

shopify liquid snippets pack

60 developer-ready code snippets for shopify themes

Version: 2.0.0

Compatibility: Shopify OS 2.0+ and Legacy themes

License: Single-store commercial use

what's included

This pack contains 60 production-ready Liquid snippets covering the full spectrum of Shopify theme development needs — from product badges and urgency messaging to SEO schema, mega menus, infinite scroll, and cookie consent. Each snippet is self-contained, well-commented, and ready to drop into any theme.

quick installation

os 2.0 themes (dawn, refresh, crave, etc.)

1. In your Shopify admin, go to **Online Store > Themes**
2. Click **Actions > Edit code** on your active theme
3. Under the **Snippets** folder, click **Add a new snippet**
4. Name it exactly as the `.liquid` file (without the `.liquid` extension)
5. Paste the snippet content and click **Save**
6. Include the snippet in any template using `{% render 'snippet-name' %}`

legacy themes (debut, brooklyn, minimal, etc.)

Same process as above. Legacy themes use `{% include 'snippet-name' %}` instead of `{% render %}`. Both tags work in most themes, but `{% render %}` is recommended as `{% include %}` is deprecated.

snippet reference

1. product-badge

File: `product-badge.liquid`

Dynamic product badges rendered as overlays on product images. Displays SALE, NEW, SOLD OUT, BEST SELLER, and LOW STOCK badges based on product data.

Usage:

```
{% render 'product-badge', product: product %}
```

Parameters:

- `product` — the product object (required)
- `low_stock_threshold` — integer, default 5

Notes: Add CSS from the header comment to your theme stylesheet or paste into `base.css`.

2. price-with-savings

File: `price-with-savings.liquid`

Displays original price struck through, current sale price, and a "Save X%" or "Save \$X" pill badge.

Usage:

```
{% render 'price-with-savings', product: product, variant: product.selected_or_first_available_variant %}
```

Parameters:

- `product` — product object (required)
- `variant` — variant object (required)
- `show_savings_percent` — boolean, default true

3. inventory-urgency

File: `inventory-urgency.liquid`

Shows low-stock urgency messaging ("Only 3 left!") when inventory is below a configurable threshold. Hidden when out of stock or stock is high.

Usage:

```
{% render 'inventory-urgency', variant: product.selected_or_first_available_variant %}
```

Parameters:

- `variant` — variant object (required)
- `threshold` — integer, default 10

4. sticky-atc-bar

File: `sticky-atc-bar.liquid`

A sticky add-to-cart bar that slides in from the bottom after the user scrolls past the main ATC button. Includes product image, title, price, and a variant selector.

Usage: Place at the bottom of `product.json` or `product-template.liquid`, outside the main product section.

```
{% render 'sticky-atc-bar', product: product %}
```

Parameters:

- `product` — product object (required)
- `scroll_offset` — pixels before bar appears, default 400

Notes: Requires the main ATC button to have `id="MainProductForm-ATC"` or update the JS selector.

5. product-tabs

File: `product-tabs.liquid`

Tabbed content interface for product pages: Description, Shipping Info, Size Guide, and Reviews. Content is pulled from metafields where available.

Usage:

```
{% render 'product-tabs', product: product %}
```

Parameters:

- `product` — product object (required)
- `show_reviews_tab` — boolean, default true

Metafields used:

- `product.metafields.custom.shipping_info` — shipping tab content (HTML)
- `product.metafields.custom.size_guide` — size guide tab content (HTML)

6. trust-badges

File: `trust-badges.liquid`

A horizontal row of trust icons: Free Shipping, Secure Checkout, Easy Returns, and Money-Back Guarantee. Inline SVG icons, fully configurable text.

Usage:

```
{% render 'trust-badges', free_shipping_threshold: 50 %}
```

Parameters:

- `free_shipping_threshold` — number, shows threshold in message, default 0 (always free)
- `show_returns` — boolean, default true
- `show_guarantee` — boolean, default true

7. recently-viewed

File: `recently-viewed.liquid`

Renders a "Recently Viewed" product grid. Liquid outputs the section shell and loads product data from `localStorage` via JavaScript.

Usage: Place on product pages inside a section or at the bottom of `product.liquid`.

```
{% render 'recently-viewed', limit: 4 %}
```

Parameters:

- `limit` — max products to show, default 4

Notes: Requires the product handle and image to be stored in localStorage on each product page view. The snippet includes the storage script.

8. announcement-bar

File: `announcement-bar.liquid`

A dismissible top announcement bar with message text, optional CTA button, and close button. Dismissed state stored in `sessionStorage`.

