

## Ejercicios de scripts

Al final tienes unos ejemplos de cómo llamar a estas funciones y procedimientos, con los resultados que debes obtener, para que te autoevalúes.

1. Crea una función que sume los 3 números (float) pasados como parámetros.
2. Crea una función que devuelva true (cierto) si el número pasado como parámetro es par, y falso en otro caso.
3. Construye una función que devuelva el máximo de los 3 números que recibe.
4. Construye una función que devuelva una cadena con tantos guiones como se indique en el parámetro.
5. Crea una función que sume los n primeros números naturales. N será indicado como parámetro.
6. Crea una función que sume los 1/n números, donde n es el parámetro que recibe la función.
7. Crea una función que calcule la hipotenusa de un triángulo rectángulo a partir de la longitud de los catetos.
8. Construye una función que dado el peso (en kg como float) y la altura (en centímetros), calcule el Índice de Masa Corporal (IMC).
9. Crea una función que use la función anterior, y devuelva un literal indicando si el IMC es "Bajo Peso", "Rango Normal", "Sobrepeso" y "Obesidad". Pruébala con tus valores.
10. Construir una función que dé como resultado el precio con IVA de un artículo. Para ello recibe como parámetros el precio del artículo sin IVA y el IVA a aplicar.
11. Construir una función que dé como resultado el importe del IVA de un artículo. Para ello recibe como parámetros el precio del artículo con IVA y el IVA aplicado.
12. Crea una función que devuelva el factorial del número n pasado por parámetro.
13. (opcional) Crea la función factorial del ejercicio anterior de forma recursiva. MySQL no permite funciones recursivas, así que deberás crearlo como un procedimiento. Para poder hacer que el procedimiento sea recursivo, debes ejecutar primero:  
`set max_sp_recursion_depth = 50`
14. (opcional) Crea una función que devuelva el número de Fibonacci de un número pasado por parámetro. Al igual que antes, hazlo con un procedimiento.
15. Crea un procedimiento que reciba como parámetros de entrada el dividendo y el divisor y devuelva como parámetros de salida el cociente y el resto. Usa este procedimiento en un programa en el que se divida 23 entre 4, y que muestre por pantalla el dividendo, divisor, cociente y resto.
16. Crea un procedimiento que reciba el código de un cliente de la base de datos jardinería y liste los datos de ese cliente, así como los pagos que ha realizado, ordenados cronológicamente. Para finalizar, que muestre la cantidad total pagada. En la imagen siguiente se muestra el detalle de lo que debe mostrar el procedimiento.

```
CODIGO CLIENTE: 1
NOMBRE CLIENTE: DGPRODUCTIONS GARDEN
CIUDAD CLIENTE: San Francisco
PAIS CLIENTE: USA
=====
ID-TRANSACCION      FECHA      FORMA      CANTIDAD
=====
ak-std-000001      2008-11-10      PayPal      2000.00
ak-std-000002      2008-12-10      PayPal      2000.00
=====
TOTAL PAGOS EFECTUADOS : 4000
```

```
CODIGO CLIENTE: 3
NOMBRE CLIENTE: Gardening Associates
CIUDAD CLIENTE: Miami
PAIS CLIENTE: USA
=====
ID-TRANSACCION      FECHA      FORMA      CANTIDAD
=====
ak-std-000003      2009-01-16      PayPal      5000.00
ak-std-000004      2009-02-16      PayPal      5000.00
ak-std-000005      2009-02-19      PayPal      926.00
=====
TOTAL PAGOS EFECTUADOS : 10926
```

## Ejemplos de uso:

Ejercicios 1 a 4:

```

mysql> use clase1;
Database changed
mysql> select suma3(1,2,4);
+-----+
| suma3(1,2,4) |
+-----+
|          7 |
+-----+
1 row in set (0.00 sec)

mysql> select par(8), par(9), par2(8), par2(9);
+-----+-----+-----+-----+
| par(8) | par(9) | par2(8) | par2(9) |
+-----+-----+-----+-----+
|      1 |      0 |      1 |      0 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select maximo(3,2,1), maximo (2,3,1), maximo(2,1,3), maximo(1,2,3);
+-----+-----+-----+-----+
| maximo(3,2,1) | maximo (2,3,1) | maximo(2,1,3) | maximo(1,2,3) |
+-----+-----+-----+-----+
|          3 |          3 |          3 |          3 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select guiones(7);
+-----+
| guiones(7) |
+-----+
| ----- |
+-----+
1 row in set (0.00 sec)

