

<b>Comenzado el</b>	lunes, 27 de marzo de 2023, 21:27
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	lunes, 27 de marzo de 2023, 21:39
<b>Tiempo empleado</b>	12 minutos 45 segundos
<b>Puntos</b>	29/30
<b>Calificación</b>	10 de 10 (97%)

Pregunta **1**

Correcta

Se puntúa 1 sobre 1

Para cada una de las personas célebres cuyos nombres aparecen a la izquierda, seleccione el aporte principal que dicha persona ha realizado al mundo de las ciencias informáticas o las ciencias exactas.



Ada Byron

Primeros conceptos fundamentales de programación (subrutinas, ciclos, etc.)



Thomas Flowers

Diseñador de Colossus, la primera máquina operable considerada como antecedente de las computadoras modernas.



Charles Babbage

Diseño de la Analytical Engine (primer diseño práctico de una computadora en el mundo)



Abu Abdallah Muḥammad ibn  
Mūsā al-Jwārizmī (Abu Yāffar)

Primeras reglas algorítmicas para las operaciones aritméticas elementales en números arábigos.



Alan Turing

Director del equipo que desarrolló Bombe, la máquina que permitió descifrar el código Enigma alemán.



¡Ok!

Pregunta **2**

Correcta

Se puntúa 2 sobre 2

¿Cuál es el problema (si lo hay) si se ejecuta el siguiente script en *Python 3*?

```
n1 = 10
n2 = 14
n1 = int(input('Ingrese un número entero: '))
n2 = float(input('Ingrese un número en coma flotante: '))
print('n1: ', n1)
print('n2: ', n2)
n2 = input('Ahora ingrese su nombre: ')
print('Felicitaciones', n2, 'ha terminado el ejercicio')
```

Seleccione una:

- ☒ a. No hay ningún problema. ¡Ok! Si tuvo alguna duda por el hecho de que la variable *n2* comenzó con un valor *int* y luego se le asignó un valor *float*, y al final cambió de nuevo a un valor de tipo cadena, no olvide que Python es un lenguaje de tipado dinámico, y por lo tanto una variable puede cambiar de tipo durante la ejecución de un programa. Otra historia es que este programa tenga sentido... ¡Pero eso no es lo que estaba en discusión!
- ☐ b. En Python 3 no hay ninguna función llamada *float()* para convertir cadenas a números flotantes.
- ☐ c. La variable *n2* se definió como *int* al asignarle el valor inicial 14, y luego se le asignó un valor *float* cargado por teclado, y al final se volvió cargar por teclado otro valor en *n2* pero ahora de tipo cadena: no se puede cambiar el tipo de una variable, y menos si era numérica y se pretende que cambie a cadena.
- ☐ d. Si una variable ya fue asignada con un valor, no se puede cambiar ese valor por otro cargado por teclado.

¡Correcto!

Pregunta **3**

Correcta

Se puntúa 2 sobre 2

Suponga la siguiente instrucción de carga por teclado en Python 3:

```
x = float(input('Ingrese un numero: '))
```

¿Cuál de las siguientes afirmaciones es **CIERTA**?

Seleccione una:

- ☐ a. Si se ingresa por teclado un valor que no puede convertirse a un número, la variable *x* quedará valiendo *None*.
- ☒ b. Si se ingresa por teclado un valor que no puede convertirse a un número, se producirá un error y la ejecución del script se interrumpirá. ¡Ok!
- ☐ c. Si se ingresa por teclado un número entero, la variable *x* quedará valiendo el valor *None*.
- ☐ d. Si se ingresa por teclado número entero, se producirá un error y la ejecución del script se interrumpirá.

¡Correcto!

Pregunta **4**

Correcta

Se puntúa 1 sobre 1

¿Qué significa **definir** una variable en Python?

Seleccione una:

- ☐ a. Indicar su tipo y su nombre o identificador.
- ☐ b. Indicar su tipo, su nombre y su tamaño.
- ☐ c. Indicar su nombre.
- ☒ d. Indicar su nombre y asignarle un valor. ✓ ¡Ok!

¡Correcto!

Pregunta **5**

Correcta

Se puntúa 2 sobre 2

¿Hay algún problema con el siguiente script en *Python 3*?

```
y = x + 15
x = 8
print('Valor final x:', x)
print('Valor final y:', y)
```

Seleccione una:

- ☐ a. Está mal realizada la visualización del resultado: en Python 3 *print* no debe escribirse con paréntesis.
- ☐ b. No hay ningún problema.
- ☐ c. La expresión `y = x + 15` no tiene sentido en Python.
- ☒ d. Lanza un error: la variable `x` no está definida en el momento en que se le suma el número 15. ✓ ¡Ok! No importa si se le asigna un valor después. Al momento de usarla, su valor debe estar ya asignado.

¡Correcto!

Pregunta **6**

Incorrecta

Se puntúa 0 sobre 1

¿Qué diferencia principal hay entre una *calculadora manual común* y una *computadora*?