Usage: Place in `theme.liquid` directly after the opening `<body>` tag.

```
{% render 'announcement-bar',
  message: 'Free shipping on all orders over $50',
  cta_text: 'Shop Now',
  cta_url: '/collections/all'
%}
```

Parameters:

- `message` — string (required)
- `cta_text` — string, optional
- `cta_url` — string, optional
- `bg_color` — hex color, default `#1a1a1a`
- `text_color` — hex color, default `#ffffff`
- `bar_id` — unique ID for sessionStorage key, default `announcement-bar`

9. back-in-stock-form

File: `back-in-stock-form.liquid`

An email capture form that replaces the ATC button on sold-out products. Posts to Klaviyo Back In Stock endpoint or a configurable custom URL.

Usage: Place conditionally in your product form.

```
{% unless product.available %}
  {% render 'back-in-stock-form', product: product, variant: product.selected_or_first_available_variant %}
{% endunless %}
```

Parameters:

- `product` — product object (required)
- `variant` — variant object (required)
- `endpoint` — form action URL, default is Klaviyo endpoint
- `klaviyo_company_id` — your Klaviyo public API key

10. quantity-breaks

File: `quantity-breaks.liquid`

Displays a tiered pricing table: Buy 1, Buy 3 (save X%), Buy 5+ (save Y%). Pricing tiers are defined via metafields.

Usage:

```
{% render 'quantity-breaks', product: product, variant: product.selected_or_first_available_variant %}
```

Parameters:

- `product` — product object (required)
- `variant` — variant object (required)

Metafields used:

- `product.metafields.custom.quantity_breaks` — JSON array of `{qty, price}` objects

Metafield example:

```
[  
  {"qty": 1, "label": "1 item", "price": 2999},  
  {"qty": 3, "label": "3 items", "price": 2499},  
  {"qty": 5, "label": "5+ items", "price": 1999}  
]
```

11. before-after-slider

File: `before-after-slider.liquid`

Image comparison slider with a draggable divider. Used for product result images (skincare, cleaning products, etc.). No external dependencies.

Usage:

```
{% render 'before-after-slider',  
  before_image: product.metafields.custom.before_image,  
  after_image: product.metafields.custom.after_image,  
  before_label: 'Before',  
  after_label: 'After'  
%}
```

Parameters:

- `before_image` — image object or URL (required)
- `after_image` — image object or URL (required)
- `before_label` — string, default `Before`
- `after_label` — string, default `After`
- `initial_position` — 0–100, default 50

12. breadcrumbs

File: `breadcrumbs.liquid`

SEO-friendly breadcrumb navigation with JSON-LD structured data. Handles product, collection, page, blog, article, and search result contexts automatically.

Usage: Place near the top of any template.

```
{% render 'breadcrumbs' %}
```

No parameters required — reads from Shopify's `request` and `page_*` objects automatically.

13. shipping-countdown

File: `shipping-countdown.liquid`

"Order in Xh Ym for delivery by [date]" countdown banner. Configurable daily cutoff time. Inline JS updates the countdown every second.

Usage:

```
{% render 'shipping-countdown',  
  cutoff_hour: 15,  
  cutoff_minute: 0,  
  delivery_days: 3  
%}
```

Parameters:

- `cutoff_hour` — 24h hour, default 15 (3pm)
- `cutoff_minute` — minutes, default 0
- `delivery_days` — business days after cutoff, default 3
- `timezone_offset` — UTC offset in hours for store timezone, default 0

14. variant-image-swatch

File: `variant-image-swatch.liquid`

Color swatch selector that updates the main product image on hover (preview) and click (select). Works with standard Shopify variant images.

Usage: Replace or supplement your theme's default variant selector.

```
{% render 'variant-image-swatch', product: product %}
```

Parameters:

- `product` — product object (required)
- `option_name` — name of the color option, default `Color`
- `show_tooltip` — boolean, default true

15. product-rating-stars

File: `product-rating-stars.liquid`

Renders a star rating display (1–5 stars) using the product's `reviews.rating` metafield. Supports half-stars. Inline SVG, no image assets needed.

Usage:

```
{% render 'product-rating-stars', product: product %}
```

Parameters:

- `product` — product object (required)
- `show_count` — boolean, shows review count, default true
- `size` — `sm`, `md`, `lg`, default `md`

Metafields used:

- `product.metafields.reviews.rating` — decimal rating value
- `product.metafields.reviews.rating_count` — integer review count

16. cart-upsell

File: `cart-upsell.liquid`

Displays a single upsell product recommendation in the cart drawer or cart page. Matches cart item tags to find a relevant upsell product.

