Contenido creado por: Isabel Maniega

Question 1 (Operators)

What is the expected output of the following code?

```
x = True
y = False
x = x or y
y = x and y
x = x or y
print(x, y)
```

A. `True False`

B. `False False`

C. `True True`

D. `False True`

Solution 1

```
In [2]: x = True
        y = False
        x = x or y    # x = True or False => x = True
        y = x and y   # y = True and False => y = False
        x = x or y    # x = True or False => x = True
        print(x, y)   # True, False
```

```
True False
```

```
In [3]: # Solución
        # A
```

Question 2 (Data Aggregates)

What is the expected output of the following code?

```
numbers = [1, 2, 3, 4, 5]
nums = numbers[2: ]
print(nums)
```

A. The program will cause an error

B. `[2,3,4,5]`

C. `[3,4,5]`

D. `[2]`

Solution 2

In [5]:

```
numbers = [1, 2, 3, 4, 5]
nums = numbers[2: ]       # [3,4,5]   (desde index 2, incluído hasta el f
print(nums)
```

```
[3, 4, 5]
```

In [6]:

```
# Solución
# C
```

Question 3 (Operators)

The `**` operator:

A. performs floating-point multiplication

B. does not exist

C. performs duplicated multiplication

D. performs exponentiation

Solution 3

In [8]:

```
2*3, 2**3
```

Out[8]:

```
(6, 8)
```

In [9]:

```
# Solución
# D
```

Question 4

What is the expected output of the following code?

```
def fun(n):
    n **= n
```

```
        return n
    print(fun(3))
```

A. `27`

B. `9`

C. The program will cause an error

D. `3`

E. `True`

Solution 4

In [11]:

```
def fun(n):
    n **= n    # n = n ** n
    return n

print(fun(3))  # 3**3 = 27
```

```
27
```

In [12]:

```
# Solución
# A
```

Question 5 (Data Aggregates)

What is the expected output of the following code?

```
    data = [1, 2, 3, 4, 5, 6]

    for i in range(1, 6):
        data[i - 1] = data[i]

    for i in range(0, 6):
        print(data[i], end=' ')
```

A. `1 2 3 4 5 6`

B. `2 3 4 5 6 1`

C. `2 3 4 5 6 6`

D. `1 1 2 3 4 5`

Solution 5

In [14]:

```
data = [1, 2, 3, 4, 5, 6]

for i in range(1, 6):        # 1,2,3,4,5
    data[i - 1] = data[i]
    # data[0] = data[1] => 2
    # data[1] = data[2] => 3
    # data[2] = data[3] => 4
    # data[3] = data[4] => 5
    # data[4] = data[5] => 6
    # ---------------------6

for i in range(0, 6):
    print(data[i], end=' ') # 2 3 4 5 6 6
```

```
2 3 4 5 6 6
```

In [15]:

```
# Solución
# C
```

Question 6 (Operators)

What is the expected output of the following code?

```
print(1 // 2 * 3)
```

A. `0`

B. `0.0`

C. `0.1666666666666`

D. `4.5`

Solution 6

In [2]:

```
print(1 // 2 * 3)
#     1 //   6
# 0

# 0 entero por ser cociente entero, y son 2 números enteros (1 y 6)
```

```
0
```

In [18]:

```
# Solución
# A
```

Question 7

What is the output of the following snippet?

```
my_list = [x * x for x in range(5)]

def fun(lst):
    del lst[lst[2]]
    return lst

print(fun(my_list))
```

A. `[1, 4, 9, 16]`

B. `[0, 1, 4, 9]`

C. `[0, 1, 9, 16]`

D. `[0, 1, 4, 16]`

Solution 7

In [20]:

```
my_list = [x * x for x in range(5)]
# [0*0, 1*1, 2*2, 3*3, 4*4]
# [0,    1,    4,    9,   16]

def fun(lst):
    del lst[lst[2]]     # BORRAS 16  (del lst[4]) (siendo lst[2]=>4)
    return lst


print(fun(my_list))
```

```
[0, 1, 4, 9]
```

In [21]:

```
# Solución
# B
```

Question 8 (Data Aggregates)

What is the expected output of the following code?

```
data = set([1, 2, 2, 3, 3, 3, 4, 4, 4, 4])
print(len(data))
```

