

`wc\_get\_products`



□□

- [□□□□](#)
 - [□□](#)
- [API □□](#)
- [□□□□□□](#)
 - [General](#)
 - [□□](#)
 - [□□](#)
 - [□□](#)
 - [□□□□](#)
 - [□□](#)
 - [□□□□□](#)
 - [□□](#)
 - [□□□□](#)
 - [□□](#)
 - [□□□□□□□□](#)
 - [□□](#)
 - [□□□□□](#)
 - [□□](#)
 - [□□□□](#)
 - [□□](#)
 - [□□](#)
 - [□□□□](#)
 - [□□](#)
 - [□□□□□□□□□□](#)

wc_get_products() WC_Product_Query
WooCommerce WP_Queries
WooCommerce
wc_get_products() WC_Product_Query
WordPress() get_posts() WP_Query()

```
// 2016 年間の商品を取得
$products = wc_get_products( array(
    'downloadable' => true,
    'date_created' => '2016-01-01...2016-12-31',
) );
```

```
// 10 個の商品 ID を取得
$query = new WC_Product_Query( array(
    'limit' => 10,
    'orderby' => 'date',
    'order' => 'DESC',
    'return' => 'ids',
) );
$products = $query->get_products();
```

```
// SKU を取得
// "PRDCT-1", "PRDCT-2" の SKU を取得
$query = new WC_Product_Query();
$query->set( 'sku', 'PRDCT' );
$products = $query->get_products();
```

wc_get_products() WC_Product_Query::get_products()

API

関数	説明
<code>wc_get_products(\$args)</code>	配列 <code>\$args</code> を取得
<code>WC_Product_Query::get_query_vars()</code>	取得可能なクエリ変数の一覧
<code>WC_Product_Query::get(string \$key, mixed \$default = '')</code>	取得可能なクエリ変数の一覧
<code>WC_Product_Query::set(string \$key, mixed \$value)</code>	クエリ変数の設定
<code>WC_Product_Query::get_products()</code>	取得可能なクエリ変数の一覧

取得可能なクエリ変数の一覧 / 関数



General

パラメータ	説明
status	ステータス。デフォルトは <code>'draft'</code> 。他の有効な値は <code>'pending'</code> , <code>'private'</code> , <code>'publish'</code> 。詳細は ProductStatus を参照。
type	タイプ。デフォルトは <code>'simple'</code> 。他の有効な値は <code>'external'</code> , <code>'grouped'</code> , <code>'simple'</code> , <code>'variable'</code> 。詳細は ProductType を参照。
include	含める製品の ID の配列
exclude	除外する製品の ID の配列
parent	親製品の ID
parent_exclude	除外する親製品の ID の配列
order	並び順。デフォルトは <code>'ASC'</code> 。他の有効な値は <code>'DESC'</code> 。詳細は orderby を参照。
orderby	並び順の基準。デフォルトは <code>'none'</code> 。他の有効な値は <code>'ID'</code> , <code>'name'</code> , <code>'type'</code> , <code>'rand'</code> , <code>'date'</code> , <code>'modified'</code> 。詳細は orderby を参照。
return	返す形式。デフォルトは <code>'ids'</code> 。他の有効な値は <code>'objects'</code> 。詳細は return を参照。



```
// 取得可能なクエリ変数の一覧
$products = wc_get_products( array( 'status' => 'draft' ) );
```

```
// 取得草稿商品
$products = wc_get_products( array( 'status' =>
\Automattic\WooCommerce\Enums\ProductStatus::DRAFT ) );
```

```
// 取得外部商品
$products = wc_get_products( array( 'type' => 'external' ) );
```

```
// 取得指定 ID 的商品
$args = array(
    'type' => 'external',
    'include' => array( 134, 200, 210, 340 ),
);
$products = wc_get_products( $args );
```

```
// 取得排除指定 ID 的商品
$products = wc_get_products( array( 'exclude' => array( $product->get_id() ) ) );
```

```
// 取得指定父级 ID 的商品
$products = wc_get_products( array( 'parent' => 20 ) );
```

```
// 取得指定排序的商品
$args = array(
    'orderby' => 'modified',
    'order' => 'DESC',
);
$products = wc_get_products( $args );
```

```
// 取得指定排序的商品
$products = wc_get_products( array( 'orderby' => 'rand' ) );
```

```
// 取得指定 ID 的商品
$products = wc_get_products( array( 'return' => 'ids' ) );
```