Usage:

```
{% render 'cart-upsell', cart: cart %}
```

Parameters:

- `cart` — cart object (required)
- `upsell_collection_handle` — handle of the upsell products collection, default `upsells`
- `heading` — string, default `You might also like`

17. collection-filter-sort

File: `collection-filter-sort.liquid`

Client-side filter and sort UI for collection pages. Filters by tag groups, sorts by price/date/title. Updates URL query params for shareable filtered URLs.

Usage: Place at the top of your collection template, before the product grid.

```
{% render 'collection-filter-sort', collection: collection %}
```

Parameters:

- `collection` — collection object (required)
- `filter_groups` — comma-separated list of tag prefixes to expose as filter groups (e.g., `color,size,material`)
- `show_sort` — boolean, default true

18. free-shipping-progress

File: `free-shipping-progress.liquid`

An animated progress bar in the cart showing how close the customer is to the free shipping threshold. Updates dynamically via JS as items are added.

Usage: Place inside your cart drawer or cart page template.

```
{% render 'free-shipping-progress', cart: cart, threshold: 75 %}
```

Parameters:

- `cart` — cart object (required)
- `threshold` — free shipping threshold in dollars, default 75
- `currency_symbol` — string, default `$`
- `success_message` — string shown when threshold reached

19. social-proof-popup

File: `social-proof-popup.liquid`

"Someone in [City] just bought [Product]" notification popup. Rotates through a list of recent purchase data from a metafield or hardcoded fallback array. Auto-dismisses after configurable delay.

Usage: Place before the closing `</body>` tag in `theme.liquid`.

```
{% render 'social-proof-popup' %}
```

Parameters:

- `delay_first` — ms before first popup, default 4000
- `delay_between` — ms between popups, default 8000
- `display_duration` — ms each popup is visible, default 4000
- `position` — `bottom-left` or `bottom-right`, default `bottom-left`

Metafields used:

- `shop.metafields.custom.recent_purchases` — JSON array of purchase objects

20. metafield-size-guide

File: `metafield-size-guide.liquid`

A modal size guide triggered by a "Size Guide" link. Content pulled from `product.metafields.custom.size_guide` (HTML). Falls back to a collection-level guide.

Usage:

```
{% render 'metafield-size-guide', product: product %}
```

Parameters:

- `product` — product object (required)
- `link_text` — string, default `Size Guide`
- `fallback_collection` — collection handle to use as fallback source

21. product-schema

File: `product-schema.liquid`

JSON-LD Product structured data with offers, aggregateRating, brand, and availability. Place inside `<head>` on product pages for Google rich results eligibility.

Usage:

```
{% render 'product-schema', product: product %}
```

Parameters:

- `product` — product object (required)
- `variant` — variant object, defaults to first available
- `brand_name` — string, falls back to shop name

Metafields used: `product.metafields.reviews.rating`, `product.metafields.reviews.rating_count`

22. faq-schema

File: `faq-schema.liquid`

JSON-LD FAQPage structured data and a visible accessible accordion. Questions and answers are loaded from a product or page metafield JSON array.

Usage:

```
{% render 'faq-schema', product: product %}  
{% render 'faq-schema', page: page %}
```

Parameters:

- `product` / `page` — source object (one required)
- `heading` — section heading, default `Frequently Asked Questions`
- `open_first` — boolean, default true

Metafields used: `product.metafields.custom.faq_items` (JSON array)

23. organization-schema

File: `organization-schema.liquid`

JSON-LD Organization block with logo, social sameAs links, and contact point. Place once in `theme.liquid <head>`.

Usage:

```
{% render 'organization-schema',  
  twitter_url: 'https://twitter.com/mybrand',  
  facebook_url: 'https://facebook.com/mybrand'  
%}
```

Parameters:

- `org_name`, `org_url`, `logo_url` — organizational identity
- `twitter_url`, `facebook_url`, `instagram_url`, `pinterest_url`, `youtube_url`, `tiktok_url` — social profile URLs
- `contact_phone`, `contact_email` — customer service contact

24. collection-schema

File: `collection-schema.liquid`

JSON-LD ItemList structured data for collection pages. Lists products with their URLs, names, images, and prices.

Usage:

```
{% paginate collection.products by 24 %}
  {% render 'collection-schema', collection: collection, limit: 24, current_page: current_page %}
{% endpaginate %}
```

Parameters:

- `collection` — collection object (required)
- `limit` — products per page, default 24
- `current_page` — pagination page number, default 1

25. canonical-tags

File: `canonical-tags.liquid`

Smart `<link rel="canonical">` handler. Strips sort/filter URL params from collection pages, strips `?variant=` from product URLs, preserves pagination, and handles all Shopify template types.

Usage: Place in `theme.liquid <head>`, replacing any existing canonical tag.

```
{% render 'canonical-tags' %}
```

No parameters required.

navigation & ux

26. mega-menu

File: `mega-menu.liquid`

Mega menu dropdown navigation with optional featured image and child link columns. Hover-triggered on desktop, click-triggered on touch. Keyboard accessible.

