

unidad didáctica 02

Actividades



1. **Php básico**
2. **bucles**
3. **arrays**
4. **funciones**
5. **funciones predefinidas**
6. **otros ejercicios**
7. **ejercicios del concurso ACEPTA EL RETO**

1. Php básico

Actividad 201

Visualiza el [vídeo](#) de *Jesús Amieiro* sobre PHP en 2020 a partir del minuto 3:32 (son 40 minutos aproximadamente).

- ¿Qué relación existe entre PHP y Facebook?
- Respecto al rendimiento, ¿qué versión mínima deberíamos utilizar?
- ¿Por qué PHP tiene mala fama?

Actividad 202

`202tresfrases.php`: Muestra 3 frases, cada una en un párrafo utilizando las tres posibilidades que existen de mostrar contenido. Tras ello, introduce dos comentarios, uno de bloque y otro de una línea.

Actividad 203

`203calculos.php`: Escribe un programa que utilice las variables `$x` y `$y`. Asígnales los valores `166` y `999` respectivamente. A continuación, muestra por pantalla el valor de cada variable, la suma, la resta, la división y la multiplicación.

Actividad 204

`204datosPersonales.php`: Escribe un programa que almacene en variables tu nombre, primer apellido, segundo apellido, email, año de nacimiento y teléfono. Luego muéstralos por pantalla dentro de una tabla.

Nombre	Bruce
Apellidos	Wayne Enterprise
Email	batman@dccomics.com
Año de nacimiento	1939
Teléfono	666123456

Actividad 205

`205datosPersonales.html` y `205datosPersonales.php`: Es el mismo ejercicio que el anterior, pero separando la lógica. En el primer archivo crearemos el formulario para introducir los datos, y luego recogemos los datos y generamos la tabla en el segundo archivo.

Actividad 206

`206madlib.html` y `206madlib.php`: A partir de un nombre, un verbo, un adjetivo y un adverbio, crea una historia que contenga dichos elementos. Por ejemplo:

- Entrada: perro / caminar / azul / rápidamente
- Salida: ¿ Te gusta caminar con tu perro azul rápidamente ?
- `206madlib2.html` y `206madlib2.php` Crea un madlib más extenso, leyendo más datos de entrada.

Actividad 207

`207anyos.php`: Tras leer la edad de una persona, mostrar la edad que tendrá dentro de 10 años y hace 10 años. Además, muestra qué año será en cada uno de los casos. Finalmente, muestra el año de jubilación suponiendo que trabajarás hasta los 67 años. En este caso, no hace falta que previamente crees un formulario, puedes probar el ejercicio via URL: `207anyos.php?edad=33`.

Nota:

```
$anyoActual = date("Y");
```

Actividad 208

`208dinero.php`: A partir de una cantidad de dinero, mostrar su descomposición en billetes (500, 200, 100, 50, 20, 10, 5) y monedas (2, 1), para que el número de elementos sea mínimo. No se utilizar ninguna instrucción condicional. Por ejemplo, al introducir `139` debe mostrar:

```
1 1 billete de 100
2 0 billete de 50
3 1 billete de 20
4 1 billete de 10
5 1 billete de 5
6 2 moneda de 2
```

Nota: Puedes forzar a realizar la división entera mediante la función `intval($dividendo, $divisor)` o pasar un número flotante a entero puedes usar la función `intval()`.

Actividad 209

`209posnegcero.php`: A partir de un numero, muestra por pantalla si el número es `positivo`, `negativo` o `cero`.

Actividad 210

`210mayor3.php`: Sin hacer uso de condiciones que utilicen dentro la condición los operadores lógicos, muestra el mayor de tres números (a, b y c).

`210mayor3c.php`: Utiliza en las condiciones los operadores lógicos.

Actividad 211

`211nombreEdad.php`: A partir de una edad muestra por pantalla:

- *bebé* si tiene menos de 3 años
- *niño* si tiene entre 3 y 12 años
- *adolescente* entre 13 y 17 años
- *adulto* entre 18 y 66
- *jubilado* a partir de 67

Actividad 212

`212reloj.php`: Escribe un programa que funcione similar a un reloj, de manera que a partir de los valores de *hora*, *minuto* y *segundo* muestre la hora dentro de un segundo. Tras las 23:59:59 serán las 0:0:0.

