

Videocorso PHP



I vettori

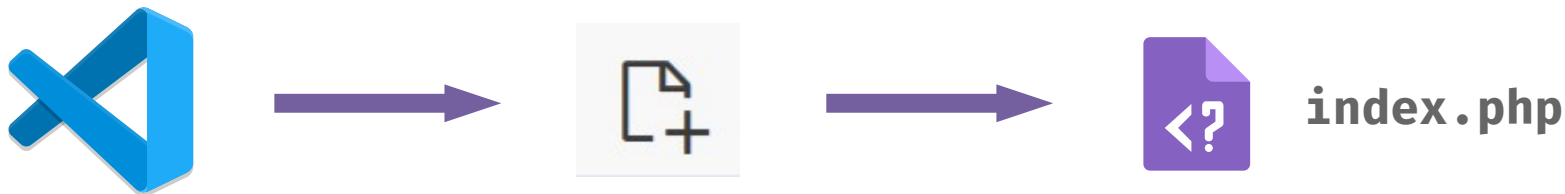
la prima vera novità di PHP

Alessandro Flora

I vettori in PHP

Come sarà impostata questa lezione

- All'interno delle slide ci saranno alcuni riferimenti utili per il successivo ripasso
- In particolare nelle slide sono riportati i concetti di base e la sintassi
- La seconda parte sarà un esercizio svolto; è vivamente consigliato provare a ragionare insieme creando uno script per provare in prima persona



I vettori in PHP

Cos'è un vettore (in generale/linguaggio di riferimento C)

- Tipo di dato composto da più dati dello stesso tipo, ad esempio numeri interi
- Dimensione fissa predeterminata (al massimo sovradimensioniamo il vettore e lo usiamo in parte)
- Accessibile mediante gli indici o i puntatori (gli indici partono da 0 perciò $a[0] = 1$ nell'esempio riportato)

```
int a[5] = {1,2,3,4,5};
```



1	2	3	4	5
---	---	---	---	---

I vettori in PHP

Cos'è un vettore in PHP

- Struttura dati decisamente più flessibile
- Sono delle liste di valori di tipo non fissato (stringhe, numeri, oggetti, altri vettori, ...)
- Possono essere classici (con indici da 0 alla lunghezza - 1) oppure associativi

```
$a = array(1, 'ciao', true, 'c', 12.4);
```



1	'ciao'	true	'c'	12.4
---	--------	------	-----	------

I vettori in PHP

Distinzione dei vettori in PHP

Distingueremo due modelli:

- Classico: ogni elemento è associato a un indice che varia tra 0 e la lunghezza - 1
- Associativo: ogni elemento è associato a una chiave univoca che può essere di qualunque tipo (stringa, numero, booleano, ...)

A differenza di linguaggi come Python gli array associativi hanno un ordine nel quale elencate le chiavi quando stampiamo o iteriamo sul vettore.

I vettori in PHP

Come inizializziamo un vettore classico in PHP

```
$a = array(  
    'valore1',  
    2,  
    'valore3',  
    true  
);
```

```
$a = array('valore1',2,'valore3',true);
```

```
$a = array();
```

```
$a = [  
    'valore1',  
    2,  
    'valore3',  
    true  
];
```

```
$a = ['valore1',2,'valore3',true];
```

```
$a = [];
```



Con dei valori di
partenza



Vuoti

I vettori in PHP

Come inizializziamo un vettore associativo in PHP

```
$a = [  
    'chiave1' => 'valore1',  
    2 => 'valore2',  
    3.45 => true  
];
```



Le chiavi devono per forza essere univoche, i valori possono anche essere duplicati



Come prima possiamo scegliere se usare le quadre o la funzione array



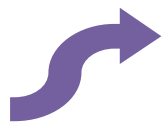
Come prima possiamo anche scrivere tutto su una riga, il codice sarà più compatto ma un po' meno leggibile

I vettori in PHP

Come stampiamo un vettore

NB: è consigliatissimo provare le funzioni seguenti aiutandosi con questa funzione

```
$a = [  
    'chiave1' => 'valore1',  
    'chiave2' => 'valore2',  
    5 => 2,  
    2.3 => true  
];  
  
print_r($a);
```



Stampa come HTML con la seguente notazione il contenuto del vettore



Array ([chiave1] => valore1 [chiave2] => valore2 [5] => 2 [2] => 1)

Se il vettore passato alla funzione fosse classico le chiavi sarebbero degli interi

I vettori in PHP

Come aggiungiamo e modifichiamo valori: vettore classico

```
$a = [];  
$b = 2;  
array_push($a, $b);
```



Aggiunge al vettore **\$a** tutti i parametri successivi (che possono essere di qualunque numero)

```
array_pop($a);
```



Rimuove l'ultimo elemento del vettore **\$a**

```
array_shift($a);
```



Rimuove il primo elemento del vettore **\$a**

```
$a[0] = 1;
```



Assegna all'indice 0 del vettore **\$a** il valore 1 (se esisteva già viene modificato altrimenti viene aggiunto)

```
array_splice($a, 1, 3);
```



Rimuove tre elementi del vettore **\$a** a partire da quello a indice 1 (escluso)

I vettori in PHP

Come aggiungiamo e modifichiamo valori: vettore associativo

```
$a = [];  
$a['chiave1'] = 'valore1';
```



Assegna all'indice **chiave1** del vettore **\$a** il valore **valore1** (se esisteva già viene modificato altrimenti viene aggiunto)

Se usata sui vettori classici non aggiorna gli indici, quindi la useremo solo sugli associativi



```
unset($a['chiave1']);
```



Rimuove il dato all'indice **chiave1** del vettore **\$a**, se non esiste tale chiave non fa nulla

I vettori in PHP

Funzioni avanzate sui vettori

`count($a);` Restituisce il numero di elementi del vettore `$a`

`array_key_exists('chiave1', $a);` Controlla se la chiave **chiave1** appartiene al vettore `$a`

`array_search(5, $a);` Restituisce la chiave di `$a` corrispondente al valore **5** oppure **false** se **5** \notin `$a`

`array_keys($a);` Restituisce un vettore contenente le chiavi di `$a`

`array_values($a);` Restituisce un vettore contenente i valori di `$a`

I vettori in PHP

Funzioni avanzate sui vettori

```
array_combine($chiavi,$valori);
```

Restituisce un vettore prendendo le chiavi da **\$chiavi** e i valori da **\$valori** (devono avere le stesse dimensioni)

```
array_walk($a,'stampaVettore');
```

Chiama la funzione da noi definita **stampaVettore** per ogni elemento del vettore **\$a** passando alla stessa come parametri la chiave e il valore

I vettori in PHP

Ordinamento di vettori

`sort($a);` Ordina **\$a** in ordine crescente

`rsort($a);` Ordina **\$a** in ordine decrescente



\$a è un vettore classico

`asort($a);` `arsort($a);`

Ordina **\$a** in ordine crescente
e decrescente per valori

`ksort($a);` `krsort($a);`

Ordina **\$a** in ordine crescente
e decrescente per chiavi



\$a è un vettore associativo

I vettori in PHP

Il costrutto foreach

Consente di iterare su un vettore estraendo il valore oppure chiave e valore per ogni elemento del vettore

```
foreach($a as $chiave=>$valore) {  
    // codice  
}
```

```
foreach($a as $valore) {  
    // codice  
}
```



Iteriamo elemento per
elemento sul vettore **\$a**