [35]:
```
print(1 // 2)  # 0
```

```
0
```

In [36]:
```
# Solución
# C
```

Question 13 (Data Aggregates)

What is the expected output of the following code?

```
data = 'Hello@Peter!!'
print(data.lower())
```

A. `hellopeter`

B. `hello@peter!!`

C. `hello@Peter!!`

D. `None`

Solution 13

In [38]:
```
data = 'Hello@Peter!!'
print(data.lower())
```

```
hello@peter!!
```

In [39]:
```
# Solución
# B
```

Question 14 (Data Aggregates)

Insert the correct snippet to convert the t tuple to a dictionary named d

Expected output:

```
{'A': 1, 'B': 2, 'C': 3}
```

Code:

```
t = (('A', 1), ('B', 2), ('C', 3))
insert code here
print(d)
```

A. `t >> d.dict`

B. `d = t(dict)`

C. `d.dict(t)`

D. `d = dict(t)`

Solution 14

In [41]:
```
# Code:

t = (('A', 1),
     ('B', 2),
     ('C', 3))

# insert code here
d = dict(t)

print(d)


# Expected output:
# {'A': 1, 'B': 2, 'C': 3}
```

```
{'A': 1, 'B': 2, 'C': 3}
```

In [42]:
```
# Solución
# D
```

Question 15 (Data Aggregates)

What is the output of the following snippet?

```
l1 = [1, 2, 3]

for v in range(len(l1)):
    l1.insert(1, l1[v])

print(l1)
```

A. `[1, 2, 3, 3, 2, 1]`

B. `[1, 2, 3, 1, 2, 3]`

C. `[1, 1, 1, 1, 2, 3]`

D. `[3, 2, 1, 1, 2, 3]`

Solution 15

In [44]:
```
l1 = [1, 2, 3]

for v in range(len(l1)):    # len(l1) = 3 => 0,1,2

  #  INICIAL     [1, 2, 3]   ------------------- l1(0) = 1

    l1.insert(1, l1[v])     # insert(index, value)
    # l1.insert(1, 1)  ====> [1, 1, 2, 3]    ----  l1(1) = 1
    # l1.insert(1, 1)  ====> [1, 1, 1, 2, 3] ----  l1[2] = 1
    # l1.insert(1, 1)  ====> [1, 1, 1, 1, 2, 3]

print(l1)
```

```
[1, 1, 1, 1, 2, 3]
```

In [45]:
```
# Solución
# C
```

Question 16 (Functions)

What is the output of the following snippet?

```
def fun(x):
    x += 1
    return x

x = 2
x = fun(x + 1)
print(x)
```

A. `4`

B. the code is erroneous

C. `3`

D. `5`

Solution 16

In [47]:
```
def fun(x):
    x += 1     # x = x + 1
    return x

x = 2
x = fun(x + 1)   # fun(3)
print(x)         # 4
```

```
4
```

In [48]:
```
# Solución
# A
```

Question 17 (Functions)

What is the expected output of the following code?

```
def func1(x):
    return str(x)

def func2(x):
    return str(2 * x)

print(func1(1) + func2(2))
```

A. `14`

B. `5`

C. The code is erroneous

D. `3`

Solution 17

In [50]:
```
# si ejecuto todas las celdas a la vez no va !!!! no sale el 14..
# ALGUNA VEZ NO SALIA EL RESULTADO
```

In [1]:
```
def func1(x):
    return str(x)
```

In [2]:
```
def func2(x):
    return str(2*x)
```

In [4]:
```
print(func1(1) + func2(2))
```

```
14
```

In [52]:
```
# Solución
# A
```

Question 18 (Data Aggregates)

What is the expected output of the following code?

```
print(type(+1E10))
print(type(5.0))
print(type('True'))
print(type(False))
```

A.

```
<class 'float'>
<class 'float'>
<class 'str'>
<class 'bool'>
```

B.

```
<class 'float'>
<class 'float'>
<class 'bool'>
<class 'bool'>
```

C.

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
```

D.

```
<class 'int'>
<class 'float'>
<class 'bool'>
<class 'bool'>```
```

Solution 18

In [54]:
```
print(type(+1E10))    # float
print(type(5.0))      # float
print(type('True'))   # str
print(type(False))    # bool
```

```
<class 'float'>
<class 'float'>
<class 'str'>
<class 'bool'>
```

In [55]:
```
print(+1E10)    # float
```

```
10000000000.0
```

In [56]:
```
# Solución
# A
```

Question 19 (Data Aggregates)

What is the expected output of the following code?