Actividad 213

`213calendario.php`: Escribe un programa similar a un calendario de manera que a partir de *dia*, *mes* y *anyo* muestre la fecha dentro de un día. Debes tener en cuenta que no todos los meses tienen 30 días. En este caso, no vamos a tener en cuenta los años bisiestos.

Actividad 214

`214ecuacion2g.php`: Crea un programa que resuelva una ecuación de 2º grado del tipo $ax^2 + bx + c = 0$. Ten en cuenta que puede tener 2, 1 o no tener solución dependiendo del valor del discriminante $b^2 - 4ac$.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Nota: Para calcular la raíz cuadrada deberás utilizar la función `sqrt()`.

Actividad 215

Investiga para qué sirve el operador nave espacial, disponible desde PHP7 (<https://www.php.net/manual/es/migration70.new-features.php>). Explica con un par de líneas su propósito y mediante código demuestra su uso.

Actividad 216

Investiga para qué sirve la instrucción `match()`, disponible desde PHP8 (<https://www.php.net/manual/es/control-structures.match.php>). Explica con un par de líneas su propósito y mediante código demuestra su uso.

2. bucles

Actividad 221

`221pares050.php`: Escribe un programa que muestre los números pares del 0 al 50 (dentro de una lista desordenada).

`221paresAB.php`: A partir del anterior, refactorizar para que funcione con `inicio` y `fin`.

Actividad 222

`222suma110.php`: Escribe un programa que sume los números del 1 al 10.

`222sumaAB.php`: A partir del anterior, refactorizar para que funcione con `inicio` y `fin`.

Actividad 223

`223potencia.php`: A partir de una base y exponente, mediante la acumulación de productos, calcula la potencia utilizando la instrucción `for`.

`223potenciaWhile.php`: Reescribe el ejercicio anterior haciendo uso sólo de `while`.

`223potenciaDowhile.php`: Reescribe el ejercicio anterior haciendo uso sólo de `do-while`.

Actividad 224

`224tablaMultiplicar.php`: Muestra dentro de una tabla HTML la tabla de multiplicar del número que reciba como parámetro. Utiliza `<thead>` con sus respectivos `<th>` y `<tbody>` para dibujar la tabla. Por ejemplo:

a	*	b	=	a*b
7	*	1	=	7
7	*	2	=	14
...				
7	*	10	=	70

Actividad 225

`225formulario.html`: Crea un formulario que permita leer una cantidad.

`225leerDatos.php`: a partir de cantidad, prepara un formulario con tantas cajas de datos como su valor.

Finalmente, en `225sumarDatos.php`: a partir de los datos de todas las cajas de la página anterior, súmalos y muestra el total.

A tener el cuenta: Para guardar un dato oculto puedes utilizar un campo de formulario de tipo oculto: `<input type="hidden" name="cantidad" value="33" />`.

Actividad 226

`226formulario.html` y `226tabla.php`: A partir de un número de `filas` y `columnas`, crear una tabla con ese tamaño. Las celdas deben estar rellenas con los valores de las coordenadas de cada celda.

Actividad 227

`227formulario.html` y `227cuadrado.php`: Basándote en el ejercicio anterior, rellena la tabla de manera que solo los bordes tengan contenido, quedándose el resto de celdas en blanco.

Actividad 228

`228formulario.html` y `228equis.php`: Basándote en el ejercicio anterior, ahora sólo debe aparecer el contenido de los dos diagonales.

