



Trabajo Práctico N° 4: Funciones

MUY IMPORTANTE PARA TODOS LOS EJERCICIOS DEL PRÁCTICO

1. Diseñar algoritmos (empleando PSEUDOCODIGO) para resolver el problema planteado
2. Realizar una traza en papel para determinar si el algoritmo diseñado es correcto. (Puede exceptuar a los ejercicios 1 y 5.
3. Transformar el algoritmo en Pseudocódigo a PHP. Usar comentarios para documentar cada función y especificar el tipo de parámetros, valor retornado. Verificar el código en ejecución, realizando las pruebas que sean necesarias hasta que el mismo sea correcto.

1. DIBUJANDO: Este ejercicio implementarlo sólo en php para poder ejecutarlo

Objetivo del ejercicio: Definir módulos **sin parámetros y sin retorno** que sean invocados desde el programa principal sólo utilizando el nombre del módulo.

- a) Diseñar un módulo que imprima el cartel (caños) : " +----+ | +----+"
- b) Diseñar un módulo que imprima el cartel (Campo) : " | | | | |"
- c) Diseñar un módulo que imprima el cartel (Linea Media) : " | | | | |"
- d) Diseñar un módulo que imprima el cartel (Linea de banda): " +-----+ +-----+ +-----+"
- e) Diseñar un algoritmo principal que llame a los módulos anteriores las veces necesarias para imprimir el plano de una cancha de fútbol-sala de la siguiente manera:

```
+-----+ +-----+ +-----+
|         |         |         |
|         |         |         |
|         |         |         |
|         |         |         |
|         |         |         |
|         |         |         |
|         |         |         |
+-----+ +-----+ +-----+
```

2. Diseñe un módulo esMultiploDe2 que reciba como parámetro de entrada (parámetro formal) un número, devuelva (retorne) **true** si es múltiplo de 2 y **false** si no lo es. (*Observación: un número B es múltiplo de A si $B = A.N$, dicho de otra manera, B es divisible por A*). (múltiplos de 2 = {2, 4, 6, 8,}).

Objetivo del ejercicio: Definir un módulo con un (1) parámetro de entrada, cuyo cuerpo incluya la expresión booleana que determine si el parámetro de entrada es múltiplo de 2.

Realice la traza del módulo para **esMultiploDe2(19)** y para **esMultiploDe2(40)**.

3. Una parte importante de la modularización, es aprender a utilizar módulos implementados por otros programadores. Vamos a utilizar funciones predefinidas en php: sin saber cuáles son las instrucciones del cuerpo y sólo conociendo su interfaz/cabecera podemos utilizarlas para resolver una expresión.

En la librería de funciones de php ya cuenta con funciones, cuya cabecera y documentación es la siguiente:

```
/**
 * Calcula la base elevada al exponente
 * @param number $base
 * @param number $exponente
 * @return number retorna el cálculo de la potencia
 */
pow($base, $exponente)

/**
 * Calcula la raiz cuadrada de un número
 * @param number $numero
 * @return number retorna el calculo de la raiz cuadrada
 */
sqrt($numero)

/**
 * Calcula el valor absoluto de un número
 * @param number $numero
 * @return number retorna el valor absoluto
 */
abs($numero)
```



a) implemente un programa principal que solicite dos números enteros M y N por teclado, y muestre en pantalla el resultado del siguiente cálculo: $M^N + \sqrt{|M|}$

(verifique que con los valores $M = -4$ y $N = 0$, el resultado es 3)
(verifique que con los valores $M = -16$ y $N = 2$, el resultado es 260)

tip: $M^N + \sqrt{|M|}$ significa “M elevado a la N más la raíz cuadrada del valor absoluto de M”.

4. CALCULANDO FÓRMULAS PARA UN TRIÁNGULO

a) Diseñar módulos para cada punto:

- i) Recibiendo como parámetro el lado de un triángulo equilátero, devolver su perímetro
- ii) Recibiendo como parámetro el lado de un triángulo equilátero, devolver su área

Tip: Recuerde que si conoce cuánto mide un lado del triángulo equilátero, entonces $\text{altura} = \sqrt{(\text{lado}^2 - (\text{lado}/2)^2)}$

b) Diseñar un algoritmo principal que lea cuánto mide el lado de un triángulo equilátero en cm (tipo entero) y luego llame a ejecutar cada uno de los módulos anteriores y muestre los resultados por pantalla con un cartel con el siguiente formato:

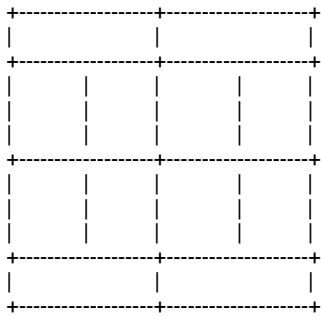
“Dado un triángulo equilátero de lado 4 cm, su perímetro es 12 cm y su área es 6,92820323027551 cm²”
Realice la traza para el triángulo de lado 4 cm.

