

Scheda Informativa sugli Array in PHP

Gli **array** in PHP sono strutture di dati flessibili che possono contenere una collezione di valori, identificabili tramite **chiavi**. PHP supporta diverse tipologie di array, inclusi array indicizzati, associativi e multidimensionali.

Tipologie di Array:

In PHP, esistono **tre tipi principali di array**:

1. **Array indicizzati** – Gli elementi sono memorizzati con chiavi numeriche (es.

```
$arr = [10, 20, 30]; ).
```

2. **Array associativi** – Utilizzano chiavi personalizzate (es. `$arr = ["nome" =>`

```
"Mario", "età" => 25]; ).
```

3. **Array multidimensionali** – Contengono altri array come elementi (es. `$arr =`

```
[[1, 2], [3, 4]]; ).
```

4. Array indicizzati:

- Gli elementi sono indicizzati da chiavi numeriche intere.
- Indice automatico: Se non viene specificato, PHP assegna un indice crescente.
- Esempio:

```
$numeri = [1, 2, 3, 4];
```

5. Array associativi:

- Gli elementi sono indicizzati da chiavi personalizzate (stringhe).
- Utili per rappresentare coppie chiave-valore.
- Esempio:

```
$persona = [  
    "nome" => "Mario",  
    "età" => 25,  
    "città" => "Roma"  
];
```

6. Array multidimensionali:

- Un array che contiene altri array come elementi.
- Permette di creare strutture complesse come matrici o tabelle.
- Esempio:

```
$matrice = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
];
```

Funzioni principali sugli Array in PHP:

Funzione	Descrizione
<code>array()</code>	Crea un array.
<code>count()</code>	Restituisce il numero di elementi di un array.
<code>array_merge()</code>	Unisce due o più array.
<code>array_push()</code>	Aggiunge uno o più elementi alla fine di un array.
<code>array_pop()</code>	Rimuove e restituisce l'ultimo elemento di un array.
<code>array_shift()</code>	Rimuove e restituisce il primo elemento di un array.
<code>array_unshift()</code>	Aggiunge uno o più elementi all'inizio di un array.
<code>array_key_exists()</code>	Verifica se una data chiave esiste in un array.
<code>in_array()</code>	Controlla se un valore esiste in un array.
<code>array_keys()</code>	Restituisce tutti i valori di chiavi.

Dichiarazione e Accesso agli Elementi:

- **Creazione di un array:**

```
$frutta = ["mela", "banana", "pera"];
```

- **Accesso agli elementi:**

Gli elementi dell'array possono essere acceduti tramite l'indice o la chiave.

```
echo $frutta[1]; // banana
```

- **Aggiunta di elementi:**

```
$frutta[] = "arancia"; // Aggiunge alla fine dell'array
```

Loop sugli Array:

Gli array in PHP possono essere percorsi utilizzando diversi tipi di cicli, il più comune è `foreach`.

- **Ciclo** `foreach`:

```
foreach ($frutta as $frutto) {  
    echo $frutto;  
}
```

- **Ciclo** `foreach` **su array associativo:**

```
foreach ($persona as $chiave => $valore) {  
    echo "$chiave: $valore";  
}
```


Array Multidimensionali:

Gli array multidimensionali possono essere percorsi utilizzando cicli annidati.

Esempio:

```
$matrice = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
];  
  
foreach ($matrice as $riga) {  
    foreach ($riga as $valore) {  
        echo $valore . " ";  
    }  
}
```

Funzioni Avanzate sugli Array:

- **array_map()** : Applica una funzione a ogni elemento di un array.

```
$numeri = [1, 2, 3];  
$quadrati = array_map(fn($n) => $n * $n, $numeri);
```

- **array_filter()** : Filtra gli elementi di un array usando una funzione di callback.

```
$numeri = [1, 2, 3, 4, 5];  
$numeri_pari = array_filter($numeri, fn($n) => $n % 2 == 0);
```

- **array_reduce()** : Riduce un array a un singolo valore.

```
$numeri = [1, 2, 3, 4];  
$somma = array_reduce($numeri, fn($carry, $item) => $carry + $item, 0);
```

Caratteristiche degli Array in PHP:

- **Tipizzazione debole:** PHP permette array misti, ovvero array che possono contenere diversi tipi di dati (interi, stringhe, ecc.).
- **Dinamici:** Gli array in PHP non richiedono una dimensione fissa e possono crescere o ridursi dinamicamente.
- **Chiavi automatiche:** PHP assegna automaticamente chiavi numeriche incrementali quando gli elementi vengono aggiunti senza specificare una chiave.

Esempio Completo:

```
$auto = [  
  "marca" => "Toyota",  
  "modello" => "Corolla",  
  "anno" => 2020,  
  "caratteristiche" => [  
    "colore" => "rosso",  
    "potenza" => "120 CV"  
  ]  
];  
  
echo $auto["marca"]; // Toyota  
echo $auto["caratteristiche"]["colore"]; // rosso
```