項目	説明
limit	取得する商品の数 posts_per_page 設定値 -1

변수	설명
page	현재 페이지 번호 (기본값: 1) 'offset'
offset	오프셋 값 (기본값: 0)
paginate	페이징 여부 (true/false) products total (총 개수) max_num_pages (최대 페이지 수)

예제

```
// 페이지 3 출력
$products = wc_get_products( array( 'limit' => 3 ) );
```

```
// 페이지 3 출력
$args = array(
    'limit' => 3,
    'page' => 1,
);
$page_1_products = wc_get_products( $args );

// 페이지 3 출력
$args = array(
    'limit' => 3,
    'page' => 2,
);
$page_2_products = wc_get_products( $args );
```

```
// 페이징된 결과 출력
$results = wc_get_products( array( 'paginate' => true ) );
echo $results->total . " 개\n";
echo '1 페이지 ' . $results->max_num_pages . " 페이지\n";
if ( count( $results->products ) > 0 ) {
    echo '첫 번째 ID: ' . $results->products[0]->get_id() . "\n";
}
```

```
// 페이지 4 출력
$args = array(
    'limit' => 4,
```

```
'offset' => 1,
);
$products = wc_get_products( $args );
```

Table

Field	Description
sku	SKU (Stock Keeping Unit) - A unique identifier for each product. Example: PRDCT-1, PRDCT-2
name	Product Name - The name of the product. Example: Test Product
tag	Product Tag - A tag used to categorize products. Example: Excellent, Modern
product_tag_id	Product Tag ID - The ID of the product tag. Example: 17, 23
category	Product Category - The category of the product. Example: shirts
product_category_id	Product Category ID - The ID of the product category. Example: 17, 23

Code

```
// Filter by SKU
$products = wc_get_products( array( 'sku' => 'PRDCT' ) );
```

```
// Filter by Name
$products = wc_get_products( array( 'name' => 'Test Product' ) );
```

```
// Filter by Tag
$products = wc_get_products( array( 'tag' => array( 'excellent', 'modern' ) ) );
```

```
// Filter by Product Tag ID
$products = wc_get_products( array( 'product_tag_id' => array( 17, 23 ) ) );
```

```
// Filter by Category
$products = wc_get_products( array( 'category' => array( 'shirts' ) ) );
```

```
// Filter by Product Category ID
$products = wc_get_products( array( 'product_category_id' => array( 17, 23 ) ) );
```



属性名	属性値
weight	重量
length	長さ
width	幅
height	高さ
price	価格
regular_price	通常価格
sale_price	セール価格
total_sales	総売上



```
// 幅 5.5、長さ 10 の商品
$args = array(
    'width' => 5.5,
    'length' => 10,
);
$products = wc_get_products( $args );

// 価格 9.99 の商品
$products = wc_get_products( array( 'price' => 9.99 ) );

// 総売上 0 の商品
$products = wc_get_products( array( 'total_sales' => 0 ) );
```



属性名	属性値
virtual	仮想商品
downloadable	ダウンロード可能
featured	特選商品
sold_individually	個別販売
manage_stock	在庫管理

属性名	値
reviews_allowed	boolean
backorders	boolean 'yes', 'no' ない 'notify' あり
visibility	boolean 'visible', 'catalog', 'search' あり 'hidden' あり ない CatalogVisibility あり ない
download_limit	integer -1 無限
download_expiry	integer -1 無限

コード

```
// 検索可能な商品を取得
$args = array(
    'downloadable' => true,
    'reviews_allowed' => false,
);
$products = wc_get_products( $args );
```

```
// 後払い商品を取得
$products = wc_get_products( array( 'backorders' => 'yes' ) );
```

```
// カタログ商品を取得
$products = wc_get_products( array( 'visibility' => 'catalog' ) );
```

```
// 検索可能な商品を取得
$products = wc_get_products( array( 'visibility' =>
\Automattic\WooCommerce\Enums\CatalogVisibility::CATALOG ) );
```

```
// 無限ダウンロード商品を取得
$products = wc_get_products( array( 'download_limit' => -1 ) );
```

在庫ステータス

属性名	値
stock_quantity	integer
stock_status	boolean 'outofstock' (あり), 'instock' (あり), ない 'onbackorder' (あり) ない ProductStockStatus あり ない