Usage:

```
{% render 'mega-menu', menu: linklists.main-menu %}
```

Parameters:

- `menu` — linklist object (required)
- `show_images` — boolean, default true

- `image_metafield` — metafield key for the panel image

27. mobile-drawer-menu

File: `mobile-drawer-menu.liquid`

Slide-in mobile navigation drawer with nested accordion sub-menus, overlay backdrop, and focus trap. Closes on Escape key.

Usage:

```
{% render 'mobile-drawer-menu', menu: linklists.main-menu %}
```

Parameters:

- `menu` — linklist object (required)
- `heading` — drawer heading, default `Menu`
- `show_close_btn` — boolean, default true
- `trigger_id` — ID of an external trigger button (optional)

28. back-to-top

File: `back-to-top.liquid`

Floating back-to-top button. Appears after scrolling 300px, fades out when near top. Smooth scrolls to top on click.

Usage: Place before `</body>` in `theme.liquid`.

```
{% render 'back-to-top' %}
```

Parameters:

- `scroll_threshold` — px before button appears, default 300
- `position` — `bottom-right` or `bottom-left`, default `bottom-right`
- `offset_bottom`, `offset_side` — positioning in px, default 24

29. search-autocomplete

File: `search-autocomplete.liquid`

Predictive search input using Shopify's `/search/suggest.json` API. Shows live product results with images, titles, and prices. Keyboard navigable with arrow keys.

Usage:

```
{% render 'search-autocomplete', placeholder: 'Search products...', results_limit: 5 %}
```

Parameters:

- `input_id` — ID for the input element
- `placeholder` — input placeholder text
- `results_limit` — number of results, default 5
- `show_images` — boolean, default true

- `debounce_ms` — debounce delay in ms, default 200

30. reading-progress-bar

File: `reading-progress-bar.liquid`

Thin progress bar at the top of the viewport that fills as the user reads a blog article. Only renders on article pages.

Usage: Place in `<head>` or at the top of your article template.

```
{% render 'reading-progress-bar', color: '#2563eb', height: 3 %}
```

Parameters:

- `color` — bar color, default `#111111`
- `height` — bar height in px, default 3
- `z_index` — CSS z-index, default 999
- `content_selector` — CSS selector for the article content element, default `article`

product page enhancements

31. product-video

File: `product-video.liquid`

Embeds a product video from a metafield URL. Auto-detects YouTube, Vimeo, or native `.mp4` and renders the appropriate embed. Lazy-loaded via IntersectionObserver.

Usage:

```
{% render 'product-video', product: product %}
```

Parameters:

- `product` — product object (required)
- `video_url` — override URL (instead of metafield)
- `autoplay` — boolean, default false
- `aspect_ratio` — `landscape` (16:9) or `portrait` (9:16), default `landscape`

Metafields used: `product.metafields.custom.video_url`

32. complementary-products

File: `complementary-products.liquid`

"Complete the look" section using Shopify's Product Recommendations API. Fetches related or complementary products via AJAX and renders a horizontal scroll strip.

Usage:

```
{% render 'complementary-products', product: product %}
```

Parameters:

- `product` — product object (required)
- `heading` — section heading, default `Complete the look`
- `limit` — number of products, default 4
- `intent` — `related` or `complementary`, default `complementary`
- `show_atc` — show Add to Cart buttons, default true

33. product-sticky-sidebar

File: `product-sticky-sidebar.liquid`

CSS-only sticky sidebar for the product form column on large screens. Reverts to normal flow on mobile. No JavaScript needed.

Usage: Apply the generated class to your product form column:

```
{% render 'product-sticky-sidebar', product: product %}
```

Parameters:

- `top_offset` — px from top when stuck, default 80
- `breakpoint` — min viewport width to enable sticky, default 1024

34. size-recommendation

File: `size-recommendation.liquid`

"What's my size?" 3-step quiz (gender, height, build) that outputs a size recommendation. Opens in a modal. Fully client-side.

Usage: Place near your size selector:

```
{% render 'size-recommendation', product: product %}
```

Parameters:

- `product` — product object (required)
- `trigger_text` — link text, default `What's my size?`
- `unit_system` — `imperial` or `metric`, default `imperial`

35. product-waitlist

File: `product-waitlist.liquid`

Email waitlist form for out-of-stock products. Replaces the ATC button when unavailable. Supports Klaviyo back-in-stock or Shopify contact form.