A. `0`

B. `2`

C. `1`

D. `3`

E. `4`

F. `10`

Solution 8

In [23]:

```
data = set([1, 2, 2, 3, 3, 3, 4, 4, 4, 4])
# 1,2,3,4
print(data)
```

```
{1, 2, 3, 4}
```

In [24]:

```
print(len(data))  # 4
```

```
4
```

In [25]:

```
# Solución
# E
```

Question 9 (Data Aggregates)

After execution of the following snippet,

the sum of all vals elements will be equal to:

```
vals = [0, 1, 2]
vals.insert(0, 1)
del vals[1]
```

A. `2`

B. `5`

C. `4`

D. `3`

Solution 9

In [27]:

```
vals = [0, 1, 2]
vals.insert(0, 1)  # [1,0,1,2]
del vals[1]        # [1, 1, 2]

vals
# SUMA: 4
```

Out[27]:

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

In [28]:

```
# si tuviéramos muchos números..
suma=0
for numero in vals:
    suma=suma+numero

suma
```

Out[28]:

```
4
```

In [29]:

```
# o con la propia palabra reservada
sum(vals)
```

Out[29]:

```
4
```

In [30]:

```
# Solución
# C
```

Question 10 (Functions)

What is the expected output of the following code?

```
data = 'abcdefg'

def func(text):
    del text[2]
    return text

print(func(data))
```

A. The code is erroneous

B. `abdef`

C. `abcef`

D. `acdef`

Solution 10

In [5]:

```
"""string = 'casa'
del string[0]
string"""

# TypeError:
# 'str' object doesn't support item deletion
```

Out[5]:

```
"string = 'casa'\ndel string[0]\nstring"
```

In [33]:

```
"""data = 'abcdefg'
```

```
def func(text):
    del text[2]
    return text

print(func(data))"""

# func('abcdefg')
# -->
# TypeError:
# 'str' object doesn't support item deletion
```

Out[33]:

```
"data = 'abcdefg'\n \ndef func(text):\n    del text[2]\n    return text
\n \nprint(func(data))"
```

In [34]:

```
# Solución
# A
```

Question 11 (Control Flow)

Which of the following sentences correctly describes

the output of the below Python code?

```
data = [4, 2, 3, 2, 1]
res = data[0]

for d in data:
    if d < res:
        res = d

print(res)
```

A. res is the sum of all the number in the list

B. None of the above

C. res is the smallest number in the list

D. res is the average of all the number in the list

E. res is the largest number in the list

Solution 11

In [36]:

```
data = [4, 2, 3, 2, 1]
res = data[0]              # res = 4

for d in data:             # [4, 2, 3, 2, 1]
    if d < res:
        res = d
        # 4 => res = 4
        # 2 => res = 2
        # 3 => res = 3
        # 2 => res = 2
        # 1 => res = 1

print(res)  # res = 1


# res is the smallest number in the list
```

```
1
```

In [37]:

```
# Solución
# C
```

Question 12 (Functions)

The following snippet:

```
def function_1(a):
    return None

def function_2(a):
    return function_1(a) * function_1(a)

print(function_2(2))
```

A. will output `4`

B. will cause a runtime error

C. will output `2`

D. will output `16`

Solution 12

In [39]:

```
"""def function_1(a):
    return None

def function_2(a):
    return function_1(a) * function_1(a)

print(function_2(2))"""

# TypeError:
# unsupported operand type(s) for *: 'NoneType' and 'NoneType'
```

Out[39]:

```
'def function_1(a):\n    return None\n \ndef function_2(a):\n    return function_1(a) * function_1(a)\n \nprint(function_2(2))'
```

In [40]:
```
# Solución
# B
```

Question 13 (Data Types)

What is the expected output of the following code?

```
print(ord('c') - ord('a'))
```

A. `3`

B. `1`

C. `0`

D. `2`

Solution 13

In [42]:
```
ord('c'), ord('a')
```

Out[42]:
```
(99, 97)
```

In [43]:
```
ord('c')-ord('a')
```

Out[43]:
```
2
```

In [44]:
```
# Solución
# D
```

Question 14 (Operators)

Which of the following code snippets will print True to the monitor?

