

Documentación: b3.php

Autor: Héctor Arciniega Valencia

Nivel B – Ejercicio 3

1 <?php

- Rol: apertura del intérprete PHP; el servidor evalúa el bloque.

2 /**

- Rol: inicio de DocBlock; documentación para personas y herramientas.

3 * Nivel B – Ejercicio 3 (Función)

- Rol: comentario; sin efecto en tiempo de ejecución.

4 * ¿Número capicúa?

- Rol: comentario; sin efecto en tiempo de ejecución.

5 *

- Rol: comentario; sin efecto en tiempo de ejecución.

6 * Entrada: num (entero ≥ 0)

- Rol: comentario; sin efecto en tiempo de ejecución.

7 * Función: esCapicua(int \$n): bool

- Rol: comentario; sin efecto en tiempo de ejecución.

8 * Salida: "SI" si es capicúa, "NO" en caso contrario.

- Rol: comentario; sin efecto en tiempo de ejecución.

9 */

- Rol: fin de DocBlock.

10

11 declare(strict_types=1);

- Rol: directiva strict_types.
- Efecto: activa tipado estricto en este archivo; prohíbe coerciones implícitas en firmas.

12

13 function validar_num(?string \$raw): array {

- Rol: declaración de función.
- Nombre: validar_num
- Parámetro \$raw: tipo declarado ?string.
- Retorno: array.
- Contrato: retorna array: {ok:bool, val?:T, msg?:string}.

14 if (\$raw === null || \$raw === '') return ['ok'=>false,'msg'=>'Dato vacío.'];

- Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.
- Comparación con cadena vacía para validar presencia de datos.

15 if (!preg_match('/^\d+\$/ ', trim(\$raw))) return ['ok'=>false,'msg'=>'Debe ser entero ≥ 0 .'];

- Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.
- Función: preg_match(patcón, sujeto, [coincidencias], [flags], [offset]). Devuelve 1/0 o false.
- Patcón: $/\^\d+\$/ \rightarrow$ ^ ancla inicio; \d dígito [0-9]; + uno o más; \$ ancla fin. Permite '0', '10', '123'. Rechaza '-1', '1.2', '3', '3 '.
- Función: trim(cadena). Elimina espacios/blancos en extremos.

16 // El patcón anterior ya garantiza no-negativo

- Rol: comentario; sin efecto en tiempo de ejecución.

17 return ['ok'=>true,'val'=>(int)\$raw];

- Rol: retorno; finaliza la función devolviendo el valor indicado.

18 }

19

20 /**

- Rol: inicio de DocBlock; documentación para personas y herramientas.

21 * Determina si un entero no negativo es capicúa (palíndromo).

- Rol: comentario; sin efecto en tiempo de ejecución.

22 * Implementación usando string: compara con su reverso.

- Rol: comentario; sin efecto en tiempo de ejecución.

23 * @param int \$n

- Rol: comentario; sin efecto en tiempo de ejecución.

24 * @return bool

- Rol: comentario; sin efecto en tiempo de ejecución.

25 */

- Rol: fin de DocBlock.

26 function esCapicua(int \$n): bool {

- Rol: declaración de función.
- Nombre: esCapicua
- Parámetro \$n: tipo declarado int.
- Retorno: bool.

27 \$s = (string)\$n;

28 \$rev = strrev(\$s);

29 return \$s === \$rev;

- Rol: retorno; finaliza la función devolviendo el valor indicado.

30 }

31

32 \$num = null;

- Rol: inicialización de variable a null.

33 \$resp = null;

- Rol: inicialización de variable a null.

34 \$error = null;

- Rol: inicialización de variable a null.

35

36 if (\$_SERVER['REQUEST_METHOD'] === 'POST') {

- Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.
- Condición: \$_SERVER['REQUEST_METHOD'] === 'POST' (método HTTP).

37 \$res = validar_num(\$_POST['num'] ?? null);

38 if (!\$res['ok']) {

- Rol: estructura condicional. Ejecuta el bloque si la condición es verdadera.

39 \$error = \$res['msg'];

40 } else {

41 \$num = (int)\$res['val'];

42 \$resp = esCapicua(\$num) ? 'SI' : 'NO';

43 }

44 }

45 ?>

- Rol: cierre del bloque PHP; desde aquí se emite HTML sin evaluar.

