

# Array PHP

Slide tratte da:

**W3CSCHOOLS.COM**

( [https://www.w3schools.com/php/php\\_arrays.asp](https://www.w3schools.com/php/php_arrays.asp) )

## Introduzione agli Array in PHP

- Un array è una variabile speciale che può contenere più di un valore alla volta.
- Se hai una lista di elementi (ad esempio una lista di nomi di auto), memorizzarli in variabili separate potrebbe diventare scomodo.
- Gli array sono strumenti potenti per gestire insiemi di dati.

## Cos'è un Array?

- Un array è una variabile speciale in grado di contenere più valori contemporaneamente.
- Utile per gestire elenchi di elementi, come nomi di persone.
- Evita la necessità di dichiarare variabili separate per ciascun elemento..

## Creare un Array in PHP

- In PHP, si utilizza la funzione `array()` per creare un array.
- Tre tipi di array in PHP:
  - Array indicizzati (con indici numerici)
  - Array associativi (con chiavi denominate)
  - Array multidimensionali (contenenti uno o più array)

## Lunghezza di un Array - La Funzione `count()`

- La funzione `count()` restituisce la lunghezza (il numero di elementi) di un array.
- Utile per determinare quante voci ci sono in un array.

## Array indicizzati in PHP

- Due modi per creare array indicizzati.
  - Automaticamente (gli indici partono da 0):

```
PHP
$cars = array("Volvo", "BMW", "Toyota");
```

- Manualmente:

```
PHP
$cars[0] = "Volvo"; $cars[1] = "BMW"; $cars[2] = "Toyota";
```

## Scorrere un array indicizzato

- Accesso agli elementi dell'array mediante indici numerici.
- Per scorrere e stampare tutti i valori di un array indicizzato, si usa un ciclo

```
PHP
$cars = array("Volvo", "BMW", "Toyota");
$arlength = count($cars);

for ($x = 0; $x < $arlength; $x++) {
    echo $cars[$x];
    echo "<br>";
}
```

## Array associativi in PHP

- Gli array associativi utilizzano chiavi denominate per gli elementi.
- Due modi per crearli:

```
PHP
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

```
PHP
$age['Peter'] = "35"; $age['Ben'] = "37"; $age['Joe'] = "43";
```

N.B. Gli indici sono **chiavi**, non numeri.

## Scorrere un array associativo

- Per scorrere e stampare tutti i valori di un array associativo, si usa un ciclo foreach.
- Esempio:

```
PHP
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

foreach($age as $x => $x_value) {
    echo "Chiave=" . $x . ", Valore=" . $x_value;
    echo "<br>";
}
```

## Array multidimensionali in PHP

- Gli array multidimensionali contengono uno o più array.
- PHP supporta array a due, tre, quattro o più livelli di profondità.
- La dimensione di un array indica quanti indici servono per selezionare un elemento.

## Array bidimensionali in PHP

- Un array bidimensionale è un array di array.
- Utile per rappresentare dati tabulari.

```
PHP
$scars = array (
    array("Volvo",22,18),
    array("BMW",15,13),
    array("Saab",5,2),
    array("Land Rover",17,15)
);
```

## Scorrere un array bidimensionale

- Per accedere agli elementi di un array bidimensionale, è necessario specificare due indici (riga e colonna).

PHP

```
echo $cars[0][0].": In stock: ".$cars[0][1].",  
sold:".$cars[0][2]."<br>";  
echo $cars[1][0].": In stock: ".$cars[1][1].", sold:  
".$cars[1][2]."<br>";
```

```
echo $cars[$riga][$colonna];
```

## Ordinamento degli array in PHP

- PHP fornisce diverse funzioni per ordinare gli array.
- Esempi di funzioni di ordinamento: `sort()`, `rsort()`, `asort()`, `ksort()`, `arsort()`, `krsort()`.
- Possono essere utilizzate per ordinare array in ordine crescente o decrescente, secondo i valori o le chiavi.

## Ordinamento di un array in ordine crescente

- La funzione `sort()` ordina un array in ordine alfabetico o numerico crescente.

PHP

```
$cars = array("Volvo", "BMW", "Toyota");  
sort($cars);
```

## Ordinamento di un array in ordine decrescente

- La funzione `rsort()` ordina un array in ordine alfabetico o numerico decrescente.

PHP

```
$cars = array("Volvo", "BMW", "Toyota");  
rsort($cars);
```

## Ordinamento di un array associativo per valore

- La funzione `asort()` ordina un array associativo in ordine crescente in base ai valori.

PHP

```
$age = array("Gio"=>"37", "Ben"=>"35", "Ale"=>"43");  
asort($age);
```

## Ordinamento di un array associativo per chiave

- La funzione `ksort()` ordina un array associativo in ordine crescente in base alle chiavi.

PHP

```
$age = array("Gio"=>"37", "Ben"=>"35", "Ale"=>"43");  
ksort($age);
```

## Ordinamento di un array associativo per valore in ordine decrescente

- La funzione `arsort()` ordina un array associativo in ordine decrescente in base ai valori.

PHP

```
$age = array("Gio"=>"37", "Ben"=>"35", "Ale"=>"43");  
arsort($age);
```

## Ordinamento di un array associativo per chiave in ordine decrescente

- La funzione `krsort()` ordina un array associativo in ordine decrescente in base alle chiavi.

PHP

```
$age = array("Gio"=>"37", "Ben"=>"35", "Ale"=>"43");  
krsort($age);
```

# Abbiamo finito!

Ora è importante fare  
parecchi esercizi!