Dos repuestas correctas

A. `print('is' in 'This IS Python code.')`

B.

```
x = ['Peter', 'Paul', 'Mary']
y = ['Peter', 'Paul', 'Mary']
print(x is y)
```

C.

```
x = 'Peter Wellert'
y = 'Peter Wellert'.lower()
print(x is y)
```

D.

```
x = 42
y = 42
print(x is not y)
```

E. `print('t' in 'Peter')`

Solution 14

In [46]:

```
# A.
print('is' in 'This IS Python code.')
```

```
True
```

In [47]:

```
# A. CAMBIADA
print('is' in 'Ths IS Python code.')
```

```
False
```

In [48]:

```
# B
x = ['Peter', 'Paul', 'Mary']
y = ['Peter', 'Paul', 'Mary']
print(x is y)
```

```
False
```

In [49]:

```
# B. CAMBIADA
x = ['Peter', 'Paul', 'Mary']
y = ['Peter', 'Paul', 'Mary']
print(x is not y)
```

```
True
```

In [7]:

```
# D
x = 42
y = 42
print(x is not y)
```

```
False
```

In [51]:

```
# D. CAMBIADA
x = 42
y = 42
print(x is y)
```

```
True
```

In [52]:

```
# E
print('t' in 'Peter')
```

```
True
```

In [53]:

```
# Solución
# A y E
```

Question 15 (Data Aggregates)

What is the expected output of the following code?

```
w = [7, 3, 23, 42]
x = w[1:]
y = w[1:]
z = w
y[0] = 10
z[1] = 20
print(w)
```

A. `[10, 20, 42]`

B. `[10, 20, 23, 42]`

C. `[7, 20, 23, 42]`

D. `[7, 3, 23, 42]`

Solution 15

In [55]:

```
w = [7, 3, 23, 42]
print('w inicialmente es: ', w)      # w = [7, 3, 23, 42]

x = w[1:]
print('x es: ', x)                   # x = [   3, 23, 42]    desde index

y = w[1:]
print('y es : ',y)                   # y = [   3, 23, 42]    desde index
print('w aqui es: ', w)              # w = [7, 3, 23, 42]

z = w
print('z es: ', z)                   # z = [7, 3, 23, 42]
print('w es: ', w)                   # w = [7, 3, 23, 42]


y[0] = 10
print('y vale:', y)                  # y = [   10, 23, 42]   cambio un 3

# Nota:   !!!!!!
# PERO Y NO DICE EN NINGUN CASO: Y = W, POR LO QUE NO AFECTA A W
# en cambio, arriba dice Y = W

z[1] = 20
print('z vale:', z)                  # z = [7, 20, 23, 42]

# ----y si altero z estoy alterando w también -----
```

```
print('w FINAL vale:', w)            # w = [7, 20, 23, 42]
```

```
w inicialmente es:  [7, 3, 23, 42]
x es:  [3, 23, 42]
y es :  [3, 23, 42]
w aqui es:  [7, 3, 23, 42]
z es:  [7, 3, 23, 42]
w es:  [7, 3, 23, 42]
y vale: [10, 23, 42]
z vale: [7, 20, 23, 42]
w FINAL vale: [7, 20, 23, 42]
```

In [56]:

```
# Solución
# C
```

Question 16 (Operators)

What value will be assigned to the x variable?

```
z = 2
y = 1
x = y < z or z > y and y > z or z < y
```

A. `False`

B. `0`

C. `1`

D. `True`

Solution 16

In [58]:

```
z = 2
y = 1
x = y < z or z > y and y > z or z < y

# x = True or [True and False] or False
# x =  T   or       F          or  F
# x =  T

print(x) # True
```

```
True
```

In [59]:

```
# Solución
# D
```

Question 17 (Functions)

What is the output of the following snippet?

```
def func(x, y):
    if x == y:
        return x
    else:
        return func(x, y-1)

print(func(0, 3))
```

A. `1`

B. `0`

C. The snippet will cause a runtime error

D. `2`

Solution 17

In [61]:
```
def func(x, y):
    if x == y:
        return x
    else:
        return func(x, y-1)


print(func(0, 3))

# func(0, 3)  => return func(x, y-1)  => return func(0, 2)
# func(0, 2)  => return func(0, 1)
# func(0, 1)  => return func(0, 0)
# func(0, 0)  => return x => 0

# --EXPLICACIÓN TEÓRICA--
# va restando 1 en la y
# hasta que se encuentra con x=0, y=0
# y en ese momento x==y devuelve True
# y retorna x, que es 0
```

```
0
```

In [62]:
```
# Solución
# B
```

Question 18 (Functions)

What is the output of the following snippet?

```
def any():
    print(var + 1, end='')

var = 1
any()
print(var)
```

A. `12`

B. `21`

C. `22`

D. `11`

Solution 18

In [64]:

```
# any no debería ser nombre de una función

def any():
    print(var + 1, end='')

var = 1

any()       # llamada a la función => 1+1=2
print(var)  # var sigue siendo 1

# => 21 (sin espacios por el "end")
```

```
21
```

In [65]:

```
# Solución
# B
```

unos ejemplos similares, con y sin variable GLOBAL, etc..