□□

```
// □□□□□□□□□□□□□□  
$products = wc_get_products( array( 'stock_quantity' => 1 ) );
```

```
// □□□□□□□□  
$products = wc_get_products( array( 'stock_status' => 'outofstock' ) );
```

```
// □□□□□□□□□□□□□□  
$products = wc_get_products( array( 'stock_status' =>  
\Automattic\WooCommerce\Enums\ProductStockStatus::OUT_OF_STOCK ) );
```

□□□□□□

□□	□□
tax_status	□□□□□□□□ 'taxable' (□□), 'shipping' (□□□□), □ 'none' (□□)□ □□ ProductTaxStatus □□□□ □
tax_class	□□□□□□□□□□
shipping_class	□□□□□□□□□□□□□□□□□□□□

□□

```
// □□□□□□□□  
$products = wc_get_products( array( 'tax_status' => 'taxable' ) );
```

```
// □□□□□□□□□□□□□□  
$products = wc_get_products( array( 'tax_status' =>  
\Automattic\WooCommerce\Enums\ProductTaxStatus::TAXABLE ) );
```

```
// □□□□ "Reduced Rate" □□□□□□□□  
$products = wc_get_products( array( 'tax_class' => 'reduced-rate' ) );
```

```
// □□□□ "Bulky" □□□□□□□□□□  
$products = wc_get_products( array( 'shipping_class' => 'bulky' ) );
```

□□□□□□

項目	型
average_rating	浮動小数点
review_count	整数

例

```
// 平均評価が 5 以上の商品
$products = wc_get_products( array( 'average_rating' => 5.0 ) );
```

```
// 評価数 1 以上の商品
$products = wc_get_products( array( 'review_count' => 1 ) );
```

例

日付範囲指定

項目	型
date_created	日付 (YYYY-MM-DD) 時間 (HH:MM:SS)
date_modified	日付 (YYYY-MM-DD) 時間 (HH:MM:SS)
date_on_sale_from	日付 (YYYY-MM-DD) 時間 (HH:MM:SS)
date_on_sale_to	日付 (YYYY-MM-DD) 時間 (HH:MM:SS)

日付範囲指定

- YYYY-MM-DD - 日付 (YYYY-MM-DD)
- >YYYY-MM-DD - 日付 (YYYY-MM-DD) より後
- >=YYYY-MM-DD - 日付 (YYYY-MM-DD) より後
- <YYYY-MM-DD - 日付 (YYYY-MM-DD) より前
- <=YYYY-MM-DD - 日付 (YYYY-MM-DD) より前
- YYYY-MM-DD...YYYY-MM-DD - 日付範囲 (YYYY-MM-DD...YYYY-MM-DD)
- TIMESTAMP - UTC 時間 (TIMESTAMP)
- >TIMESTAMP - UTC 時間 (TIMESTAMP) より後
- >=TIMESTAMP - UTC 時間 (TIMESTAMP) より後
- <TIMESTAMP - UTC 時間 (TIMESTAMP) より前
- <=TIMESTAMP - UTC 時間 (TIMESTAMP) より前
- TIMESTAMP...TIMESTAMP - UTC 時間範囲 (TIMESTAMP...TIMESTAMP)

例

```
// 2016 年 1 月 1 日至 12 月 31 日
$products = wc_get_products( array(
    'downloadable' => true,
    'date_created' => '2016-01-01...2016-12-31',
) );
```



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```
/**
 * 在 WP_Query 对象中添加自定义查询变量 'customvar' 并返回结果
 * @param array $query - WP_Query 对象
 * @param array $query_vars - WC_Product_Query 对象
 * @return array 返回 $query 对象
 */
function handle_custom_query_var( $query, $query_vars ) {
    if ( ! empty( $query_vars['customvar'] ) ) {
        if ( ! isset( $query['meta_query'] ) ) {
            $query['meta_query'] = array();
        }
        $query['meta_query'][] = array(
            'key' => 'customvar',
            'value' => sanitize_text_field( $query_vars['customvar'] ),
        );
    }

    return $query;
}
add_filter( 'woocommerce_product_data_store_cpt_get_products_query',
    'handle_custom_query_var', 10, 2 );
```

1

```
$products = wc_get_products( array( 'customvar' => 'somevalue' ) );
```

1 #2

1 1 2026-03-09 11:53:58 UTC

1 1 2026-03-09 11:53:58 UTC