Shopify Cheatsheet

A quick reference guide to Shopify, covering essential features, Liquid templating, API interactions, and common tasks for managing your online store.

Shopify Basics

Core Concepts

Shopify Admin: The web-based interface for managing your store.

Themes: Control the look and feel of your storefront.

Apps: Extend Shopify's functionality with third-party integrations.

Products: Items you sell in your store.

Collections: Groups of products.

Orders: Records of customer purchases.

Navigation

Products	Manage your products, inventory, and variants.
Orders	View and manage customer orders.
Customers	Manage customer profiles and information.
Analytics	Track your store's performance with reports and dashboards.
Online Store	Customize your theme, manage navigation, and create pages.
Apps	Install and manage apps to extend Shopify's features.

Settings

General	Store details, currency, and time zone.
Payments	Configure payment gateways (e.g., Shopify Payments, PayPal).
Checkout	Customize the checkout process.
Shipping and delivery	Set up shipping rates and methods.
Taxes and duties	Configure tax settings.
Notifications	Customize email and SMS notifications.

Liquid Templating

Basic Syntax

`{{ variable }}` - Output a variable's value.

`{% tag %}` - Logic and control flow.

`{# comment #}` - Comments.

Common Tags

`for`	Loop through items in a collection.
	`{% for product in collection.products %}`
	`  {{ product.title }}`
	`{% endfor %}`

`if/else`	Conditional logic.
	`{% if product.available %}`
	`  <p>In Stock</p>`
	`{% else %}`
	`  <p>Out of Stock</p>`
	`{% endif %}`

`assign`	Assign a value to a variable.
	`{% assign my_variable = 'Hello' %}`
	`{{ my_variable }}`

`include`	Include a snippet. `{% include 'product-card' %}`

`case/when`	Switch statement.
	`{% case template %}`
	`  {% when 'product' %}`
	`  <p>Product Page</p>`
	`  {% when 'collection' %}`
	`  <p>Collection Page</p>`
	`  {% endcase %}`

Filters

`date`	Format a date. `{{ article.published_at | date: '%B %d, %Y' }}`
`money`	Format a number as currency. `{{ product.price | money }}`
`capitalize`	Capitalize the first word. `{{ product.title | capitalize }}`
`downcase`	Convert to lowercase. `{{ product.title | downcase }}`
`upcase`	Convert to uppercase. `{{ product.title | upcase }}`
`truncate`	Truncate a string. `{{ product.description | truncate: 50 }}`

Shopify API

API Basics

Shopify's API allows you to interact with your store's data programmatically. Use it to create, read, update, and delete resources like products, orders, and customers.

Authentication: Use API keys or OAuth to authenticate your requests.

Endpoints: Access resources via RESTful endpoints (e.g., `/admin/api/2023-07/products.json`).

Rate Limits: Be mindful of API rate limits to avoid being throttled.

Common Endpoints

`GET /admin/api/2023-07/products.json`	List all products.
`POST /admin/api/2023-07/products.json`	Create a new product.
`GET /admin/api/2023-07/products/{product_id}.json`	Get a specific product.
`PUT /admin/api/2023-07/products/{product_id}.json`	Update a product.
`DELETE /admin/api/2023-07/products/{product_id}.json`	Delete a product.
`GET /admin/api/2023-07/orders.json`	List all orders.

Example Request (Ruby)

```ruby
require 'shopify_api'

ShopifyAPI::Base.site = "https://{api_key}:{password}@{shop_name}.myshopify.com/admin"

products = ShopifyAPI::Product.find(:all)

products.each do |product|
  puts product.title
end
```

Common Tasks

Theme Customization

Editing Theme Files	Use the Theme Editor or download theme files to edit HTML, CSS, and Liquid code.
Creating Snippets	Create reusable code snippets for common elements like product cards or headers.
Using Theme Sections	Add customizable sections to your theme's homepage and other pages.
Implementing Custom CSS	Add custom CSS to modify the appearance of your theme.

Product Management

Adding Products	Add product details, images, and pricing information in the Shopify admin.
Managing Inventory	Track inventory levels and set up inventory alerts.
Creating Collections	Group products into collections for easier browsing.
Setting Up Product Variants	Add variants for different sizes, colors, and other options.

Order Fulfillment

Processing Orders	View and manage customer orders in the Shopify admin.
Fulfilling Orders	Mark orders as fulfilled and provide tracking information.
Handling Returns	Manage returns and refunds.
Using Shipping Apps	Integrate with shipping apps to automate shipping processes.