# Discounts: General Principles (PRO)

# Moodle Shop

The discount principle relies on a general mechanism extensible by class overriding. It is only available in the “Pro” version of the Moodle shop.

A discount class instance will check a particular condition to trigger or not trigger the discount.

If several shop instances are used, the discount policy of each shop is independent.

## Discount scope

Discounts can affect the whole product catalogue, or only a part of the catalogue, as an explicit list of selected products.

## Discount computation, stacking

Discounts are defined in a stack, evaluated in stack order. Each instance is associated with an operator that influences the computation process.

- Cumulative: (default): Accumulates the discount with the previous discounts in the list
- Replace and stop: the discount replaces the previous calculation of the stack. All previous discounts are cancelled. The evaluation of the discount list is stopped.
- Stop if applicable: The discount is applied in cumulative mode, then the evaluation of the list is stopped if the discount was applicable.
- Stop if not applicable: The discount is applied in cumulative mode, then the evaluation of the list is stopped if the discount was NOT applicable.

The final discount is the final result of the stack evaluation, namely **the sum of the individual discounts 'per product line' granted by each discount instance activated** during the computation process.

When taxes apply to product lines, the discount is computed with the same tax as the product line it acts upon. The computation process computes the net and the tax of all the discounts.

## Processing phases

### Session processing

During the cart selection phase, and before the order is saved, discount computation is done from session data. As not all the information driving the discount conditions is always available (non-logged-in unidentified buyer, condition depending contextually on the order, such as a total purchase amount, or a combination of products), the computation is done “as best as possible with the available information”.

It is up to each concrete discount class to define how it computes the discounts applicable and presentable to the customer during this phase.

### Computation at purchase order creation

When the order is saved in the database (during the payment phase), the whole stack is played and the discount results are added to the current order. The discount application processing is divided into 2 phases:

- The applicability check: checks the rules that validate or invalidate the application of the discount.
- The application to the order/invoice.

During computation, the discount context parameters are recorded as production parameters.

### Recomputation when viewing or modifying the invoice / purchase order

Reminder: It is possible to build an invoice through the backoffice, or to modify an order as long as it has not been reconciled (i.e. recorded in the “official” accounting).

Modifications of an invoice or order, in the current version:

- Do not change the result of the application of the discounts initially granted at the time the order was saved. The order will therefore not benefit from new discount clauses configured AFTER it was saved.
- May, however, change the computation result if the discount parameters have changed in the meantime.

## Credits

- Valéry Frémaux (valery@activeprolearn.com) Design - Development
- Florence Labord (florence@activeprolearn.com) Functional review - Tests

Back to the structure - Back to the component summary Local Shop (Boutique) - Back to the catalog

From:
https://docsen.activeprolearn.com/ - **Documentation Moodle ActiveProLearn**

Permanent link:
**https://docsen.activeprolearn.com/doku.php?id=local:shop:structure:discounts**

Last update: **2026/11/13 21:59**