



Controla el flujo del programa

Desarrollador PHP (orientado a objetos)

Nivel 1 / Lección 2 / Actividad 1

IMPORTANTE

Para resolver tu actividad, **guárdala** en tu computadora e **imprímela**.

Si lo deseas, puedes conservarla para consultas posteriores, ya que te sirve para reforzar tu aprendizaje. No es necesario que la envíes para su revisión.

Propósito de la actividad

Implementar funciones y estructuras de control para manejar el flujo de ejecución de un programa.

Practica lo que aprendiste

- I. Menciona cuatro operaciones que puedes realizar con arreglos en PHP.
 1. Agregar elementos a un arreglo con `array_push()` o con el operador `[]`.
 2. Eliminar elementos de un arreglo con `unset()` o `array_pop()`.
 3. Ordenar un arreglo con `sort()` o `asort()`.
 4. Buscar elementos dentro de un arreglo con `in_array()` o `array_search()`.



II. Lee los enunciados para relacionar las columnas.

- | | |
|---|--|
| a. Tipo de dato que almacena símbolos y cadenas de caracteres. | (e) Sentencias de salto |
| b. Normas que permiten controlar el flujo de un programa a partir del resultado de una condición. | (d) Funciones |
| c. Ejecutan instrucciones de manera repetitiva y reducen la cantidad de líneas de código. | (a) String |
| d. Sirven para reutilizar sentencias y mantener en orden el código. | (d) Estructuras de control condicionales |
| e. Cambian el flujo de ejecución de un ciclo. | (c) Estructuras de control iterativas |

III. Escribe los pasos que debes seguir para declarar una función.

Escribe la palabra clave function.

Escribe el nombre de la función.

Abre y cierra paréntesis () para los parámetros (pueden ir vacíos si no tiene).

Abre y cierra llaves {} para delimitar el cuerpo de la función.

Dentro de las llaves, escribe el código que deseas que la función ejecute.

(Opcional) Usa return si necesitas devolver un valor.



IV. Completa el siguiente código y contesta las preguntas.

```
<?php
echo "Funciones y Estructuras de control.";

/* Declaración de la función 'puede_trabajar' */
_____ puede_trabajar($edades){
    $trabajadores = [];

    for($i = 0; $i < 5; $i++){
        if($edades[$_] < 18 ){
            echo "El usuario de {$edades[$i]} años es menor de edad y aún no puede
trabajar.";
            continue;
        }____{
            echo "El usuario de {_____} años es mayor de edad y puede trabajar";
            $trabajadores__ = "Trabajador de {$edades[$i]} años.";
        }
    }

    return $trabajadores;
}

/* Llamada a la función 'puede_trabajar()' */
_____edades = [12, 18, 19, 20, 25, 35];
$trabajadores = puede_trabajar($edades);
?>
```

a. ¿Cuál es el resultado después de ejecutar el código completo?

Se imprimen mensajes indicando si cada usuario puede o no trabajar,
dependiendo de su edad.

Además, se guarda en \$trabajadores una lista con los usuarios mayores de edad
(mayores o iguales a 18 años), excepto el último número porque el ciclo solo recorre
hasta 5 elementos.

b. Menciona las estructuras de control que se utilizan en el código.

for (estructura iterativa)

if y else (estructuras condicionales)

continue (sentencia de salto)

c. Describe con tus propias palabras la función que realiza la
sentencia **continue**.

La sentencia continue hace que se salte el resto del código dentro del
ciclo en esa vuelta, y pase directamente a la siguiente iteración.

Se usa cuando no se quiere que se ejecute más código para ciertos
casos dentro del ciclo.