In [66]:

```
# any no debería ser nombre de una función, siendo palabra reservada
# pero, en todo caso..
```

In [67]:

```
# ejemplo 1: cambiando nomnbre de función por funcion_any
```

In [68]:

```
x = 1
def funcion_any():
    print(x + 5, end='')

funcion_any()
print(x)
```

```
61
```

In [69]:

```
# ejemplo 2: variable local x
```

In [70]:

```
"""
x = 1

def funcion_any():
    x = x + 5
    print(x, end='')

funcion_any()
print(x)
"""
# UnboundLocalError: local variable 'x' referenced before assignment

# AL SER LOCAL, la puede ejecutar, pero NO OPERAR.
```

Out[70]:

```
"\nx = 1\n\ndef funcion_any():\n    x = x + 5\n    print(x, end='')\n\nf
uncion_any()\nprint(x)\n"
```

In [71]:

```
# ejemplo 3: operarla si que lo hace
```

In [72]:

```
x = 1

def funcion_any():
    print(x, end='')

funcion_any()
print(x)
```

```
11
```

In [73]:

```
# ejemplo 4: variable global x
```

In [74]:

```
x = 1
def funcion_any():
    global x
    x = x + 5
    print(x, end='')

funcion_any()
print(x)
```

```
66
```

Question 19 (Data Aggregates)

What is the output of the following snippet?

```
my_list = [0, 1, 2, 3]
x = 1
for elem in my_list:
    x *= elem
print(x)
```

A. `0`

B. `6`

C. `1`

Solution 19

In [76]:

```python
my_list = [0, 1, 2, 3]
x = 1
for elem in my_list:
    x *= elem
    # x = x * elem
    # x = 1 * 0     => x = 0
    # x = 0 * 1     => x = 0
    # y asi hasta el final

print(x)  # 0
```

```
0
```

In [77]:

```python
# Solución
# A
```

Question 20 (Control Flow)

Consider the following programm to calculate a discount percentage:

```python
day = input('Enter the day of the week:')
discount = 3

if day == 'Wednesday':
    discount += 5
elif day == 'Thursday':
    discount += 7
elif day == 'Saturday':
    discount += 10
elif day == 'Sunday':
    discount += 20
else:
    discount += 2
```

Which of the following inputs will get the user a discount of 5 %?

A. `Thursday`

B. `Saturday`

C. `Sunday`

D. `Friday`

E. `Wednesday`

Solution 20

In [79]:

```
day = input('Enter the day of the week: ')
discount = 3

if day == 'Wednesday':
    discount += 5
elif day == 'Thursday':
    discount += 7
elif day == 'Saturday':
    discount += 10
elif day == 'Sunday':
    discount += 20
else:                  # Monday, Tuesday, Friday
    discount += 2

# de las soluciones el único posible es Friday

# imprimimos el descuento
print(discount, " %")
```

```
Enter the day of the week: Friday
5  %
```

In [80]:

```
# Solución
# D
```

Question 21 (Data Types)

You are an intern for ABC electric cars company.

You must create a function that calculates the average velocity of their vehicles on a 1320 foot (1/4 mile) track.

Consider the following code.

```
distance = ???(input('Enter the distance travelled in
feet'))
distance_miles = distance/5280  # convert to miles

time = ???(input('Enter the time elapsed in seconds'))
time_hours = time/3600  # convert to hours

velocity = distance_miles/time_hours
print('The average Velocity : ', velocity, 'miles/hour')


The output must be as precise as possible.
What would you insert instead of ??? and ???
```

A.

```
float
int
```

B.

```
float
float
```

C.

```
int
float
```

D.

```
int
int
```

Solution 21

In [82]:

```
distance = float(input('Enter the distance travelled in feet: '))   # 30.
distance_miles = distance/5280  # convert to miles

time = float(input('Enter the time elapsed in seconds: '))          # 120
time_hours = time/3600  # convert to hours

velocity = distance_miles/time_hours
print('The average Velocity : ', velocity, 'miles/hour')
```

```
Enter the distance travelled in feet: 30000
Enter the time elapsed in seconds: 120
The average Velocity :  170.45454545454544 miles/hour
```