Actividad 229

`229cuadradoMultiplicar.php`: Crea un programa que muestre por pantalla un cuadrado exactamente igual (fíjate bien en los encabezados, tanto de las filas como de las columnas) al de la imagen con las tablas de multiplicar.

x	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10
2	0	2	4	6	8	10	12	14	16	18	20
3	0	3	6	9	12	15	18	21	24	27	30
4	0	4	8	12	16	20	24	28	32	36	40
5	0	5	10	15	20	25	30	35	40	45	50
6	0	6	12	18	24	30	36	42	48	54	60
7	0	7	14	21	28	35	42	49	56	63	70
8	0	8	16	24	32	40	48	56	64	72	80
9	0	9	18	27	36	45	54	63	72	81	90
10	0	10	20	30	40	50	60	70	80	90	100

3. arrays

Actividad 231

`231aleatorios50.php`: Rellena un array con 50 números aleatorios comprendidos entre el 0 y el 99, y luego muéstralo en una lista desordenada. Para crear un número aleatorio, utiliza la función `rand(inicio, fin)`. Por ejemplo:

```
1 | $num = rand(0, 99)
```

Actividad 232

`232bola8.html`: Prepara un formulario con un caja de texto que realice a una pregunta al usuario.

`232bola8.php`: A partir del anterior, crea un programa que muestre la pregunta recibida y genere una respuesta de manera aleatoria entre un conjunto de respuestas predefinidas, almacenadas en un array: *Si, no, quizás, claro que sí, por supuesto que no, no lo tengo claro, seguro, yo diría que sí, ni de coña, etc...*

Este ejercicio se basa en el juego de la [Bola 8 mágica](#).

Actividad 233

`233mates.php`: A partir del ejercicio 230, genera un array aleatorio de 33 elementos con números comprendidos entre el 0 y 100 y calcula:

- El mayor
- El menor
- La media

Actividad 234

`234sexos.php`: Rellena un array de 100 elementos de manera aleatoria con valores `M` o `F` (por ejemplo `["M", "M", "F", "M", ...]`). Una vez completado, vuelve a recorrerlo y calcula cuantos elementos hay de cada uno de los valores almacenando el resultado en un array asociativo `['M' => 44, 'F' => 66]` (no utilices variables para contar las `M` o las `F`). Finalmente, muestra el resultado por pantalla

Actividad 235

`235monedas.php`: Vuelve a realizar el ejercicio 207, el de las monedas (500, 200, 100, 50, 20, 10, 5, 2, 1), pero haciendo uso de arrays y un bucle. Almacena el resultado en un array asociativo. Muestra el resultado en una lista desordenada únicamente con las cantidades que tienen algún valor.

Actividad 236

`236alturas.php`: Mediante un array asociativo, almacena el nombre y la altura de 5 personas (`nombre => altura`). Posteriormente, recorre el array y muéstralo en una tabla HTML. Finalmente añade una última fila a la tabla con la altura media.

Actividad 237

`237personas.php`: Mediante un array bidimensional, almacena el nombre, altura y email de 5 personas. Para ello, crea un array de personas, siendo cada persona un array asociativo: `[['nombre'=>'Aitor', 'altura'=>182, 'email'=>'aitor@correo.com'], [...], ...]`. Posteriormente, recorre el array y muéstralo en una tabla HTML.

Actividad 238

`238leerCantidad.html` y `237leerPersonas.php`: a partir de un formulario con un campo de `cantidad` de personas, generar un nuevo formulario para leer el nombre, altura y email de `cantidad` personas.

`238gestionarPersonas.php`: A partir de las personas introducidas, mostrar sus datos en una tabla, y posteriormente, destacar los datos del más alto y el del más bajo.

Actividad 239

`239tablaDistintos.php`: Rellena un array bidimensional de 6 filas por 9 columnas con números aleatorios comprendidos entre 100 y 999 (ambos incluidos). Todos los números deben ser distintos, es decir, no se puede repetir ninguno.

Muestra a continuación por pantalla el contenido del array de tal forma que:

- La columna del máximo debe aparecer en azul.
- La fila del mínimo debe aparecer en verde
- El resto de números deben aparecer en negro.

