Creando métodos

¡Comencemos!

¿Qué aprenderás?

- Definir métodos.
- Llamar métodos.
- Conocer el orden en que se ejecuta un código que posee métodos.

Introducción

En este capítulo aprenderemos a crear métodos desde cero. Esto nos permitirá simplificar códigos que ya hemos escrito y reutilizar código.

¡Vamos con todo!

Definiendo un método

En Ruby, un método se define utilizando la siguiente estructura:

```
def nombre_del_metodo # Definimos con def y un nombre.
 # Serie de instrucciones que ejecutará el método.
 # ...
 # ...
end # Terminamos de definir con end.
```

Todas las instrucciones definidas dentro del método serán ejecutadas cuando hagamos un llamado a este.

Creando nuestro primer método

```
def imprimir_menu
 puts 'Menú: '
 puts '1) Opción 1'
 puts '2) Opción 2'
 puts '3) Opción 3'
 puts '4) Salir'
end
```

Llamando al método

A los amigos, los llamamos por su nombre; a los métodos, de la misma forma. Si queremos llamar al método `imprimir_menu` simplemente escribiremos el nombre del método.

```
imprimir_menu
imprimir_menu()
```

Los paréntesis son optativos

Los paréntesis son optativos, sin embargo, la guía de estilo recomienda no utilizarlos.

Código más limpio es mejor código.

Cuidado con los paréntesis

Al utilizar paréntesis estos tienen que estar junto al nombre u obtendremos un error.

```
imprimir_menu () #ArgumentError: wrong number of arguments (given 1, expected 0)
```

Este error lo estudiaremos en el siguiente capítulo cuando estudiemos qué es un argumento.

El orden de ejecución

En el momento que ejecutamos un código Ruby, este se lee de arriba a abajo. Ruby necesita leer las definiciones para luego poder utilizarlas.

- Llamar a un método antes de definirlo genera un error.

```
saludar()
def saludar
 puts "hola"
end
```

Esto es equivalente a ocupar una variable antes de asignarle un valor.

Pero cuando nosotros inspeccionamos un código, es más sencillo comenzar leyendo desde el punto de partida.

```
def imprimir_menu
 puts 'Escoge una opción:'
 puts '-----------------'
 puts '1 - Acción 1'
 puts '2 - Acción 2'
 puts 'Escribe "salir" para terminar el programa'
 puts 'Ingrese una opción:'
end

# Partimos leyendo desde aquí.
opcion_menu = 'cualquier valor'
while opcion_menu != 'salir' || opcion_menu != 'Salir'
 imprimir_menu # Aquí leemos imprimir_menu
 opcion_menu = gets.chomp
```

```
 if opcion_menu == '1'
     puts 'Realizando acción 1'
 elsif opcion_menu == '2'
     puts 'Realizando acción 2'
 elsif opcion_menu != 'salir' || opcion_menu != 'Salir'
     puts 'Saliendo'
 else
     puts 'Opción inválida'
 end
end
```