

Come prelevare i prodotti nelle categorie speciali in PrestaShop

domenica, 25 marzo 2018

Spesso nello sviluppo di moduli di **PrestaShop**, o semplicemente nel modificarne uno già esistente, potresti avere la necessità di prelevare un elenco di prodotti, magari con una caratteristica particolare in comune. **Prestashop** ha molte funzioni che consentono di risparmiare tempo e che possiamo utilizzare a questo scopo.

Vediamo come ottenere vari prodotti e gestirli.



Sommario

Prelevare i prodotti da una categoria tramite ID

Recuperare i prodotti scontati

Recuperare i prodotti più venduti

Recuperare i nuovi arrivi

Prelevare i prodotti da una categoria tramite ID

La funzione che andremo ad utilizzare, è un metodo della classe Category, che si trova in:

classes/Category.php

Ecco la definizione del metodo:

```
public function getProducts( $id_lang, $p, $n, $order_by = null, $order_way = null,
$get_total = false, $active = true, $random = false, $random_number_products = 1,
$check_access = true, Context $context = null );
```

I parametri utilizzati dal metodo, sono:

\$id_lang: ID della lingua in cui vogliamo ottenere i prodotti.

\$p: numero di pagina, associato al successivo parametro.

\$n: numero di prodotti per pagina.

\$order_by: nome della colonna per cui si preferisce ordinare i prodotti recuperati.

\$order_way: la modalità di ordinamento (crescente/decrescente).

\$get_total: se impostato su true, restituisce solamente il numero totale di prodotti presenti nella categoria. Annulla il valore di \$p ed \$n.

\$active: se impostato su true, restituisce solo i prodotti attivi.

\$random: se true, recupera i prodotti in modalità casuale. Annulla il valore di \$order_by e \$order_way.

\$random_number_products: numero di prodotti da restituire se è attivato random.

\$check_access: se impostato su true, verifica se il cliente corrente può vedere i prodotti di questa categoria.

Ma vediamo un esempio pratico.

Ecco il metodo migliore per ottenere prodotti da qualsiasi categoria in **PrestaShop**, usato non solo nelle pagine delle categorie, ma anche da alcuni dei moduli di default:

```
// Recupera i primi 10 prodotti di una categoria
$category = new Category( 3 );
$products = $category->getProducts( $this->context->language->id, 1, 10 );
```

Nota come siano stati valorizzati solo i parametri obbligatori.

```
// Recupera il numero di prodotti totali per quella categoria
$category = new Category( 3 );
$products = $category->getProducts( $this->context->language->id, 1, 10, 'price', 'asc',
true );
```

Nonostante siano stati impostati il campo di ordinamento e il metodo di ordinamento, essendo *true* il sesto parametro (*\$get_total*), verrà restituito solamente un intero corrispondente al numero totale di prodotti presenti nella categoria.

```
// Ottiene 10 prodotti in ordine casuale
$category = new Category( 3 );
$products = $category->getProducts( $this->context->language->id, 1, 10, 'price', 'asc',
false, true );
```

Nonostante siano stati impostati il campo di ordinamento e il metodo di ordinamento, i prodotti verranno restituiti in ordine casuale, dal momento che è impostato su *true* l'ottavo parametro (*\$random*).

Recuperare i prodotti scontati

Il metodo che andiamo ad utilizzare è:

```
Product::getPricesDrop( $id_lang, $page_number = 0, $nb_products = 10, $count = false,  
$order_by = null, $order_way = null, $beginning = false, $ending = false, Context  
$context = null );
```

metodo statico della classe Product, che si trova in:

classes/Product.php

In questo caso, trattandosi di una metodo statico, non è necessario istanziare un oggetto della classe Product:

```
// Recupera 10 prodotti scontati  
$products = Product::getPricesDrop( (int)$this->context->language->id, 0, 10 );  
  
// Recupera 10 prodotti scontati, in ordine di prezzo crescente, ma la cui offerta parte  
il 01 gennaio 2018  
$products = Product::getPricesDrop( (int) $this->context->language->id, 0, 10, false,  
'price', 'asc', '2018-01-01 00:00:00' );
```

Recuperare i prodotti più venduti

Il metodo che andiamo ad utilizzare è:

```
ProductSale::getBestSales( $id_lang, $page_number = 0, $nb_products = 10, $order_by = null, $order_way = null );
```

definita sempre nella classe `Product`.

Ecco un esempio:

```
// Recupera i 10 prodotti piÃ¹ venduti  
$products = ProductSale::getBestSales( $this->context->language->id, 0, 10 );
```

Recuperare i nuovi arrivi

Il metodo che andiamo ad utilizzare è:

```
Product::getNewProducts( $id_lang, $page_number = 0, $nb_products = 10, $count = false,  
$order_by = null, $order_way = null, Context $context = null );
```

definita anche questa nella class Product.

Come utilizzarla? È presto detto:

```
// Preleva 10 prodotti considerati come nuovi arrivi  
$products = Product::getNewProducts( (int) $this->context->language->id, 0, 10 );
```