EcomLabs

STEP-BY-STEP GUIDE

Shopify metaobjects: an editable FAQ

The full example for moving the FAQ and the size chart out of your theme code: the metaobject definition, the entries, the product metafield, the Liquid section with its metaobject_list setting, and the checklist.

FOR	Shopify stores with their own or a duplicated theme you can edit
TESTED WITH	Shopify CLI 3.88.1 · shopify theme check on a local scratch theme, no store connected
UPDATED	November 19, 2026
DOCS	shopify.dev/docs/api/liquid/objects/metaobject
ARTICLE	ecomlabs.dev/g/metaobjetos-en
AUTHOR	Luis Morales · @luismoralesdev

Contents

Before you start	What you will build
Step 01	The faq definition
Step 02	Entries and the product metafield
Step 03	The Liquid section
Step 04	The size chart on the product
Step 05	Validate, push and connect
When you finish	Checklist

BEFORE YOU START

What you will build

An FAQ section whose content lives in the admin, not in code. Each question is an entry of a `faq` metaobject. The section decides where the questions come from, in this order:

1. The ones someone picks in the theme editor through a `metaobject_list` setting.
2. With no selection, on a product page, the ones in that product's `custom.faqs` metafield.
3. Failing that, every active entry of type `faq`.

You need permission to edit the store's content and themes. Work on a duplicated or unpublished theme, never directly on the live one. To validate the code on your machine, install Shopify CLI: `shopify theme check` checks Liquid syntax and the schema without connecting to the store.

STEP 01

The faq definition

1. In the admin, go to **Content > Metaobjects** (or **Settings > Custom data**) and click **Add definition**.
2. Name it **FAQ**. Make sure the type ends up as `faq`: Liquid uses that name, and you can edit it before saving.
3. Add the fields with **Add field**:

LABEL	KEY	TYPE	REQUIRED
Question	`question`	Single line text (`single_line_text_field`)	Yes
Answer	`answer`	Rich text (`rich_text_field`)	Yes

4. Under **Field options**, keep **Question** as the display name so the entry list reads on its own.
5. Under **Access**, leave **Storefronts** selected. The section setting does not work if the definition is not available to the storefront.
6. Save.

For the size chart, repeat with the name **Size chart** (type `size_chart`) and three fields: `title` (single line text), `image` (file, images only) and `notes` (multi-line text). Rich text supports headings, lists, links, bold and italics, but not tables, which is why the chart itself is an image.

STEP 02

Entries and the product metafield

1. In **Content > Metaobjects**, open **FAQ** and click **Add entry**.
2. Type the question and the answer. The handle is generated from the display name; if you want to change it, do it now, before code or a setting uses it.
3. Keep the entry **Active** and save. A Draft entry does not show on the storefront.
4. Repeat for each question.

If some questions belong to a specific product, create the product metafield:

1. In **Settings**, open the metafield definitions for **Products** and click **Add definition**.
2. Name: **FAQs**, with namespace and key `custom.faqs`.
3. Type: **Metaobject**, definition **FAQ**, **List of values**. Leave **Storefronts** selected and save.
4. On each product, in the metafields section, click **Select entry** and pick its questions.

For the size chart, create `custom.size_chart` the same way, with **One value** and the **Size chart** definition.

STEP 03

The Liquid section

Save this file as `sections/faq-metaobjects.liquid`. It passed `shopify theme check` with no offenses:

```
{%- liquid
  assign faqs = section.settings.faqs
  if faqs == blank
    assign faqs = product.metafields.custom.faqs.value
  endif
  if faqs == blank
    assign faqs = metaobjects.faq.values
  endif
-%}

{%- if faqs != blank -%}
  <section class="faq">
    {%- if section.settings.heading != blank -%}
      <h2>{{ section.settings.heading | escape }}</h2>
    {%- endif -%}
    {%- for faq in faqs -%}
      <details class="faq__item">
        <summary>{{ faq.question.value | escape }}</summary>
        {{ faq.answer | metafield_tag }}
      </details>
    {%- endfor -%}
  </section>
{%- endif -%}

{% schema %}
{
  "name": "FAQ (metaobjects)",
  "settings": [
    {
      "type": "text",
      "id": "heading",
      "label": "Heading",
      "default": "Frequently asked questions"
    },
    {
      "type": "metaobject_list",
      "id": "faqs",
```

```
      "label": "Questions",
      "metaobject_type": "faq",
      "limit": 20
    }
  ],
  "presets": [{ "name": "FAQ (metaobjects)" }]
}
{% endschema %}
```

- `.value` reads a field's value; `metafield_tag` takes the whole field and renders the rich text as HTML.
- `metaobject_type` must match the definition's type. `limit` goes up to 50.
- `<details>` and `<summary>` open and close each answer from the keyboard, with no JavaScript.
- If your theme translates editor labels, swap `"Heading"` and `"Questions"` for keys from your locale files.

STEP 04

The size chart on the product

Save this snippet as `snippets/size-chart.liquid` and call it from the product section with `{% render 'size-chart', product: product %}`:

```
{% doc %}
  Renders the size chart linked to a product.
  @param {product} product - The product whose custom.size_chart metafield is read
  @example
  {% render 'size-chart', product: product %}
{% enddoc %}
{%- assign chart = product.metafields.custom.size_chart.value -%}
{%- if chart != blank -%}
  <div class="size-chart">
    <h3>{{ chart.title.value | escape }}</h3>
    {{ chart.image | metafield_tag }}
    {{ chart.notes | metafield_tag }}
  </div>
{%- endif -%}
```

One size chart serves many products: you create it once and each product references it. For the image, `metafield_tag` renders an `<img>` with `loading="lazy"` and the file's alt text, so write that alt text under **Content > Files**.

STEP 05

Validate, push and connect

1. In the theme folder: `shopify theme check --path .`. It must finish with no offenses.
2. Push to a new unpublished theme: `shopify theme push --unpublished`. The live theme is not touched.
3. Open that theme in the editor, add the **FAQ (metaobjects)** section and pick entries under **Questions**.

4. Check a page with no selection and a product with `custom.faqs` filled in: the section should take questions from each source.
5. Publish the theme only after you have reviewed it.

> **CAUTION**
>
> With the *publishable* capability on, a Draft entry returns `nil` and loops skip it. A `for` loop covers at most 50 entries; for more, use `{% paginate metaobjects.faq.values by 100 %}`, which allows up to 250 per page. Renaming the handle of an entry in use breaks manually set references in Liquid. Deleting a definition deletes all of its entries and cannot be undone.

WHEN YOU FINISH

Checklist

- [ ] The `faq` definition has the keys `question` and `answer`, and its type is `faq`.
- [ ] **Storefronts** stays selected on the definition and on the product metafield.
- [ ] Entries are Active and their handles do not change once in use.
- [ ] `custom.faqs` is a list of references to `faq`; `custom.size_chart` is a single reference.
- [ ] The code uses `metaobjects.faq`, not `shop.metaobjects.faq`.
- [ ] `.value` to read, `metafield_tag` on the field.
- [ ] The setting's `limit` is 50 or less; more than 50 entries go through `paginate`.
- [ ] `shopify theme check` passes and the theme was tested unpublished.