

## Desafío 2 - Objetos II

<https://github.com/DesafioLatam/TallerObjetosII>

### Instrucciones

- Crea una carpeta y guarda cada archivo `.rb` con el número de la pregunta, de manera local con **Sublime** o **Atom**.
- Guarda los cambios y súbelos a tu repositorio de GitHub.
- Luego de pusheados los últimos cambios, sube el link de GitHub a la plataforma Empieza.

### Ejercicio 1: Sintaxis

- Copia el siguiente código y ejecútalo. Luego corrige los errores para poder ejecutar ambos métodos y finalmente obtener el valor de la variable de *a*.

```
class Anything
  def foo
    a = 5
  end

  def bar
    @a += 1
  end
end

any = Anything.new
any.foo
any.bar
any.a
```

### Ejercicio 2: Sintaxis

Corrige el siguiente código para instanciar el objeto y generar la salida de manera correcta.

```
class Car(model, year)
  @model = model
  @year = year
end

car = Car.new('Camaro', 2016)
puts "Mi auto favorito es un #{car.model} del año #{car.year}!"
```

### Ejercicio 3: Sintaxis

Copia el siguiente código y ejecútalo. Luego corrige los errores para poder imprimir el nombre de la tienda.

```
class Store
  def initialize(name)
    name = @name
  end
end

store = store.new('Tienda 1')
puts store.name
```

### Ejercicio 4: Constructor con argumentos

Crea una clase llamada *Dog* cuyo constructor reciba como argumento un *hash* con la siguiente estructura:

```
propiedades = {nombre: 'Beethoven', raza: 'San Bernardo', color: 'Café'}
```

Instanciar un nuevo perro a partir de las propiedades contenidas en el *hash*. Luego generar un método llamado **ladrar** que, mediante interpolación, imprima *"Beethoven está ladrando!"*

### Ejercicio 5: Traductor entero a código Morse

Se tiene la clase *Morseable* que contiene un método de instancia *generate\_hash* los datos de traducción de número entero a código morse.

```
class Morseable
  def initialize(number)
    @number = number
  end

  def generate_hash(number)
    # Esto es una aberración y debe ser refactorizado!
    h = '-----' if number == 0
    h = '.-....' if number == 1
  end
end
```

```

h = '....-' if number == 2
h = '....-' if number == 3
h = '....-' if number == 4
h = '....-' if number == 5
h = '....-' if number == 6
h = '....-' if number == 7
h = '....-' if number == 8
h = '....-' if number == 9
end

def to_morse
  self.generate_hash(@number)
end
end

m = Morseable.new(3)
print m.to_morse

```

Se pide:

- Refactorizar el código del método de instancia *generate\_hash* para que los datos estén contenidos en un *hash* donde **los números serán las claves y la traducción los valores**. El método *generate\_hash* además debe retornar la traducción del número recibido como argumento.

## Ejercicio 6: Arrays y objetos

Dado el siguiente código:

```

class Student
  attr_accessor :name
  def initialize()
    @name = name
  end
end

nombres = %w(Alicia Javier Camila Francisco Pablo Josefina)

```

1. Iterar los nombres para crear un nuevo arreglo de objetos de Student.
2. Modificar el objeto para que pueda recibir una nota del alumno.
3. Agregar un getter para la nota.
4. Modificar la iteración para asignar notas incrementales de 1 en adelante.
5. Utilizar un map para obtener todas las notas de los alumnos a partir del arreglo de estudiantes.

## Ejercicio 7: Mascota Virtual

Vamos a generar una mascota virtual y interactuaremos con ella.

- Crea una clase llamada MyPet, La cual, al inicializarse genere una mascota con los siguientes parametros:

```
def initialize name
  @name = name
  @sleep = false
  @satisfied = 10 # Esta lleno
  @fullIntestine = 0 # No necesita ir
  puts @name + ' nace '
end
```

- Crea un método para hacer caminar a tu mascota e ir al baño, puedes seguir este ejemplo:

```
def walk
  puts 'Haces caminar a ' + @name + ' '
  @satisfied -= 2
  @fullIntestine = 2
  timeLapse
end
```

- Prueba el siguiente Método para hacer pasar el tiempo:

```
def timeLapse
  if @satisfied > 0
    # Mueve el alimento del estomago al intestino.
    @satisfied = @satisfied - 1
    @fullIntestine = @fullIntestine + 1
  else # Nuestro mascota esta hambrienta!
    if @sleep
      @sleep = false
      puts '¡Se despierta de repente!'
    end
    puts '¡' + @name + ' esta hambriento! En su desesperacion, ¡Murio de Hambre!'
    exit # Sale del programa.
  end

  if @fullIntestine >= 10
    @fullIntestine = 0
    puts '¡Uy! ' + @name + ' tuvo un accidente...'
  end

  if hungry?
    if @sleep
      @sleep = false
      puts '¡Se despierta de repente!'
    end
    puts 'El estomago de ' + @name + 'retumba...'
  end

  if needToGo?
    if @sleep
      @sleep = false
      puts 'Se despierta de repente!'
    end
  end
```

```
puts @name + ' hace la danza del baño...'
end
end
```

- Genera el método comer considerando que llenara su panza y pasara el tiempo.
- Genera el método dormir considerando que pasara el tiempo y vaciara su panza.
- Generar un menu para interactuar con tu mascota (alimentarlo, hacerlo dormir, salir a caminar) debes poder iterar las opciones, hasta que, Si la mascota muere de hambre debe terminar el programa.