Usage:

```
{% unless product.available %}
  {% render 'product-waitlist', product: product, variant: product.selected_or_first_available_variant %}
{% endunless %}
```

Parameters:

- `product`, `variant` — product/variant objects (required)
- `heading` — form heading, default `Join the Waitlist`
- `endpoint` — POST URL, default `/contact`
- `klaviyo_list_id` — Klaviyo list ID for direct subscribe

36. sold-count

File: `sold-count.liquid`

"X sold in the last 24 hours" urgency message. Reads from a metafield if set; otherwise generates a deterministic random number seeded by product ID and current day.

Usage:

```
{% render 'sold-count', product: product, min_count: 8, max_count: 47 %}
```

Parameters:

- `product` — product object (required)
- `min_count`, `max_count` — random range, defaults 8–47
- `label_singular`, `label_plural` — message labels
- `icon` — show fire icon, default true

Metafields used: `product.metafields.custom.sold_today`

37. product-personalization

File: `product-personalization.liquid`

Custom text / engraving / monogram input that stores as a line item property. Includes live character counter. Works with Shopify's native cart form.

Usage: Inside your product form, before the ATC button:

```
{% render 'product-personalization',  
  product: product,  
  label: 'Engraving Text',  
  max_length: 20  
%}
```

Parameters:

- `product` — product object (required)
- `label` — input label, default `Personalisation`
- `property_name` — line item property key, default `Custom Text`
- `max_length` — character limit, default 30
- `required` — boolean, default false

38. gift-wrap-option

File: `gift-wrap-option.liquid`

Gift wrap checkbox and message textarea inside the product form. Stores as line item properties. Optionally adds a "Gift Wrap" variant to the cart for upcharge.

Usage: Inside your product form, before the ATC button:

```
{% render 'gift-wrap-option', product: product, label: 'Add gift wrapping (+$4.99)' %}
```

Parameters:

- `product` — product object (required)
- `label` — checkbox label
- `price` — upcharge in cents, default 0 (free)
- `gift_wrap_variant_id` — variant ID to add to cart for upcharge
- `max_message_length` — gift message character limit, default 200

cart enhancements

39. cart-note

File: `cart-note.liquid`

Expandable order notes section. Toggle button reveals a textarea. Submits as `cart.note` via the cart form or AJAX (for cart drawers).

Usage:

```
{% render 'cart-note', cart: cart %}
```

Parameters:

- `cart` — cart object (required)
- `label` — toggle label, default `Add order note`
- `max_length` — character limit, default 500
- `open_by_default` — auto-open if `cart.note` exists, default true

40. cart-item-properties

File: `cart-item-properties.liquid`

Renders line item properties (personalisation text, gift messages, etc.) in a clean format inside the cart. Skips properties with underscore-prefixed names.

Usage: Inside your cart item loop:

```
{% for item in cart.items %}
  {% render 'cart-item-properties', item: item %}
{% endfor %}
```

Parameters:

- `item` — cart line item object (required)
- `exclude_empty` — skip blank properties, default true

- `style` — `list` or `inline`, default `list`
- `separator` — key/value separator, default `:`

41. express-checkout-buttons

File: `express-checkout-buttons.liquid`

Isolated wrapper for Shopify's dynamic checkout buttons (Shop Pay, Apple Pay, Google Pay). Includes a styled "or" divider. Must be placed inside a `{% form 'product', product %}` block.

Usage: After your ATC button inside a product form:

```
{% render 'express-checkout-buttons', product: product, show_divider: true %}
{{ form | payment_button }}
```

Parameters:

- `product` — product object
- `show_divider` — show OR divider, default `true`
- `divider_text` — divider label, default `or`

42. minimum-order-warning

File: `minimum-order-warning.liquid`

Warning banner when cart total is below a minimum order value. Updates dynamically via `cart:updated` / `cart:add` custom events.

Usage:

```
{% render 'minimum-order-warning', cart: cart, minimum: 50 %}
```

Parameters:

- `cart` — cart object (required)
- `minimum` — minimum order value in dollars (required)
- `currency_symbol` — default `$`
- `message` — custom message with `{remaining}` and `{minimum}` placeholders
- `hide_checkout` — disable checkout button when below minimum, default `false`

43. cart-recommendations

File: `cart-recommendations.liquid`

"You might also like" section in the cart. Fetches recommendations for the first cart item via Shopify's API and renders a horizontal scroll strip.

Usage:

```
{% render 'cart-recommendations', cart: cart %}
```

Parameters:

- `cart` — cart object (required)
- `heading` — section heading, default `You might also like`
- `limit` — number of products, default 4
- `intent` — `related` or `complementary`, default `related`

collection page

44. infinite-scroll

File: `infinite-scroll.liquid`

Replaces pagination with infinite scroll. Uses IntersectionObserver on the last product card. Preserves browser back-button history via `History.pushState`.