```
x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
x[::2] = 10, 20, 30, 40, 50, 60
print(x)
```

A. `[1, 10, 3, 20, 5, 30, 7, 40, 9, 50, 60]`

B. `[1, 2, 10, 20, 30, 40, 50, 60]`

C. `[10, 2, 20, 4, 30, 6, 40, 8, 50, 60]`

D. The code is erroneous

Solution 19

In [58]:
```
# ejemplo previo
x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
x[::2]
```

Out[58]:
```
[1, 3, 5, 7, 9]
```

In [59]:
```
# el propio ejercicio

"""
x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
# x[::2] = [1, 3, 5, 7, 9]

x[::2] = 10, 20, 30, 40, 50, 60
print(x)
"""


# ValueError:
# attempt to assign sequence of size 6 to extended slice of size 5
```

Out[59]:
```
'\nx = [1, 2, 3, 4, 5, 6, 7, 8, 9]\n# x[::2] = [1, 3, 5, 7, 9]\n\nx[::2]
= 10, 20, 30, 40, 50, 60\nprint(x)\n'
```

In [60]:
```
# Solución
# D
```

Question 20 (Operators)

What will be the output of the following snippet?

```
a = 1
b = 0
a = a ^ b
b = a ^ b
b = a ^ b
print(a, b)
```

A. `1 0`

B. `0 1`

C. `1 1`

D. `0 0`

Solution 20

In [62]:
```
a = 1                       #  a = 1
b = 0                       #  b = 0
a = a ^ b     # a = 1 ^ 0 => a = 1
b = a ^ b     # b = 1 ^ 0 => b = 1
b = a ^ b     # b = 1 ^ 1 => b = 0
print(a, b)   # 1 0
```

```
1 0
```

In [63]:
```
# Solución
```

```
# A
```

Question 21

Consider the following list.

```
data = [1, 5, 10, 19, 55, 30, 55, 99]
```

Which of the code snippets below would produce a new list like the following?

```
[1, 5, 10, 99]
```

A.

```
data.pop(5)
data.remove(19)
data.remove(55)
data.remove(55)
```

B.

```
data.pop(1)
data.pop(3)
data.pop(4)
data.pop(6)
```

C.

```
data.pop(5)
data.pop(19)
data.pop(55)
```

D.

```
None of the above
```

E.

```
data.remove(5)
data.remove(19)
data.remove(55)
```

Solution 21

In [65]:
```
# datos iniciales:    data = [1, 5, 10, 19, 55, 30, 55, 99]
# lista final:        data = [1, 5, 10, 99]
```

```
data = [1, 5, 10, 19, 55, 30, 55, 99]
print('data inicial:', data)
data.pop(5)          # borro posición 5 => 30
print('data después de pop(5):', data)
data.remove(19)    #
print('data después de remove(19):', data)
data.remove(55)
print('data después del primer remove(55):', data)
data.remove(55)
print('data después del segundo remove(55) y final:', data)
```

```
data inicial: [1, 5, 10, 19, 55, 30, 55, 99]
data después de pop(5): [1, 5, 10, 19, 55, 55, 99]
data después de remove(19): [1, 5, 10, 55, 55, 99]
data después del primer remove(55): [1, 5, 10, 55, 99]
data después del segundo remove(55) y final: [1, 5, 10, 99]
```

In [66]:
```
# Solución
# A
```

Question 22 (Functions)

What is the expected output of the following code?

```
v = 1

def fun():
    global v
    v = 2
    return v

print(v)
```

A. `2`

B. `None`

C. The program will cause an error

D. `1`

Solution 22

In [68]:
```
v = 1

def fun():
    global v
    v = 2
    return v

print(v)
```

1

In [69]:
```
# Solución
# D
```

Question 23 (Data Aggregates)

What is the expected output of the following code?

```
data1 = 'a', 'b'
data2 = ('a', 'b')
print(data1 == data2)
```

A. `False`

B. `1`

C. `True`

D. `0`

Solution 23

In [71]:
```
data1 = 'a', 'b'
print('type de data1:', type(data1))
print('data1:', data1)

data2 = ('a', 'b')
print('type de data1:', type(data2))
print('data2:', data2)

print("\n")

print(data1 == data2)
```

```
type de data1: <class 'tuple'>
data1: ('a', 'b')
type de data1: <class 'tuple'>
data2: ('a', 'b')


True
```

In [72]:
```
# Solución
# C
```

Question 24 (Functions)

What does the following code do?

```
def a(b, c, d):
    pass
```

A. Defines a list and initializes it

B. Defines an empty class

C. Defines a function, which passes its parameters through

D. Defines a function, which does nothing

E. None of the above

Solution 24

In [74]:
```
def a(b, c, d):
    pass
```

In [75]:
```
# Solución
# D
```

Question 25 (Error Handling)

The following statement ...