In [83]:

```
# Solución
# B

# YO PUSE FEET COMO INT, REVISARLO!!!
```

Question 22 (Control Flow)

What is the expected output of the following code?

```
my_list = [[3-i for i in range(3)] for j in range(3)]
result = 0

for i in range(3):
    result += my_list[i][i]
```

```
print(result)
```

A. `7`

B. `2`

C. `4`

D. `6`

Solution 22

In [85]:

```
my_list = [[3-i for i in range(3)] for j in range(3)]

# range(3) => 0, 1, 2
# 3-i => 3, 2, 1
# [[3,2,1] for j in range(3)]
# [[3,2,1] for 0, for 1, y for 2]

print(my_list)

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

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

Out[85]:

```
'\n[\n[3,2,1],\n[3,2,1],\n[3,2,1]\n]\n'
```

In [86]:

```
my_list = [[3-i for i in range(3)] for j in range(3)]

# range(3) => 0, 1, 2
# 3-i => 3, 2, 1
# [[3,2,1] for j in range(3)]
# [[3,2,1] for 0, for 1, y for 2]

result = 0

for i in range(3):
    result += my_list[i][i]
    # result = result + my_list[i][i]
    # my_list[0][0] = 3
    # result =    0     +     3           => result = 3
    #           INICIAL    LISTA EN[0][0]

    # my_list[1][1] = 2
    # result =    3     +     2           => result = 5

    # my_list[2][2] = 1
    # result =    5     +     1           => result = 6

print(result)    # 6
```

```
6
```

```
In [87]: # Solución
         # D
```

Question 23 (Operators)

What is the output of the following code?

```
x = "2"
y = 2 * x
print(y)
```

A. `4`

B. The program will cause an error

C. `22`

D. `2x`

Solution 23

```
In [89]: x = "2"
         y = 2 * x      # string * un número


         print(y)    # 22   [22 string]
```

```
22
```

```
In [90]: # comprobamos que es un string
         type(y)
```

```
Out[90]: str
```

```
In [91]: # Solución
         # C
```

recordatorio

```
In [92]: string = '2x'
         string
```

```
Out[92]: '2x'
```

```
In [93]: 3*string
```

```
Out[93]: '2x2x2x'
```

```
In [94]: string*3
```

```
Out[94]: '2x2x2x'
```

Question 24 (Data Aggregates)

What is the expected output of the following code?

```
data = ()
print(data.__len__())
```

A. `0`

B. `None`

C. The code is erroneous

D. `1`

Solution 24

```
In [8]: data = ()
        print(data.__len__())
```

```
0
```

```
In [97]: # es equivalente a len(data) !!!
```

```
In [98]: data = ()
         print(len(data))
```

```
0
```

```
In [99]: # Solución
         # A
```

Question 25 (Control Flow)

What is the expected output of the following code?

```
data = [1, {}, (2,), (), {3}, [4, 5]]
points = 0

for i in range(len(data)):
    if type(data[i]) == list:
        points += 1
    elif type(data[i]) == tuple:
        points += 10
    elif type(data[i]) == set:
        points += 100
    elif type(data[i]) == dict:
        points += 1000
    else:
        points += 10000

print(points)
```

A. `10221`

B. `21102`

C. `11121`

D. `10212`

E. `11112`

Solution 25

In [101…

```
data = [1, {}, (2,), (), {3}, [4, 5]]
data
```

Out[101…

```
[1, {}, (2,), (), {3}, [4, 5]]
```

In [102…

```
data = [1, {}, (2,), (), {3}, [4, 5]]
len(data)
```

Out[102…

```
6
```

In [103…

```
# el propio ejercicio
```

In [104…

```
data = [1, {}, (2,), (), {3}, [4, 5]]
points = 0

for i in range(len(data)):                  # range(6) => 0-1-2-3-4-5
    if type(data[i]) == list:
        points += 1
    elif type(data[i]) == tuple:
        points += 10
    elif type(data[i]) == set:
        points += 100
    elif type(data[i]) == dict:
        points += 1000
    else:
        points += 10000

print(points)
```

```
11121
```

In [105…

```
# Solución
# C
```

resolución paso a paso printeando

In [106…

```python
data = [1, {}, (2,), (), {3}, [4, 5]]
points = 0

for i in range(len(data)):                 # range(6) => 0-1-2-3-4-5
    if type(data[i]) == list:
        print('Es una lista:', data[i])
        print('\n')
        points += 1
    elif type(data[i]) == tuple:
        print('Es una tupla: ', data[i])
        print('\n')
        points += 10
    elif type(data[i]) == set:
        print('Es un set: ', data[i])
        print('\n')
        points += 100
    elif type(data[i]) == dict:
        print('Es un diccionario: ', data[i])
        print('\n')
        points += 1000
    else:
        print('NI LISTA, TUPLA, SET O DICIONARIO: ', data[i])
        print(data[i])
        print('\n')
        points += 10000

print(points)