4. funciones

Actividad 241

`241arrayPar.php`: Crea las siguientes funciones:

- Una función que averigüe si un número es par: `esPar(int $num): bool`
- Una función que devuelva un array de tamaño `$tam` con números aleatorios comprendido entre `$min` y `$max`: `arrayAleatorio(int $tam, int $min, int $max) : array`
- Una función que reciba un `$array` por referencia y devuelva la cantidad de números pares que hay almacenados: `arrayPares(array &$array): int`

Actividad 242

`242parametrosVariables.php`: Crea las siguientes funciones:

- Una función que devuelva el mayor de todos los números recibidos como parámetros: `function mayor(): int`. Utiliza las funciones `func_get_args()`, etc... No puedes usar la función `max()`.
- Una función que concatene todos los parámetros recibidos separándolos con un espacio: `function concatenar(...$palabras) : string`. Utiliza el operador ...

Actividad 243

`243matematicas.php`: Añade las siguientes funciones:

- `digitos(int $num): int` → devuelve la cantidad de dígitos de un número.
- `digitoN(int $num, int $pos): int` → devuelve el dígito que ocupa, empezando por la izquierda, la posición `$pos`.
- `quitaPorDetras(int $num, int $cant): int` → le quita por detrás (derecha) `$cant` dígitos.
- `quitaPorDelante(int $num, int $cant): int` → le quita por delante (izquierda) `$cant` dígitos.

Para probar las funciones, haz uso tanto de paso de argumentos posicionales como argumentos con nombre.

Actividad 244

`244biblioteca.php`: crea un archivo con funciones para sumar, restar, multiplicar y dividir dos números.

`244arrayFunciones.php`: haciendo uso de un array que almacene el nombre de las funciones del archivo anterior, a partir de dos números recibidos por URL, recorre el array e invoca a las funciones de manera dinámica haciendo uso de funciones variable.

Actividad 245

`245euros.php`: Crea una biblioteca con dos funciones:

- `peseta2euros`: pasa de pesetas a euros
- `euros2pesetas`: pasa de euros a pesetas

Cada función debe recibir dos parámetros:

- La cantidad a transformar
- La cotización, con un parámetro por defecto con el factor de transformación.

`245calculadoraEuros.php`: utiliza `244euros.php` y prueba las funciones pasando tanto cantidades con la cotización por defecto, como con nuevas cotizaciones. Recuerda que 1 euro son/eran 166.36 pesetas.

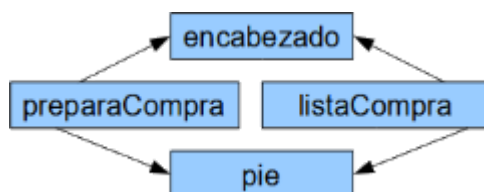
Actividad 246

`246preparaTiquetCompra.php`: A partir de una cantidad de productos, leer el nombre y coste de la cantidad de productos indicados (similar al ejercicio 237, pero esta vez no hace falta crear el formulario con la cantidad, se recibe mediante un parámetro GET via URL).

`246imprimeTiquetCompra.php`: Tras leer los datos del tiquet de compra, enumera en una tabla los productos, con su precio en euros y pesetas, y finalmente, en una última fila, totalizar en ambas monedas.

Actividad 247

A partir de los archivos creados en el ejercicio anterior, crea una plantilla mediante includes: `247preparaCompra.php`: similar a `246preparaTiquetCompra.php`, pero separando el encabezado (Supermercado Severo en `h1`) y el pie (Tu supermercado de confianza) en ficheros externos y referenciando a ellos mediante `include`.



`247listaCompra.php`: recibe los datos del anterior, y reutiliza parte de `246imprimeTiquetCompra.php` cambiando la tabla por una lista desordenada de los productos junto a su precio.

Actividad 248

Vamos a simular un formulario de acceso:

- `248login.php`: el formulario de entrada, que solicita el usuario y contraseña.
- `248compruebaLogin.php`: recibe los datos y comprueba si son correctos (los usuarios se guardan en un array asociativo) pasando el control mediante el uso de `include` a:
 - `248ok.php`: El usuario introducido es correcto
 - `248ko.php`: El usuario es incorrecto. Informar si ambos están mal o solo la contraseña. Volver a mostrar el formulario de acceso.

