

2. Tipos y Operadores en Python - Actividades.

Los siguientes ejercicios están tomados del libro digital “Introducción a la Programación con Python 3”, y se podrán realizar en el Entorno Interactivo de Ejecución de Python.

14 - ¿Qué resultados se obtendrán al evaluar las siguientes expresiones Python? Calcula a mano el valor resultante de cada expresión y comprueba, con la ayuda del ordenador, si tu resultado es correcto:

- A. $2 + 3 + 1 + 2$
- B. $2 + 3 * 1 + 2$
- C. $(2 + 3) * 1 + 2$
- D. $(2 + 3) * (1 + 2)$
- E. $+-6$
- F. $-++6$
- G. $-3 / 2 - 1$
- H. $-3 // 2 - 1$

15 - Traduce las siguientes expresiones matemáticas a Python y evalúalas. Trata de utilizar el menor número posible de paréntesis

a) $2 + (3 \cdot (6/2))$

b) $\frac{4+6}{2+3}$

c) $(4/2)^5$

d) $(4/2)^{1+2^2}$

e) $(-3)^2$

f) $-(3^2)$

19 - Qué resulta de ejecutar estas tres líneas?

```
>>> x = 10
>>> x = x * 10
>>> x
```

20 - Evalúa el polinomio $x^4 + x^3 + 2x^2 - x$ en $x=1.1$. Utiliza variables para evitar teclear varias veces el valor de x (El resultado es 4.1151)

22 - Que valor tiene z tras evaluar estas sentencias?

```
>>> z = 2
>>> z += 2
>>> z += 2 - 2
>>> z *= 2
>>> z *= 1 + 1
>>> z /= 2
>>> z %= 3
>>> z /= 3 - 1
>>> z -= 2 + 1
>>> z -= 2
>>> z **= 3
>>> z
```

23 - Evalúa estas expresiones y sentencias en el mismo orden en el que aparecen e indica lo que muestra el intérprete de Python como respuesta

```
>>> a = 'b'
>>> a + 'b'
>>> a + 'a'
>>> a * 2 + 'b' * 3
>>> 2 * (a + 'b')
>>> 2 * ('a' + 'b')
```

24 - Que resultado se obtendrán al evaluar las siguientes expresiones y asignaciones Python? Calcula primero a mano el valor resultante de cada expresión y comprueba, con la ayuda del ordenador, si tu resultado es correcto.

```
>>> 'a' * 3 + '/' * 5 + 2 * 'abc' + '+'
>>> palindromo = 'abcba'
>>> (4 * '<' + palindromo + '>' * 4) * 2
>>> subcadena = '=' + '-' * 3 + '='
>>> '10' * 5 + 4 * subcadena
>>> 2 * '12' + '.' + '3' * 3 + 'e-' + 4 * '76'
```

25 - Identifica regularidades en las siguientes cadena, y escribe expresiones que, partiendo de subcadenas más cortas y utilizando los operadores de concatenación y repetición, produzcan las cadenas que se muestra. Introduce variables para formar las expresiones cuando lo consideres oportuno.

- A. '%%%%. / . / <-><->'
- B. '(@)(@)(@)=====(@)(@)(@)====='
- C. 'asdfasdfasdf=---=---=---=?????asdfasdf'
- D. '.....*****_*****_.....*****_*****_'

28 - Que resulta de evaluar estas expresiones?

```
>>> str(2.1) + str(1.2)
'2.11.2'
>>> int(str(2) + str(3))
23
>>> str(int(12.3)) + '0'
'120'
>>> int('2'+ '3')
23
>>> str(2+3)
'5'
>>> str(int(2.1) + float(3))
'5.0'
```

30 - Que resultados se obtendrán al evaluar las siguientes expresiones Python? Calcula primero a mano el valor resultante de cada expresión y comprueba, con la ayuda del intérprete por línea de comandos si tu resultado es correcto

- A. `int(exp(2 * log(3)))`
- B. `round(4*sin(3 * pi / 2))`
- C. `abs(log10(.01) * sqrt(25))`
- D. `round(3.21123 * log10(100), 3)`