(Tómese su tiempo para pensar y discutir esta pregunta... No encontrará la respuesta directamente en la Ficha 01).

Seleccione una:

- ☐ a. Ninguna.
- ☐ b. Las calculadoras manuales comunes no pueden procesar texto ni otros tipos de datos no numéricos. Las computadoras sí.
- ☐ c. Las computadoras son programables, mientras que las calculadoras no.
- ☒ d. Las calculadoras no pueden componer ni desplegar imágenes, mientras que las computadoras sí.

Incorrecto... Las calculadoras normalmente están pensadas para disponer de una pequeña pantalla que muestre números, pero podrían disponer de pantallas con mejor resolución (de hecho, las "calculadoras que grafican funciones" las tienen, si bien en muchos casos esas "calculadoras gráficas" son en realidad computadoras de bolsillo).

La lectura de las dos primeras secciones de la Ficha 01 podría darle pistas sobre la respuesta correcta, pero lo mejor sería discutirlo con sus compañeros y/o sus profesores.

Pregunta **7**

Correcta

Se puntúa 1 sobre 1

¿Qué se entiende, en general, por *error de compilación*?

Seleccione una:

- ☐ a. Es un error en la *lógica* del programa, que provoca que al ejecutarse el programa arroje resultados incorrectos.
- ☐ b. Es un error en el *hardware de la computadora*, que provoca una falla grave de funcionamiento de todos los programas.
- ☒ c. Es un error en la *sintaxis* del programa, que provoca que el programa no pueda comenzar a ejecutarse (si es compilado) o no pueda seguir ejecutándose (si es interpretado) al llegar a la línea con ese error. ¡Ok!
- ☐ d. Es un error producido por una *operación imposible de ejecutar*, aunque sintácticamente bien escrita (por ejemplo, una división por cero), que provoca que el programa se interrumpa de forma abrupta y anormal una vez que comenzó a ejecutarse

¡Correcto!

Pregunta **8**


Correcta

Se puntúa 2 sobre 2

¿Hay algún error en el siguiente script de instrucciones en Python 3?

```
nombre = input('Nombre: ')
edad = int(input('Edad: '))
antiguedad = Edad + 1
print('Datos recibidos - Nombre: ', nombre, 'Edad: ', edad, 'Antiguedad:', antiguedad)
```

Seleccione una:

- ☐ a. El error es que la función print() de Python 3 no puede usarse mostrar al mismo tiempo más de dos variables (acompañadas de sus respectivos mensajes).
- ☐ b. El error es el uso de la función int() en la segunda carga: no existe tal función en Python 3.
- ☒ c. El error es que la variable *edad* se definió en minúsculas al hacer la carga, y luego se usó con  ¡Ok! Las variables *edad* y *Edad* son dos variables distintas en Python.
- ☐ d. No hay ningún error.

¡Correcto!

Pregunta **9**


Correcta

Se puntúa 2 sobre 2

Dado un algoritmo, llamamos *instrucciones primitivas* o *acciones primitivas* a aquellos pasos mínimos del algoritmo que necesariamente debe saber aplicar quien ejecute el algoritmo (por ejemplo, para hacer una suma de dos números de varios dígitos, las operaciones primitivas mas básicas son alinear los números hacia la derecha, y sumar números de un dígito).

Suponga que se quiere plantear un algoritmo para dibujar un tablero de ajedrez (sin las fichas... SÓLO el tablero). ¿Cuál de las siguientes opciones describe **mejor** el conjunto de acciones primitivas que sería necesario aplicar?

Seleccione una:

- ☐ a. { Dibujar cuadrados (sólo el contorno) }
- ☐ b. { Dibujar triángulos (solo el contorno) }
- ☐ c. { Dibujar líneas rectas horizontales, Dibujar líneas rectas verticales }
- ☒ d. { Dibujar cuadrados, Pintar por dentro un cuadrado con un color dado }  ¡Ok!

¡Correcto!

Pregunta **10**

Correcta

Se puntúa 1 sobre 1

¿Cuáles son los motivos por los cuales una persona que sabe resolver un problema, querría programar y usar una computadora para resolverlo?

Seleccione una:

- ☒ a. Porque al programar una computadora para resolver el problema, ganará tiempo y ahorrará esfuerzo en el futuro: la computadora puede obtener las soluciones muy rápidamente, y con precisión. ✓ ¡Ok!
- ☐ b. No hay motivos para que lo haga: Si sabe resolver el problema, no necesita una computadora y no hay motivo para usarla.
- ☐ c. Porque al programar una computadora, tendrá la garantía de una solución correcta.
- ☐ d. Porque sólo programando una computadora obtendrá soluciones numéricamente precisas y sin errores ni pérdida de precisión por valores decimales.

¡Correcto!

Pregunta **11**

Correcta

Se puntúa 1 sobre 1

¿Qué relación existe entre los conceptos de algoritmo y programa?

