**Falso / verdadero**

* Los programadores raramente definen sus propias funciones: Falso. Los programadores a menudo usan funciones para reducir códigos duplicados y para hacer un programa mejor organizado y más modular. (p.167)
* Una función solo puede invocarse en un lugar de un programa: Falso. Una función puede ser llamada en muchos puntos diferentes en un programa. (página 169)

* La información se puede pasar a una función a través de parámetros: Cierto.
* Cada función de Python devuelve algún valor: Cierto. Si no devolvemos explícitamente un valor con una declaración de devolución, se devuelve el valor Ninguno. (p.183)
* En Python, algunos parámetros se pasan por referencia: Falso. Python no permite pasar parámetros por referencia. Python pasa los parámetros "por valor". (p.186)
* En Python, una función puede devolver solo un valor: Falso. Las funciones pueden devolver más de un valor al enumerar las expresiones en la declaración de devolución. (p.182)
* Las funciones de Python nunca pueden modificar un parámetro: Falso. Las funciones de Python pueden modificar objetos mutables como listas y objetos. (p 187-189)
* Una razón para usar funciones es reducir la duplicación de código: Cierto.
* Las variables definidas en una función son locales para esa función: Cierto.
* Es una mala idea definir nuevas funciones si hace que un programa sea más extenso: Falso. Las funciones pueden ser útiles para organizar el código, incluso si el resultado final es hacer que el programa sea un poco más largo. (p.153)

**Opción múltiple**

* La parte de un programa que usa una función se llama: (b) llamador
* Una definición de función de Python comienza con: (a) def
* Una función puede enviar la salida de vuelta al programa con un (n): un retorno
* Los parámetros formales y reales son emparejados por: (b) posición
* ¿Cuál de los siguientes no es un paso en el proceso de invocación de funciones? : (d) El control vuelve al punto justo antes de llamar a la función.
* En Python, los parámetros reales se pasan a las funciones: (a) por valor
* ¿Cuál de las siguientes no es una razón para usar funciones? (d) demostrar superioridad intelectual
* Si una función devuelve un valor, generalmente se debe llamar desde: (a) una expresión
* Una función sin declaración de devolución devuelve: (d) Ninguno
* Una función puede modificar el valor de un parámetro real solo si es: (a) mutable