

Prueba del Módulo 1

¡Bien hecho! Has llegado al final del Módulo 1 y has completado un paso importante en su educación en programación Python. Aquí hay un breve resumen de las áreas temáticas que has cubierto en el Módulo 1:

- trabajar con módulos en Python; importar, crear y usar módulos;
- emplear módulos STL de Python como (*math*, *random*, y *platform*)
- construir y usar paquetes en Python;
- PIP (el Instalador de Paquetes de Python).

Ahora estás listo para tomar la **prueba del módulo**, que te ayudará a evaluar lo que has aprendido hasta ahora.

La siguiente prueba se basa en lo que acabas de aprender. Hay dieciocho preguntas en total y debes obtener al menos un 70 % para aprobar.

¡Buena suerte!

Pregunta 1

Sabiendo que una función llamada `fun()` reside dentro de un módulo llamado `mod`, selecciona la forma correcta de importarlo.

`import fun`

`import fun from mod`

`from fun import mod`



`from mod import fun`

Pregunta 2

Sabiendo que una función llamada `fun()` reside dentro de un módulo llamado `mod`, y se ha importado usando la siguiente línea:

```
import mod
```

Selecciona la forma en que se puede invocar desde tu código.

`fun()`

`mod::fun()`



`mod.fun()`

`mod->fun()`

Pregunta 3

Una función que devuelve una lista de todas las entidades disponibles en un módulo lleva por nombre:

`listmodule()`



`dir()`

`content()`

`entities()`

Pregunta 4

Un archivo *pyc* contiene:



código compilado de Python

un intérprete de Python

un compilador de Python

código fuente de Python

Pregunta 5

Cuando se importa un módulo, su contenido:

puede ser ejecutado (explícitamente)



se ejecuta una vez (implícitamente)

se ejecuta tantas veces como se importe

es ignorado

Pregunta 6

Una variable predefinida de Python que almacena el nombre del módulo actual lleva por nombre:

`__module__`

`__mod__`

`__modname__`



`__name__`

Pregunta 7

La siguiente línea de código:

```
from a.b import c
```

causa la importación de:

la entidad **a** del módulo **b** del paquete **c**



la entidad **c** del módulo **b** del paquete **a**

la entidad **b** del módulo **a** del paquete **c**

la entidad **c** del módulo **a** del paquete **b**

Pregunta 8

¿Cuál es el valor esperado asignado a la variable `result` después de que se ejecute el siguiente código?

```
1 import math
2
3 result = math.e != math.pow(2, 4)
4 print(int(result))
5
```



1

True

False

0

Pregunta 9

¿Cuál es el resultado esperado del siguiente código?

```
1 from random import randint
2
3 for i in range(2):
4     print(randint(1, 2), end='')
5
```



11, 12, 21, o 22

Existen millones de combinaciones posibles y no se puede predecir el resultado exacto.

12

12, o 21

Pregunta 10

Selecciona las sentencias verdaderas. (Selecciona dos respuestas)



La función `system` del módulo `platform` devuelve una cadena con el nombre del sistema operativo.

La función `processor` del módulo `platform` devuelve un número entero con la cantidad de procesos que se están ejecutando actualmente en tu sistema operativo.



La función `version` del módulo `platform` devuelve una cadena con la versión de tu instalación de Python.

La función `version` del módulo `platform` devuelve una cadena con la versión de tu sistema operativo.

Pregunta 11

Durante la primera importación de un módulo, Python despliega los archivos `.pyc` en el directorio llamado:



`__pycache__`

`__modules__`

`__init__`

`__hashbang__`

Pregunta 12

El conjunto de caracteres escrito como `#!` se emplea para:

hacer que una entidad de módulo en particular sea privada

crear un docstring (cadena de documentación)

decirle a un sistema operativo MS Windows cómo ejecutar el contenido de un archivo Python



decirle a un sistema operativo Unix o similar a Unix cómo ejecutar el contenido de un archivo Python

Pregunta 13

Se puede obtener una lista de las dependencias de los paquetes en pip empleando el comando:

`deps`

`dir`



`show`

`list`

Pregunta 14

El comando `pip list` presenta una lista de:

comandos pip disponibles

paquetes locales obsoletos

☒ paquetes instalados localmente

todos los paquetes disponibles en PyPI

Pregunta 15

¿Cuáles de las siguientes sentencias son verdaderas acerca del comando `pip search`? (Selecciona dos respuestas)

Busca solo a través de los nombres de los paquetes.

Todas sus búsquedas están limitadas a paquetes instalados localmente

☒ Necesita una conexión a Internet para funcionar

☒ Busca en todos los paquetes de PyPI.

Pregunta 16

¿Cuáles de las siguientes sentencias son verdaderas acerca del comando `pip install`? (Selecciona dos respuestas)

☒ Instala un paquete por usuario cuando la opción `--user` es especificada.

☒ Permite al usuario instalar una versión específica del paquete.

Siempre instala la versión más reciente del paquete y eso no se puede cambiar.

Instala un paquete en todo el sistema cuando la opción `--system` es especificada.

Pregunta 17

¿Cuál de las siguientes sentencias es verdadera acerca de la actualización de paquetes de Python ya instalados?

☒ Es realizada por el comando `install` acompañado de la opción `-U`.

Es un proceso automático que no requiere la atención del usuario.

Solo se puede hacer desinstalando e instalando los paquetes una vez más.

Se puede hacer reinstalando el paquete usando el comando `reinstall`.

Pregunta 18

¿Qué comando de `pip` se puede emplear para eliminar un paquete instalado?

`pip --uninstall package`

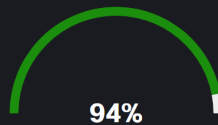
`pip install --uninstall package`



`pip uninstall package`

`pip remove package`

[Revisar Evaluación](#)



Has obtenido un 94%.

Felicidades, has acreditado la prueba.