5. funciones predefinidas

Todos los ejercicios se deben realizar creando nuevas funciones para encapsular el código. Además de la propia función, el ejercicio debe contener código para poder probarlo.

Actividad 251

`251fraseImpares.php`: Lee una frase y devuelve una nueva con solo los caracteres de las posiciones impares.

Actividad 252

`252vocales.php`: A partir de una frase, devuelve la cantidad de cada una de las vocales, y el total de ellas.

Actividad 253

`253analizador.php`: A partir de una frase con palabras sólo separadas por espacios, devolver

- Letras totales y cantidad de palabras
- Una línea por cada palabra indicando su tamaño

Nota: no se puede usar `str_word_count`.

`253analizadorWC.php`: Investiga que hace la función `str_word_count`, y vuelve a hacer el ejercicio.

Actividad 254

`254cani.php`: EsCrIbE uNa FuNclóN qUe TrAnSfOrMe UnA cAdEnA eN cAnI.

Actividad 255

`255palindromo.php`: Escribe una función que devuelva un booleano indicando si una palabra es palíndroma (se lee igual de izquierda a derecha que de derecha a izquierda, por ejemplo, "ligar es ser agil").

Actividad 256

`256codificar.php`: Utilizando las funciones para trabajar con caracteres, a partir de una cadena y un desplazamiento:

- Si el desplazamiento es 1, sustituye la A por B, la B por C, etc.
- El desplazamiento no puede ser negativo
- Si se sale del abecedario, debe volver a empezar
- Hay que respetar los espacios, puntos y comas.

Actividad 257

`257filtrado.html`: Crea un programa que permita al usuario leer un conjunto de números separados por espacios.

`257filtrado.php`: El programa filtrará los números leídos para volver a mostrar únicamente los números pares e indicará la cantidad existente.

- 1 Dame números: 1 4 7 9 23 10 8
- 2 Los 3 números pares son: 4 10 8

Actividad 258

`258investiga.php`: Investiga las siguientes funciones de cadena (explica para qué sirven mediante comentarios, y programa un pequeño ejemplo de cada una de ellas): `ucwords`, `strrev`, `str_repeat` y `md5`.

Los siguientes ejercicios se basan en la generación de números aleatorios.

Actividad 259

`259generador.php`: Crea una función que permite generar una letra aleatoria, mayúscula o minúscula.

Actividad 260

`260generaContrasenya.php`: Crea una función que a partir de un tamaño, genera una contraseña aleatoria compuesta de letras y dígitos de manera aleatoria.

Actividad 261

`261quinielas.php`: Crea las siguientes funciones:

- `quinigol()` : array → Genera un array multidimensional con 6 resultados aleatorios con combinaciones `[012M, 012M]`.
- `quiniela()` : array → Genera un array con una combinación de quiniela generada de manera aleatoria: 14 resultados con 1X2 y el pleno al quince con `[012M, 012M]`.
- `tabla(array $quiniela)` : string → transforma un array de una quiniela en una tabla HTML.

6. otros ejercicios

Aquí entontrarás ejercicios genéricos de programación para usar con cualquier lenguaje. Son genéricos porque te pueden servir con cualquier lenguaje que quieras aprender.

- Ejercicios [PHP de W3Schools](#)
- Ejercicios [PHP de JavaTpoint](#)

7. ejercicios del concurso ACEPTA EL RETO

Estos son 10 ejercicios para resolver de la excelente propuesta [Acepta el Reto](#) .

Deberían resolverse en menos de 2h cada uno (como mucho).

1. [Códigos de Barras](#)
2. [Encriptación de Mensajes](#)
3. [¿Cuántos números capicúa?](#)
4. [Feliz No Cumpleaños](#)
5. [Acepta la Suma](#)
6. [Sudokus Correctos](#)
7. [Liga de Pádel](#)
8. [Escudos del ejército romano](#)
9. [Las Cartas del Abuelo](#)
10. [Manías tenemos todos](#)