Observación: una variante del ejercicio es definir un módulo, que a partir de la medida del lado del triángulo equilátero, calcule (y retorne) la medida de la altura del triángulo equilátero.

5. DIBUJANDO: Este ejercicio implementarlo sólo en php para poder ejecutarlo

Objetivo del ejercicio: Reusar módulos del ejercicio 1.

a) Reutilice los módulos del ejercicio 1 (y agregue los que sean necesarios), para dibujar una cancha de tenis como la siguiente:



6. Diseñe módulos para calcular superficie circular y superficie de corona circular. Modularizar de forma conveniente para reusar funcionalidad. (Considere que la superficie de la corona circular se obtiene como la resta de las superficies de los círculos que comparten su centro). Escriba un programa principal que solicite los datos para mostrar en pantalla el área de una corona circular. Dé dos ejemplos de entradas y haga la traza del algoritmo principal (con la correspondiente invocación a los módulos definidos).



7. Para una competencia de ciclismo se requiere calcular la velocidad en m/seg del 1° y 2° puesto. El usuario del programa proveerá información de: distancia en metros de la carrera y el tiempo en horas, minutos y segundos de cada uno de los puestos.

8. Un laboratorio de fármacos elabora un antialérgico basado en Loratadina y en Betametasona. Al realizar la composición se utiliza agua destilada según esta proporción: 10% de la cantidad de Loratadina más 15% de la cantidad de Betametasona. Diseñe en Pseudocódigo, traduzcalo apropiadamente al lenguaje PHP. Asegúrese que los parámetros actuales no coincidan en nombre con los parámetros formales:



- a. Una función **calcAguaDestilada** que recibe por parámetros la cantidad de Loratadina y la cantidad de Betametasona. La función retorna la cantidad de agua. (recuerde que la cantidad de agua destilada es igual a la suma del 10% de la cantidad de Loratadina más el 15% de la cantidad de Betametasona)
- b. Un programa principal que solicita la cantidad de Betametasona y la cantidad de Loratadina y, utilizando la función del inciso a), muestra la cantidad de agua necesaria.

Realizar la traza para los siguientes valores:

- i) 130 de Loratadina y 100 de Betametasona
- ii) 50 de Loratadina y 80 de Betametasona

9. En el quinto grado de una escuela primaria están aprendiendo figuras geométricas. La maestra necesita un programa que calcule distintas medidas de un cilindro. El requerimiento es que cada alumno, a partir de la altura y el radio obtenga un detalle de la figura.

Por ejemplo: para una altura de 5cm y una base de radio 3.5 cm el alumno debe obtener el siguiente detalle:

Cilindro con h=5 y r=3.5:

longitud de la circunferencia de la base (cm) = 21.98
superficie base (cm²) = 38.465
superficie lateral del cilindro (cm²) = 109.9
superficie total cilindro (cm²) = 186.83
volumen del cilindro (cm³) = 192.325

Debe modularizar convenientemente para obtener las distintas medidas del cilindro.
Documente los parámetros y retorno de todas las funciones.

Utilice la constante **M_PI**¹ o la función **pi()** de PHP para obtener el valor 3.14

Al realizar la traducción a php, utilice la función **round**² de PHP para mostrar sólo 2 decimales

10. Una empresa quiere transmitir datos mediante la línea telefónica, pero les preocupa que sus teléfonos pudieran estar intervenidos. Los números que se necesitan transmitir son enteros de **cuatro dígitos**. El número se encripta de la siguiente manera: reemplace cada dígito con el residuo de la división entre "la suma del dígito más 7" y 10. Posteriormente, intercambie el primer dígito con el tercero, e intercambie el segundo dígito con el cuarto. También se necesita realizar el proceso inverso: dado un entero encriptado, desencriptarlo para formar el número original.

Especifique una función para la encriptación cuya entrada es un número y la salida es un número encriptado. Luego **especifique una función** para la desencriptación, cuya entrada es un número encriptado y la salida es el número desencriptado.

Observación: Puede que los números obtenidos sean de menos de cuatro dígitos. Por ejemplo el número 3333 queda encriptado como un 0, y si ingresa 0 en la desencriptación volverá a generar el número 3333.

Especifique un programa principal que solicite un número de 4 dígitos al usuario y le muestre el número encriptado, internamente el programa principal toma el número encriptado y al finalizar muestra el número desencriptado.

Realice la traza para el número 1058.

11. Diseñe un módulo llamado **esMultiplo** que recibirá dos números como parámetro y devolverá **true** si el primer parámetro es múltiplo del segundo parámetro, **false** en caso contrario.

¹ <http://php.net/manual/es/function.pi.php>

² <http://php.net/manual/es/function.round.php>