```

Ejercicios 5 a 9:

```
mysql> select sumaNaturales(3), sumaNaturales(8);
+-----+-----+
| sumaNaturales(3) | sumaNaturales(8) |
+-----+-----+
| 6 | 36 |
+-----+-----+
1 row in set (0.00 sec)

mysql> select sumaFracciones(3), sumaFracciones(8);
+-----+-----+
| sumaFracciones(3) | sumaFracciones(8) |
+-----+-----+
| 1.83333333730697632 | 2.7178573608398438 |
+-----+-----+
1 row in set (0.00 sec)

mysql> select hipotenusa(3,4), hipotenusa(5.6, 8.9);
+-----+-----+
| hipotenusa(3,4) | hipotenusa(5.6, 8.9) |
+-----+-----+
| 5 | 10.515226364135742 |
+-----+-----+
1 row in set (0.00 sec)

mysql> select imc(70, 180), imc(50,160), imc(70,160);
+-----+-----+-----+
| imc(70, 180) | imc(50,160) | imc(70,160) |
+-----+-----+-----+
| 21.604938507080078 | 19.53125 | 27.34375 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select varcharIMC(70, 180), varcharIMC(50,160), varcharIMC(70,160);
+-----+-----+-----+
| varcharIMC(70, 180) | varcharIMC(50,160) | varcharIMC(70,160) |
+-----+-----+-----+
| Rango normal | Peso Bajo | Sobrepeso |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Ejercicios 10 a 13:

```
mysql> select precioConIVA(100,21), precioConIVA(2000,21), precioConIVA(348,21);
+-----+-----+-----+
| precioConIVA(100,21) | precioConIVA(2000,21) | precioConIVA(348,21) |
+-----+-----+-----+
| 121 | 2420 | 421.0799865722656 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select importeIVA(121,21), importeIVA(2420,21), importeIVA(421.08,21);
+-----+-----+-----+
| importeIVA(121,21) | importeIVA(2420,21) | importeIVA(421.08,21) |
+-----+-----+-----+
| 21 | 420 | 73.07999420166016 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select factorial(2), factorial(5), factorial(10);
+-----+-----+-----+
| factorial(2) | factorial(5) | factorial(10) |
+-----+-----+-----+
| 2 | 120 | 3628800 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> set max_sp_recursion_depth = 50;
Query OK, 0 rows affected (0.00 sec)

mysql> call factorialRecursivo(2, @numero);
Query OK, 0 rows affected (0.00 sec)

mysql> call factorialRecursivo(5, @numero1);
Query OK, 0 rows affected (0.00 sec)

mysql> call factorialRecursivo(10, @numero2);
Query OK, 0 rows affected (0.00 sec)

mysql> select @numero, @numero1, @numero2;
+-----+-----+-----+
| @numero | @numero1 | @numero2 |
+-----+-----+-----+
| 2 | 120 | 3628800 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Ejercicio 14 y 15:

```
mysql> call fibonacci(2,@numero);
Query OK, 0 rows affected (0.00 sec)

mysql> call fibonacci(5,@numero1);
Query OK, 0 rows affected (0.00 sec)

mysql> call fibonacci(10,@numero2);
Query OK, 0 rows affected (0.00 sec)

mysql> select @numero, @numero1, @numero2;
+-----+-----+-----+
| @numero | @numero1 | @numero2 |
+-----+-----+-----+
| 1       | 5        | 55       |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> set @dividendo=23;
Query OK, 0 rows affected (0.00 sec)

mysql> set @divisor=4;
Query OK, 0 rows affected (0.00 sec)

mysql> call division(@dividendo,@divisor,@cociente,@resto);
Query OK, 0 rows affected (0.00 sec)

mysql> select @dividendo, @divisor, @cociente, @resto;
+-----+-----+-----+-----+
| @dividendo | @divisor | @cociente | @resto |
+-----+-----+-----+-----+
|          23 |         4 |          5 |        3 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```