

Taller 2 Python  
Samuel Botero Rivera  
cc: 1000654007

- a) 3 clases
- b) se utiliza para que el código de python corra solo cuando el archivo desde el cual se inicializa sea el `_main_`. para así evitar correr código no intencionado cuando otros módulos son importados,
- c) Si se elimina esta línea, y el archivo se importa como módulo en otro script, se ejecutará todo el código del archivo, lo que podría no ser deseado, ya que probablemente se ejecutarán operaciones como la creación de objetos o la ejecución de funciones cuando el archivo se importe.
- d) 2 objetos
- e) apostador1: Juan. apostador2: Ricardo
- f) `self` en `loteria = Loteria(value, self)` representa al apostador que está jugando en ese momento.
- g) 1
- h) 500  
Has perdido lo que apostaste  
100
- i) 400  
Has ganado 720.0  
720.0
- j) apostador
- k) probability, value
- l) se agrega como un método más dentro de la clase Loteria

```
n) def changeProbability(cls, nprobability):  
n)         cls.probability = nprobability
```

- m) `loteria1.changeProbability(0.99)`  
- (donde `loteria1` es un objeto de tipo `Loteria`)

- n) es correcto. el código sigue funcionando
- o) 5 métodos si se tiene en cuenta `__init__`
- p) No, si el apostador1 gana, el apostador2 no necesariamente gana también. Esto se debe a que el resultado del juego en la clase `Loteria` se determina de manera independiente para cada apostador a través del método `playGame`, que utiliza un número aleatorio entre 0 y 1 para determinar si el apostador gana o no. ambos apostadores tienen la misma probabilidad de ganar, pero se genera un número aleatorio cada vez que un apostador juega, no es el mismo para ambos apostadores.
- q) En python no es posible restringir variables a constantes. Pero sí es posible señalar que la intención de cierta variable es ser constante, esto declarando la variable en letras mayúsculas. Si se define `PROBABILITY` como constante, no tiene sentido

definir un método que cambia su valor. esto es una mala práctica de programación, más no generaría ningún error en la ejecución.

- r) devuelven float
- s) self es una referencia a la instancia de la clase desde la que se está llamando al método. en este caso hace referencia a una instancia de clase Loteria, que lleva a cabo el método playGame(). Si omities self, perdería la referencia a la instancia actual, lo cual haría que no pueda acceder a los atributos de la clase ni pasar la instancia a otros objetos
- t) self pasa por referencia, al ser un objeto que lleva a cabo el metodo play( un apostador). value pasa por valor, al ser una variable primitiva.