```
assert x == 0
```

A. will stop the program if x is equal to `0`

B. is erroneous

C. has no effect

D. will stop the program if x is not equal to `0`

Solution 25

In [77]:
```
x = 0
assert x == 0
```

In [2]:
```
"""
x = 2
assert x == 0, 'x no es 0'
"""
```

```
# AssertionError: x no es 0
```

Out[2]: "\nx = 2\nassert x == 0, 'x no es 0'\n"

In [79]:
```
# Solución
# D
```

Question 26 (Operators)

What value will be assigned to the x variable?

```
z = 10
y = 0
x = z > y or z == y
```

A. `True`

B. `False`

C. `1`

Solution 26

In [81]:
```
z = 10
y = 0
x = z > y or z == y   # x = True or False = True

print(x)
```

True

In [82]:
```
# Solución
# A
```

Question 27 (Data Types)

Which of the following operators can be used with strings?

1. +
2. *
3. -
4. in

A. `1, 2, 3`

B. `1, 2, 4`

C. `1, 2, 3, 4`

D. `1, 2`

Solution 27

In [1]:
```
s1 = 'hola'
s2 = 'mundo'
s1, s2
```

Out[1]:
```
('hola', 'mundo')
```

In [85]:
```
# 1
s1 + s2
```

Out[85]:
```
'holamundo'
```

In [86]:
```
# 2
# s1 * s2
# TypeError: can't multiply sequence by non-int of type 'str'
```

In [87]:
```
# 2 (otra opción, strings con números)
3 * s1
```

Out[87]:
```
'holaholahola'
```

In [88]:
```
# 3
# s1 - s2
# TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

In [89]:
```
# 3 (otra opción, string con números)
# 3-s1
# TypeError: unsupported operand type(s) for -: 'int' and 'str'
```

In [90]:
```
# d
s1 in s2
```

Out[90]:
```
False
```

In [91]:
```
# posibles 1,2,4
```

In [92]:
```
# Solución
# B
```

Question 28

What do you call a computer program which directly executes instructions written in a programming language?

A. A compiler

B. An interpreter

C. A translater

Solution 28

In [94]:
```
# Python es un ejemplo de lenguaje interpretado
```

In [95]:
```
# Solución
# B
```

Question 29 (Data Aggregates)

What is the expected output of the following code?

```
x = {(1, 2): 1, (2, 3): 2}
print(x[1, 2])
```

A. `{(2, 3): 2}`

B. `1`

C. `{(1, 2): 1}`

D. The code is erroneous

Solution 29

In [97]:
```
x = {(1, 2): 1,
     (2, 3): 2}

print(x[1, 2])
```

```
1
```

In [98]:
```
# otros ejemplos:
```

In [99]:
```
x = {(1, 2): 1,
     (2, 3): 2,
     (3, 4): 3,
     (4, 5): 4}

print(x[1, 2])
print(x[2, 3])
print(x[3, 4])
print(x[4, 5])
```

```
1
2
3
4
```

In [100…
```
# Solución
# B
```

Question 30 (Control Flow)

Which of the code snippet below will print the following to the monitor?

```
Paul
Mary
Jane
```

A.

```
data = ['Peter', 'Paul', 'Mary', 'Jane']
for d in data:
    if len(d) != 4:
        print(d)
```

B.

```
data = ['Peter', 'Paul', 'Mary', 'Jane']
da = data[1:]
for d in data:
    print(d)
```

C.

```
data = ['Peter', 'Paul', 'Mary', 'Jane']
for d in data:
    print(d)
```

D.

```
data = ['Peter', 'Paul', 'Mary', 'Jane']
for d in data:
    if len(d) == 4:
        print(d)
```

Solution 30

In [102…
```
# A
data = ['Peter', 'Paul', 'Mary', 'Jane']
for d in data:
    if len(d) != 4:
```

```
        print(d)

# buscamos:
# Paul
# Mary
# Jane
```

```
Peter
```

In [103…

```
# B
data = ['Peter', 'Paul', 'Mary', 'Jane']
da = data[1:]
for d in data:
    print(d)

# buscamos:
# Paul
# Mary
# Jane
```

```
Peter
Paul
Mary
Jane
```

In [104…

```
# C
data = ['Peter', 'Paul', 'Mary', 'Jane']
for d in data:
    print(d)

# buscamos:
# Paul
# Mary
# Jane
```

```
Peter
Paul
Mary
Jane
```

In [105…

```
# D
data = ['Peter', 'Paul', 'Mary', 'Jane']
for d in data:
    if len(d) == 4:
        print(d)

# buscamos:
# Paul
# Mary
# Jane
```

```
Paul
Mary
Jane
```

In [106…

```
# Solución
# D
```

Gracias por la atención

Isabel Maniega