46 <!doctype html>

- Rol: declaración HTML5 (modo estándar).

47 <html lang="es">

- Atributos: lang="es": idioma del documento.

48 <head>

49 <meta charset="utf-8" />

- Atributos: charset="utf-8": codificación de caracteres.

50 <title>NB-B3 Número capicúa</title>

51 <style>

52 body{font-family:system-ui, -apple-system, Segoe UI, Roboto, Arial, sans-serif; margin:24px}

- CSS: font-family: pila de fuentes del sistema para legibilidad y rendimiento. margin: 24px; margen externo del elemento.

53 fieldset{border:1px solid #ccc;padding:12px;border-radius:8px;max-width:520px}

- CSS: border: 1px solid #ccc; estilo y color del borde. padding: 12px; relleno interno del elemento. border-radius: 8px; radios de las esquinas. max-width: 520px; ancho máximo para diseño fluido.

54 label{display:block;margin:8px 0 4px}

- CSS: display:block: ocupa toda la línea disponible. margin: 8px 0 4px; margen externo del elemento.

55 input[type=number]{width:100%;padding:8px;border:1px solid #bbb;border-radius:6px}

- CSS: width: 100%; ancho del control. padding: 8px; relleno interno del elemento. border: 1px solid #bbb; estilo y color del borde. border-radius: 6px; radios de las esquinas.

56 button{padding:8px

14px;border:0;border-radius:8px;background:#07a;color:#fff;cursor:pointer;margin-top:10px}

- CSS: padding: 8px 14px; relleno interno del elemento. border: 0; estilo y color del borde. border-radius: 8px; radios de las esquinas. background: #07a; color de fondo. color: #fff; color del texto. cursor:pointer: apariencia de mano al pasar el mouse.

57 .error{color:#b00020;font-weight:700}

- CSS: color: #b00020; color del texto.

58 .result{margin-top:12px;padding:8px;border-left:4px solid #07a;background:#eef7ff}

- CSS: padding: 8px; relleno interno del elemento. background: #eef7ff; color de fondo.

59 code{background:#f7f7f7;padding:2px 4px;border-radius:4px}

- CSS: background: #f7f7f7; color de fondo. padding: 2px 4px; relleno interno del elemento. border-radius: 4px; radios de las esquinas.

60 </style>

61 </head>

62 <body>

63 <h1>NB-B3 – ¿Número capicúa?</h1>

64 <form method="post" novalidate>

- Atributos: novalidate: desactiva validación HTML5; validación queda del lado servidor.

65 <fieldset>

66 <legend>Entrada</legend>

67 <label for="num">Número entero ≥ 0</label>

- Atributos: for="num": asocia la etiqueta con el control id="num".

68 <input type="number" id="num" name="num" min="0" step="1" value="<?php echo htmlspecialchars(\$_POST['num'] ?? ''); ?>" />

- Función: htmlspecialchars(texto[, ENT_QUOTES, 'UTF-8']). Escapa &, <, > y comillas.
- Seguridad: usar ENT_QUOTES y 'UTF-8' evita XSS por comillas simples y dobles.
- Atributos: type="number": tipo de control. id="num": identificador único en el DOM. name="num": clave enviada en la petición (GET). min="0": valor mínimo permitido por el navegador. step="1": incremento permitido (enteros). value="<?php...>": relleno dinámico; se escapa con htmlspecialchars para evitar XSS.

69 <button type="submit">Evaluar</button>

- Atributos: type="submit": tipo de control.

70 <?php if (\$error): ?><div class="error"><?php echo htmlspecialchars(\$error); ?></div><?php endif; ?>

- Rol: apertura del intérprete PHP; el servidor evalúa el bloque.
- Función: htmlspecialchars(texto[, ENT_QUOTES, 'UTF-8']). Escapa &, <, > y comillas.
- Seguridad: usar ENT_QUOTES y 'UTF-8' evita XSS por comillas simples y dobles.

71 </fieldset>

72 </form>

73

74 <?php if (\$resp !== null): ?>

- Rol: apertura del intérprete PHP; el servidor evalúa el bloque.

75 <div class="result">

76 Salida:

77 <div>¿Capicúa? <code><?php echo \$resp; ?></code></div>

78 </div>

79 <?php endif; ?>

- Rol: apertura del intérprete PHP; el servidor evalúa el bloque.

80 </body>

81 </html>