Usage:

```
{% paginate collection.products by 24 %}
<div data-product-grid>
  {% for product in collection.products %}
    <div data-product-card>...</div>
  {% endfor %}
</div>
{% render 'infinite-scroll', paginate: paginate, collection: collection %}
{% endpaginate %}
```

Parameters:

- `paginate` — paginate object (required)
- `collection` — collection object (required)
- `product_grid_selector` — CSS selector for the grid container, default `[data-product-grid]`
- `product_card_selector` — CSS selector for product cards, default `[data-product-card]`

45. collection-hero-banner

File: `collection-hero-banner.liquid`

Collection header with background image, title, description, and optional CTA. Image sourced from collection or metafield. Supports light/dark text overlay.

Usage:

```
{% render 'collection-hero-banner', collection: collection, text_color: 'light' %}
```

Parameters:

- `collection` — collection object (required)
- `show_title`, `show_description` — booleans, default true
- `text_color` — `light` or `dark`, default `light`
- `min_height` — minimum banner height in px, default 280
- `overlay_opacity` — 0–100, default 40
- `cta_text`, `cta_url` — optional CTA button

Metafields used: `collection.metafields.custom.hero_image`, `collection.metafields.custom.hero_subtitle`

46. active-filters-display

File: `active-filters-display.liquid`

Renders active filter chips with individual remove buttons and a "Clear all" link. Requires Shopify's native product filtering to be enabled.

Usage:

```
{% render 'active-filters-display', results: collection %}
```

Parameters:

- `results` — collection or search object with `.filters` (required)
- `clear_text` — "Clear all" label, default `Clear all`

47. product-quick-view

File: `product-quick-view.liquid`

Modal quick view triggered by a "Quick View" button on product cards. Fetches product JSON via AJAX, renders image, variants, price, and ATC button without leaving the collection page.

Usage:

```
{# In product-card.liquid: #}
{% render 'product-quick-view', product: product, trigger_only: true %}

{# In collection.liquid (once): #}
{% render 'product-quick-view' %}
```

Parameters:

- `product` — product object (for trigger button)
- `trigger_only` — boolean, render trigger only vs modal shell only
- `trigger_text` — button label, default `Quick View`

48. collection-compare

File: `collection-compare.liquid`

Side-by-side product comparison for up to 3 products. Checkboxes on cards add products to a sticky comparison tray. "Compare" button opens a modal table.

Usage:

```
{# In product-card.liquid: #}
{% render 'collection-compare', product: product, trigger_only: true %}

{# In collection.liquid (once): #}
{% render 'collection-compare', collection: collection %}
```

Parameters:

- `product` / `collection` — context objects
- `trigger_only` — boolean
- `max_items` — maximum comparable products, default 3
- `compare_text` — Compare button label

homepage sections

49. featured-collection-tabs

File: `featured-collection-tabs.liquid`

Tabbed collection display for the homepage. Each tab shows products from a different collection. All content is server-rendered; tabs switch via JS only.

Usage:

```
{% render 'featured-collection-tabs',
  tab_collections: 'new-arrivals,best-sellers,sale',
  tab_labels: 'New Arrivals,Best Sellers,Sale',
  products_per_tab: 4
%}
```

Parameters:

- `tab_collections` — comma-separated collection handles (required)
- `tab_labels` — comma-separated tab labels (optional, falls back to collection titles)
- `products_per_tab` — default 4
- `show_view_all` — show "View all" link, default true
- `heading` — section heading

50. testimonials-carousel

File: `testimonials-carousel.liquid`

Auto-playing testimonials slider with dot indicators and prev/next buttons. Auto-play pauses on hover. Falls back to demo testimonials if no metafield is set.

Usage:

```
{% render 'testimonials-carousel', heading: 'Customer Reviews', autoplay_ms: 5000 %}
```

Parameters:

- `heading` — section heading
- `autoplay_ms` — slide interval in ms, default 4000
- `show_arrows`, `show_dots` — boolean controls, default true

Metafields used: `shop.metafields.custom.testimonials` (JSON array)

51. logo-bar

File: `logo-bar.liquid`

Horizontally scrolling marquee of brand or press logos. Duplicates the list for a seamless infinite loop. Pauses on hover. Optional greyscale filter with colour on hover.

Usage:

```
{% render 'logo-bar', heading: 'As seen in', speed: 25 %}
```

Parameters:

- `heading` — label above logos
- `speed` — scroll duration in seconds, default 30
- `logo_height` — image height in px, default 40
- `greyscale` — boolean, default true
- `gap` — gap between logos in px, default 48

Metafields used: `shop.metafields.custom.logo_bar_images` (file_reference list)

52. stats-counter

File: `stats-counter.liquid`

Animated number counters (e.g. "10,000+ customers") that count up from 0 when the section scrolls into view. Supports 3 or 4 stat columns.

