

## Liquid for Shopify – Dev Cheatsheet

### 0) Mental model

-----

- \* **Liquid = templating language**: outputs strings/HTML by combining **objects** (Shopify data), **filters**
- \* **Deterministic**: no async, no DOM, no events. Think **server-side string builder**.
- \* **Scope**: variables are local to the current template/snippet unless explicitly passed.

### 1) Syntax at a glance

-----

- \* **Output**: `{{ product.title }}`
- \* **Logic/flow**: `{% if ... %}{% endif %}`, `{% for ... %}{% endfor %}`
- \* **Comments**: `{% comment %}...{% endcomment %}` or `{% # single line %}`
- \* **Whitespace control**: `{%- ... -}` or `{%- ... -%}` trims spaces
- \* **Assign**: `{% assign price = product.price | money %}`
- \* **Capture (multi-line string)**: `{% capture json %}{ "id": {{ product.id }} }{% endcapture %}`
- \* **Include partial**: `{% render 'card-product', product: product, index: forloop.index %}`
- \* **Cycle**: `{% cycle 'odd', 'even' %}`
- \* **Increment/decrement**: `{% increment counter %}`, `{% decrement counter %}`

### 2) Core objects (Shopify)

-----

- \* **Global**: shop, routes, settings, request, cart, customer, collections, linklists, blogs
- \* **Contextual**: product, collection, article, page, search, section, block
- \* **Checkout**: limited theme access
- \* **Metafields**: `product.metafields.namespace.key` → value object (`.value`, `.type`)
- \* **Metaobjects**: `shop.metaobjects['type']['handle']`

> JS analogy: objects are like server-provided JSON you can only read.

### 3) Truthiness & comparisons

-----

- \* Falsy: `nil`, `false`, `""`, `[]`
- \* `{% if product %}`, `{% if product.title != blank %}`, `{% unless condition %}`
- \* Operators: `==`, `!=`, `>`, `<`, `>=`, `<=`, `contains`, `and`, `or`

### 4) Loops

-----

```
{% for p in collection.products limit: 8 offset: 0 %}
  {{ forloop.index }} – {{ p.title }}
{% else %}
  <p>No products.</p>
{% endfor %}
```

Pagination:

```
{% paginate collection.products by 24 %}
...items...
{% if paginate.pages > 1 %}
  {{ paginate | default_pagination }}
{% endif %}
{% endpaginate %}
```

### 5) Filters you'll use constantly

-----

- \* Strings: `downcase`, `upcase`, `strip`, `truncate: 30`, `replace`, `split`, `join`
- \* Numbers: `plus`, `minus`, `times`, `divided_by`, `round`, `ceil`, `floor`

- \* Arrays: map, uniq, sort, reverse, where, first, last, size
- \* Formatting: money, money\_with\_currency, date, json
- \* URLs/Images: image\_url, image\_tag
- \* Safety: escape, newline\_to\_br, strip\_html

## 6) Sections & blocks (OS 2.0)

-----

Section file: sections/hero.liquid

```
{% schema %}
{
  "name": "Hero",
  "settings": [
    { "id": "heading", "type": "text", "label": "Heading" },
    { "id": "image", "type": "image_picker", "label": "Background" }
  ],
  "blocks": [
    { "type": "button", "name": "Button",
      "settings": [{ "id": "label", "type": "text" }, { "id": "link", "type": "url" }] }
  ],
  "max_blocks": 3,
  "presets": [{ "name": "Hero" }]
}
{% endschema %}
```

## 7) Snippets & render

-----

```
* {% render 'badge', status: product.available %}
```

## 8) Forms

-----

```
{% form 'product', product, id: 'add-to-cart' %}
<select name="id">
  {% for v in product.variants %}
    <option value="{{ v.id }}">{{ v.title }}</option>
  {% endfor %}
</select>
<button type="submit">Add to cart</button>
{% endform %}
```

## 9) Routing helpers

-----

```
* routes.root_url, routes.cart_url, routes.account_url
* {{ 'Cart' | link_to: routes.cart_url }}
```

## 10) Metafields & metaobjects

-----

```
* {{ product.metafields.custom.subtitle.value }}
* {{ shop.metaobjects['participant']['john-doe'].fields }}
```

## 11) Debugging

-----

```
* {{ product | json }}
* {% if some_var != blank %} ... {% endif %}
```

## 12) Common gotchas

-----

- \* "10" vs 10 in comparisons → cast with plus: 0
- \* contains differs for strings vs arrays
- \* blank vs empty
- \* Pass args explicitly to render

-----  
 Teaching Walkthrough (Start → Finish)  
 =====

- 1) Setup: clone, shopify theme dev, tour files.
- 2) Basics: output, tags, assign, if/for.
- 3) Filters & collections: where/map/sort.
- 4) Sections & schema: build feature-list.
- 5) Snippets & render: props style.
- 6) Images: image\_url + image\_tag.
- 7) Metafields: product.metafields.custom.subtitle.
- 8) Forms: {% form 'product', product %}.
- 9) Pagination & routes.
- 10) Debugging: json, blank guards.
- 11) Conventions: use render, guards, no hardcoded URLs.
- 12) Capstone: Featured collection section.

Quick JS/PHP equivalence  
 -----

Output → {{ x }} → echo x  
 Logic → {% if %}, {% for %} → if, for  
 Helpers → filters → helper functions  
 Partial → render → function calls  
 Data → product, collection → JSON objects