Seleccione una:

- ☐ a. Un programa es un algoritmo que sólo puede ser interpretado por una persona.
- ☐ b. Son exactamente lo mismo.
- ☒ c. Un programa es un algoritmo que puede ser interpretado y ejecutado por un computador. ✓ ¡Ok!
- ☐ d. Ninguna relación.

¡Correcto!

Pregunta **12**

Correcta

Se puntúa 2 sobre 2

¿Hay algún inconveniente en el siguiente script elemental de Python? (Suponga que no hay otras instrucciones previas al script mostrado)

```
valor = 5
print(valor)
valor = 'mundo'
print(valor)
valor = True
print(valor)
```

Seleccione una:

- ☐ a. Producirá un error al intentar ejecutar la tercera línea: `valor = 'mundo'`
- ☒ b. No hay ningún problema. ✓ ¡Correcto!
- ☐ c. Producirá un error al intentar ejecutar la quinta línea: `valor = True` porque está cambiando el tipo de la variable (de *str* a *bool*).
- ☐ d. Producirá un error al intentar ejecutar la última línea: `print(valor)`
- ☐ e. Producirá un error al intentar ejecutar la quinta línea: `valor = True` porque el valor *True* no existe en Python.

¡Correcto!

Pregunta **13**

Correcta

Se puntúa 2 sobre 2

¿Hay algún error en la siguiente secuencia de instrucciones en Python?

```
b = None
c = 40
a = c - b + 5
print(a)
```

Seleccione una:

- ☐ a. No hay error alguno.
- ☒ b. La variable *b* está definida, pero con el valor *None* cuando se ejecuta la tercera línea. La suma no puede ejecutarse y lanza un ✓ ¡Ok! error.
- ☐ c. La constante *None* no tiene ningún significado y no existe en Python. Lanza un error en la primera línea.
- ☐ d. No se pueden hacer operaciones de sumas y restas combinadas en una misma instrucción.

¡Correcto!

Pregunta **14**

Correcta

Se puntúa 2 sobre 2

¿Cuál es el valor que terminará valiendo la variable **res** luego del siguiente bloque de instrucciones?

```
a = 23
b = 5
res = a // b
```

Respuesta:



¡Ok! El operador `//` (doble barra) efectivamente calcula el *cociente entero* y los decimales se truncan.

Pregunta **15**

Correcta

Se puntúa 2 sobre 2

¿Cuál es el valor que termina valiendo la variable **res** luego de la siguiente secuencia, en la que se usa el operador *resto* o *módulo* de una división?

```
x1 = 8
x2 = 10
res = x1 % x2
```

Respuesta:



¡Ok! El resto de la división entre `x1` y `x2` es igual al mismo valor `x1` si se da el caso que `x2` es mayor que `x1`. Revise la ficha 01 si no le queda claro, pregunte en clase si sigue sin quedar claro, y siempre pruebe a ejecutar este script y ver qué pasa...

Pregunta **16**

Correcta

Se puntúa 2 sobre 2

¿Qué valor queda valiendo la variable **p** luego de la siguiente secuencia de instrucciones en Python?

```
p = 7
q = 1
q = p
```

Respuesta:



¡Correcto!



Pregunta **17**

Correcta

Se puntúa 1 sobre 1

En general, una **expresión** es una fórmula en la cual se usan *operadores* (como suma, resta, producto, comparación, etc.) sobre diversas variables y constantes (que reciben el nombre de *operandos* de la expresión). Son ejemplos válidos los siguientes:  $3 * a + 2$ ,  $b / c - 4$ ,  $(7 - r) / (4 + a)$ ,  $a > b$ ,  $x + 2 >= 10$ .

¿Es correcta la siguiente definición?

"Una **expresión aritmética** es una expresión en la cual el resultado final es un número"

Seleccione una:

- ☒ Verdadero ✓
- ☐ Falso

¡Correcto!

Pregunta **18**

Correcta

Se puntúa 1 sobre 1

¿Es posible que la misma persona que diseña un algoritmo sea también quien ejecute ese algoritmo?

Seleccione una:

- ☒ Verdadero ✓
- ☐ Falso

¡Correcto!

Pregunta **19**

Correcta

Se puntúa 1 sobre 1

¿Puede decirse que un proceso planteado para que tenga un comienzo en un momento dado pero de tal forma de no detenerse jamás, *es un algoritmo*?

Seleccione una:

- ☐ Verdadero
- ☒ Falso ✓

¡Correcto!

Pregunta **20**

Correcta

Se puntúa 1 sobre 1

Suponga que se le pide desarrollar un programa que muestre en pantalla todos y cada uno de los números naturales (todos los enteros positivos) ¿Puede hacerse un programa así?

Seleccione una:

☐ Verdadero

☒ Falso ✓

¡Correcto!

[◀ Video Motivacional 01: Lo que muchas instituciones no enseñan](#)

Ir a...



[Guía de Ejercicios Prácticos - Ficha 01 ▶](#)