**HOMEWORK**

**• ¿Cuál es la diferencia entre chomp y chop?**

**R//=**

Tanto chomp como chop son funciones de manipulación de cadenas de caracteres en el lenguaje de programación Perl.

* La función **chomp** elimina el último carácter de una cadena si ese carácter es un salto de línea (\n). Por lo tanto, chomp se utiliza comúnmente para eliminar los saltos de línea al final de una entrada de usuario o al leer un archivo de texto.

Por ejemplo:

my $string = "Hola\n";

chomp($string); # elimina el salto de línea

print $string; # imprime "Hola" sin salto de línea

* La función chop elimina el último carácter de una cadena, independientemente de cuál sea. Se usa comúnmente para eliminar el último carácter de una cadena de entrada en ciertas situaciones.

Por ejemplo:

my $string = "Hola";

chop($string); # elimina el último carácter "a"

print $string; # imprime "Hol" sin la "a" al final

En conclusión, chomp elimina solo el último carácter si es un salto de línea, mientras que chop elimina el último carácter de una cadena, independientemente de su tipo.

**• ¿Cuáles son los alcances de las diferentes variables y para qué sirven?**

**R//=** Los diferentes alcances de las variables (local, global, instancia, clase y estática) sirven para controlar la disponibilidad y visibilidad de las variables en diferentes partes del programa. Cada tipo de alcance de variable se utiliza para almacenar información de diferentes maneras y se adapta a diferentes situaciones en el desarrollo del programa:

* La variable local es una variable definida dentro de una función o bloque de código que solo es accesible dentro de ese bloque o función. Se utiliza comúnmente para almacenar información temporal que solo se necesita dentro de una función específica y se elimina de la memoria cuando la función o el bloque termina.
* La variable global es una variable definida fuera de cualquier función o bloque que es accesible desde cualquier parte del programa. Se inicializa al inicio del programa y mantiene su valor hasta que el programa termina. Se utiliza comúnmente para almacenar información que necesita ser accesible desde varias funciones o bloques.
* La variable de instancia es una variable que pertenece a una instancia particular de una clase en los lenguajes de programación orientados a objetos. Cada instancia de la clase tiene su propia copia de la variable de instancia, y se utiliza comúnmente para almacenar información que es específica de una instancia particular de una clase.
* La variable de clase es una variable que pertenece a toda la clase y es compartida por todas las instancias de la clase. Se utiliza comúnmente para almacenar información que es común a todas las instancias de una clase.
* La variable estática es una variable que mantiene su valor entre llamadas a una función o bloque de código. Se inicializa solo una vez, la primera vez que se llama a la función o al bloque de código, y mantiene su valor entre llamadas posteriores. Se utiliza comúnmente para mantener información persistente entre llamadas a funciones o bloques.

**• Una cadena de caracteres (string) es un objeto de la clase String. ¿Para qué se utilizan los métodos reverse, capitalize y length?**

**R//=** Los métodos reverse, capitalize y length son métodos de la clase String en la programación orientada a objetos.

* El método reverse() invierte el orden de los caracteres de una cadena. Por ejemplo, si tenemos la cadena "hola", al aplicar el método reverse() obtendremos "**aloh**".
* El método capitalize() convierte el primer carácter de una cadena en mayúscula y el resto en minúsculas. Por ejemplo, si tenemos la cadena "hola mundo", al aplicar el método capitalize() obtendremos "Hola mundo".
* El método length() devuelve el número de caracteres de una cadena. Por ejemplo, si tenemos la cadena "hola", al aplicar el método length() obtendremos **4**, que es la cantidad de caracteres de la cadena.

**Escribir diferentes programas que, basados en la entrada del usuario, calculen el área de:**

• Un círculo:

**#** Pedir al usuario que ingrese el radio del círculo:

puts "Ingrese el radio del círculo:"

radio = gets.to\_f

**#** Calcular el área de un círculo

def area\_circulo(radio)

return Math::PI \* radio \*\* 2

end

**#** Mostrar el resultado al usuario

puts "El área del círculo es: #{area\_circulo(radio)}"

• Un triángulo

**#** Pedir al usuario que ingrese la base y la altura del triángulo

print "Ingrese la base del triángulo: "

base = gets.to\_f

print "Ingrese la altura del triángulo: "

altura = gets.to\_f

**#** Calcular el área de un triángulo

def area\_triangulo(base, altura)

return (base \* altura) / 2.0

end

**#** Mostrar el resultado al usuario

puts "El área del triángulo es: #{area\_triangulo(base, altura)}"

• Un cuadrado

**#** Pedir al usuario que ingrese el lado del cuadrado

print "Ingrese el lado del cuadrado: "

lado = gets.to\_f

**#**Calcular el área de un cuadrado

def area\_cuadrado(lado)

return lado \*\* 2

end

**#**Mostrar el resultado al usuario

puts "El área del cuadrado es: #{area\_cuadrado(lado)}"

• Un rectángulo

**#** Pedir al usuario que ingrese la base y la altura del rectángulo

print "Ingrese la base del rectángulo: "

base = gets.to\_f

print "Ingrese la altura del rectángulo: "

altura = gets.to\_f

**#** Calcular el área del rectángulo

def area\_rectangulo(base, altura)

return base \* altura

end

**#** Mostrar el resultado al usuario

puts "El área del rectángulo es: #{area\_rectangulo(base, altura)}"

• Un trapecio

**#** Pedir al usuario que ingrese la base mayor, la base menor y la altura del trapecio

puts "Ingrese la base mayor del trapecio: "

base\_mayor = gets.to\_f

puts "Ingrese la base menor del trapecio: "

base\_menor = gets.to\_f

puts "Ingrese la altura del trapecio: "

altura = gets.to\_f

**#** Calcular el área del trapecio

def area\_trapecio(base1, base2, altura)

return ((base1 + base2) \* altura) / 2.0

end

**#** Mostrar el resultado al usuario

puts "Las medidas ingresadas son: Base mayor = #{base\_mayor}, Base menor = #{base\_menor} y Altura = #{altura}"

puts "El área del trapecio es: #{area\_trapecio(base\_mayor, base\_menor, altura)}"