Usage:

```
{% render 'stats-counter',  
  stat_1_num: '25,000+', stat_1_label: 'Happy Customers',  
  stat_2_num: '98%',   stat_2_label: 'Satisfaction Rate',  
  stat_3_num: '50+',   stat_3_label: 'Products'  
%}
```

Parameters:

- `stat_1_num` through `stat_4_num` — number strings (suffixes like +, %, ★ are preserved)
- `stat_1_label` through `stat_4_label` — label text below each number
- `heading` — optional section heading

53. video-hero

File: `video-hero.liquid`

Full-width hero section with autoplaying, muted, looping background video. Overlays headline, subheading, and CTA buttons. Falls back to poster image on unsupported browsers.

Usage:

```
{% render 'video-hero',
  video_url: 'https://cdn.shopify.com/.../hero.mp4',
  poster_url: 'https://cdn.shopify.com/.../poster.jpg',
  heading: 'Summer Collection 2025',
  cta_text: 'Shop Now',
  cta_url: '/collections/summer'
%}
```

Parameters:

- `video_url` — .mp4 URL (required, or set via metafield)
- `poster_url` — fallback image
- `heading`, `subheading` — hero text
- `cta_text`, `cta_url` — primary CTA
- `cta_secondary_text`, `cta_secondary_url` — secondary link
- `overlay_opacity` — 0–100, default 45
- `min_height` — section min-height in px, default 560
- `text_align` — `left`, `center`, or `right`

performance & technical

54. image-lazy-load

File: `image-lazy-load.liquid`

Image component with native lazy loading, low-quality placeholder (LQIP) blur-up effect, and responsive `srcset`. Full image fades in after loading.

Usage:

```
{% render 'image-lazy-load',
  image: product.featured_image,
  widths: '360,720,1080',
  sizes: '(max-width: 600px) 100vw, 50vw',
  alt: product.title
%}
```

Parameters:

- `image` — Shopify image object or URL (required)
- `widths` — comma-separated srcset widths
- `sizes` — sizes attribute
- `alt` — alt text
- `aspect_ratio` — CSS aspect-ratio value e.g. `4/3`
- `loading` — `lazy` or `eager`, default `lazy`
- `lqip_width` — placeholder width in px, default 20

55. critical-css-inline

File: `critical-css-inline.liquid`

Template for inlining critical above-fold CSS and loading the full stylesheet deferred using the `media="print"` preload technique. Improves LCP and FCP scores.

Usage: In `theme.liquid <head>`, replacing your main stylesheet link:

```
{% render 'critical-css-inline',
  critical_asset: 'critical.css',
  full_stylesheet: 'base.css'
%}
```

Parameters:

- `critical_asset` — critical CSS filename in `/assets/`
- `full_stylesheet` — main CSS filename, default `base.css`
- `full_stylesheet_2` — optional second CSS file
- `noscript_fallback` — add noscript fallback, default true

56. external-scripts-defer

File: `external-scripts-defer.liquid`

Loads third-party scripts (analytics, chat widgets, pixels) after the page loads. Supports `idle` (`requestIdleCallback`), `load`, and `delay` strategies to minimise render-blocking impact.

Usage: Place before `</body>` in `theme.liquid`, then configure the SCRIPTS array:

```
{% render 'external-scripts-defer', strategy: 'idle' %}
```

Parameters:

- `strategy` — `idle`, `load`, or `delay`, default `idle`
- `delay_ms` — delay in ms when strategy = `delay`, default 3000

Notes: Edit the `SCRIPTS` array inside the snippet's `<script>` block to add your third-party URLs.

57. cookie-consent

File: `cookie-consent.liquid`

GDPR cookie consent banner with Accept and Decline buttons. Dismissed state persists in localStorage. Dispatches `cookies:accepted` / `cookies:declined` custom events for conditional script loading.

Usage: Place before `</body>` in `theme.liquid`.

```
{% render 'cookie-consent',
  message: 'We use cookies to improve your experience.',
  accept_text: 'Accept All',
  decline_text: 'Decline'
%}
```

Parameters:

- `message` — banner message text
- `accept_text`, `decline_text` — button labels

- `policy_text`, `policy_url` — privacy policy link
- `position` — `bottom` or `top`, default `bottom`
- `expiry_days` — days before re-showing, default 365

58. preload-fonts

File: `preload-fonts.liquid`

Outputs `<link rel="preload">` tags for custom font files and `@font-face` declarations with `font-display: swap`. Eliminates render blocking font requests and reduces layout shift.

