

GABRIELE ROMANATO

Utilizzo della funzione `filter_var()` in PHP

La funzione `filter_var()` è uno strumento potente e versatile offerto da PHP per validare e filtrare dati provenienti da input esterni, come moduli HTML, URL, indirizzi email, e molto altro. È particolarmente utile per migliorare la sicurezza e l'affidabilità delle applicazioni web.

Sintassi

```
filter_var(mixed $value, int $filter = FILTER_DEFAULT,  
array|int $options = 0): mixed
```

- **\$value**: Il valore da filtrare o validare.
- **\$filter**: Il filtro da applicare. Se non specificato, viene usato `FILTER_DEFAULT`.
- **\$options**: Un array associativo di opzioni o una costante con flag.

Esempi di utilizzo

Validazione di un indirizzo email

```
$email = "utente@example.com";  
  
if (filter_var($email, FILTER_VALIDATE_EMAIL)) {  
    echo "Indirizzo email valido.";  
} else {  
    echo "Indirizzo email non valido.";  
}
```

Filtraggio di un numero intero

```
$number = "123";

$is_integer = filter_var($numero, FILTER_VALIDATE_INT);

if (!$is_integer) {
    echo "Valore intero: $integer";
} else {
    echo "Il valore non è un intero valido.";
}
```

Sanificazione di una stringa

```
$input = "<h1>Ciao</h1>";

$sanitized = filter_var($input, FILTER_SANITIZE_STRING);

echo $sanitized;
```

Nota: FILTER_SANITIZE_STRING è deprecato a partire da PHP 8.1. Si consiglia di utilizzare funzioni alternative come htmlspecialchars() per la sanificazione dell'output HTML.

Altri filtri disponibili

PHP fornisce una varietà di filtri per validare e sanificare i dati, tra cui:

- FILTER_VALIDATE_BOOLEAN
- FILTER_VALIDATE_FLOAT
- FILTER_SANITIZE_EMAIL
- FILTER_SANITIZE_URL

Conclusione

La funzione `filter_var()` è uno strumento fondamentale per gestire dati provenienti da input esterni in modo sicuro. Grazie alla sua flessibilità, può essere facilmente integrata in qualsiasi flusso di validazione e sanificazione dei dati in PHP.