"""
[1,       {},    (2,),     (),     {3},      [4, 5]]
10000 +  1000 +   10  +    10   +  100  +      1
                                   SET

{3} NO ES DICCIONARIO! ES SET, NO TIENE {'CLAVE', VALOR}
"""
```

```
NI LISTA, TUPLA, SET O DICIONARIO:  1
1


Es un diccionario:  {}


Es una tupla:  (2,)


Es una tupla:  ()


Es un set:  {3}


Es una lista: [4, 5]


11121
```

```
Out[106… "\n[1,       {},    (2,),     (),     {3},      [4, 5]]\n10000 +  1000 +
          10  +    10   +  100  +      1\n                                 SET\n
          \n{3} NO ES DICCIONARIO! ES SET, NO TIENE {'CLAVE', VALOR}\n"
```

```
In [107… 10000 +  1000 +   10  +    10   +  100  +      1
```

```
Out[107… 11121
```

Question 26 (Operators)

What is the expected output of the following code?

```
x = 4.5
y = 2

print(x // y)
```

A. `2.5`

B. `2.0`

C. `2`

D. `2.25`

Solution 26

```
In [109… x = 4.5

         y = 2

         print(x // y)  # COCIENTE-> 2.0

         # ES COCIENTE ENTERO, PERO UNO DE LOS NÚMEROS ES DECIMAL EN LA DIVISIÓN
         # POR ESO NO ES 2, SINO, 2.0
```

```
2.0
```

```
In [110… # Solución
         # B
```

Question 27

What is the expected output of the following code?

```
num = 2 + 3 * 5
print(Num)
```

A. `17`

B. `17.0`

C. `25`

D. The code is erroneous

Solution 27

In [112…

```
"""
num = 2 + 3 * 5     # 17
print(Num)          # NO EJECUTA
"""

# NameError: name 'Num' is not defined



# OJO con la N mayúscula!
# no son la misma variable: "num" y "Num"
```

Out[112…

```
'\nnum = 2 + 3 * 5     # 17\nprint(Num)          # NO EJECUTA\n'
```

In [113…

```
# Solución
# D
```

Question 28 (Operators)

What is the output of the following snippet if the user enters two lines containing 3 and 6 respectively?

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

print(x * y)
```

A. `36`

B. `18`

C. `333333`

D. `666`

Solution 28

```
In [115…  # x = 3
          # y = 6
          x = input()         # 3 string
          y = int(input())    # 6 entero

          print(x * y)        # 3 (6 veces)     '3' 6 veces ==> 333333  (string)
```

```
3
6
333333
```

```
In [116…  type(x*y)
```

```
Out[116…  str
```

```
In [117…  # Solución
          # C
```

Question 29 (Basics)

You develop a Python application for your company.

You want to add notes to your code so other team members will understand it.

What should you do?

A. Place the notes inside of parentheses on any line

B. Place the notes after the last line of code separated by a blank line

C. Place the notes before the first line of code separated by a blank line

D. Place the notes after the `#` sign on any line

Solution 29

```
In [119…  # Solución
          # D
```

Question 30

What is the expected output of the following code?

```
print(2 ** 3 ** 2 ** 1)
```

A. `16`

B. `512`

C. `128.0`

D. `64`

E. `16.0`

F. The code is erroneous

Solution 30

In [121…

```
print(2 ** 3 ** 2 ** 1)
#               2 ** 1 => 2
#          3 **    2    => 9
#     2 **    9         => 512


# recordar que se toman de derecha a izquierda
# ojo, con paréntesis, multiplicaciones y divisiones, y las sumas y resta

# pero en este caso, es un único operador. se operan de DERECHA A IZQUIER
```

```
512
```

In [122…

```
# Solución
# B
```

Gracias por la atención

Isabel Maniega