Usage: Place in `theme.liquid <head>` before stylesheet links.

```
{% render 'preload-fonts',
  font_1_asset: 'my-font-regular.woff2',
  font_1_family: 'MyFont',
  font_1_weight: '400',
  font_2_asset: 'my-font-bold.woff2',
  font_2_weight: '700'
%}
```

Parameters:

- `font_1_asset`, `font_1_family`, `font_1_weight`, `font_1_style` — primary font
- `font_2_asset`, `font_2_family`, `font_2_weight`, `font_2_style` — secondary font
- `preconnect_google` — add Google Fonts preconnect headers, default false

email & social

59. social-share-buttons

File: `social-share-buttons.liquid`

Share buttons for Twitter/X, Facebook, Pinterest, and WhatsApp. Uses native Web Share API on mobile when available. Works for products, articles, or any URL.

Usage:

```
{% render 'social-share-buttons', product: product %}
{% render 'social-share-buttons', article: article %}
```

Parameters:

- `product` / `article` — source object (auto-fills url/title/image)
- `url`, `title`, `image_url` — manual overrides
- `show_twitter`, `show_facebook`, `show_pinterest`, `show_whatsapp`, `show_native` — boolean toggles
- `label` — label above buttons, default `Share`
- `icon_size` — icon size in px, default 20

60. newsletter-popup

File: `newsletter-popup.liquid`

Time-delayed or exit-intent email capture popup. `sessionStorage` prevents re-showing in the same session; `localStorage` controls the longer dismiss window. Supports Klaviyo or Shopify contact form submission.

Usage: Place before `</body>` in `theme.liquid`.

```
{% render 'newsletter-popup',
  heading: 'Get 10% Off Your First Order',
  delay_seconds: 6,
  klaviyo_list_id: 'ABC123'
%}
```

Parameters:

- `heading`, `subheading`, `offer_text` — popup copy
- `button_text` — submit label, default `Subscribe`
- `delay_seconds` — seconds before showing, default 5 (0 = exit-intent only)
- `exit_intent` — trigger on cursor leaving viewport, default true
- `dismiss_days` — days before re-showing, default 7
- `klaviyo_list_id` — Klaviyo list ID
- `bg_image_url` — left-panel background image

61. instagram-feed-placeholder

File: `instagram-feed-placeholder.liquid`

Static Instagram-style feed grid using images from a metafield image list. GDPR-safe: no API calls or third-party embeds. Includes profile header with username and follower count.

Usage:

```
{% render 'instagram-feed-placeholder',
  heading: 'Follow us on Instagram',
  instagram_handle: 'mystore',
  columns: 6,
  image_limit: 6
%}
```

Parameters:

- `heading` — section heading
- `instagram_handle` — Instagram username (no @)
- `profile_url` — full profile URL
- `columns` — grid columns, default 6
- `image_limit` — max images to display, default 6
- `show_profile` — show profile header, default true
- `cta_text` — profile link label, default `View Profile`

Metafields used: `shop.metafields.custom.instagram_handle`, `shop.metafields.custom.instagram_images`, `shop.metafields.custom.instagram_followers`

general notes

css

Most snippets include a `<style>` block with scoped CSS. For production, extract these styles and add them to your theme's main stylesheet (`base.css` or `theme.scss`) to avoid render-blocking inline styles.

javascript

All JavaScript in these snippets is vanilla JS — no jQuery or framework dependencies. Scripts use `DOMContentLoaded` or `load` event listeners to avoid conflicts.

metafields

Snippets that use metafields reference the namespace `custom` (e.g., `product.metafields.custom.size_guide`). If your theme uses a different namespace, update the metafield references accordingly. Metafields must be created in **Settings > Custom data** in your Shopify admin.

currency formatting

Price formatting uses `| money` and `| money_without_trailing_zeros` Liquid filters, which respect the store's currency settings automatically.

accessibility

Snippets include basic ARIA attributes (`aria-label` , `role` , `aria-expanded`) where applicable. Full accessibility compliance depends on your theme's base implementation.

compatibility

Feature	OS 2.0	Legacy
All snippets	Yes	Yes
<code>{% render %}</code> tag	Yes	Yes (use <code>{% include %}</code> if needed)
Metafield access	Yes	Yes (Liquid access)
Section schema settings	N/A — snippets only	N/A

Tested against: Dawn 14+, Refresh, Crave, Debut, Brooklyn, Minimal.

support

For questions or issues, contact: support@ecomtools.dev

Please include:

- Your theme name and version
- The snippet name
- A description of the issue or error message

license

Single-store commercial license. You may use these snippets on one live Shopify store and its associated development/staging stores. Redistribution, resale, or inclusion in theme products for resale is not permitted.