# Liquid – Les bases

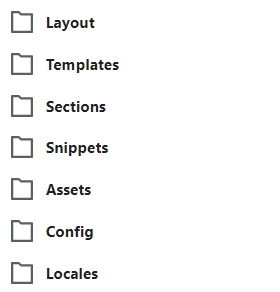
Liquid – Les bases

Le site git du langage Liquid : https://shopify.github.io/liquid/

Shopify utilise le langage Liquid, il n’est utilisé que pour Shopify. L’on peut se baser sur un thème existant où en crée un vide nous-mêmes :

Pour créer, un thème vide : <https://shopify.dev/tools/theme-kit/getting-started>

Lorsque le thème est récupéré, on se retrouve avec un arbre comme celui-ci, si « Sections » et « Snippets » ne se sont pas créé c’est normal, mais je conseille de le faire tout de suite :



Layout est quasiment vide, on peut y placer les fichiers le plus généraux comme « thème.liquid » ou « password.liquid ».

On retrouve dans ces fichiers les balises <head> et <body> ainsi que les include du Css et Js.

Locales regroupe les langues du site. De base un thème vide a le fichier « en.default »

Le dossier Assets regroupe les fichiers Json, >Js et Css.

Les dossiers Sections et Snippets sont explicites.

Une image contenant table

Description générée automatiquement

On retrouve dans le dossier Config deux fichiers «settings\_data.json » à ne pas modifier, il regroupe toutes les données du site.

Le deuxième « settings\_schema.json » est utilisé pour les paramètres généraux. Liquid met à notre disposition un système de schéma permettant à l’administrateur de personnaliser sa boutique.

Ce fichier utilise ce système et les données rentrées sont accessibles dans l’ensemble des fichiers de codes du site.

Exemple dans un fichier quelconque :

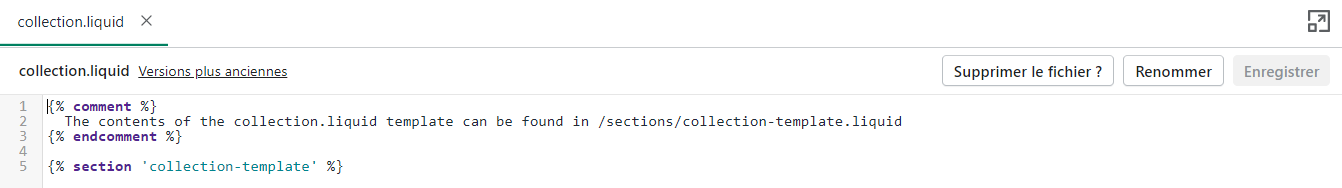
<div style= « color : {{ settings.id\_de\_ma\_couleur }} ;»>

{{ settings.id\_de\_mon\_texte }}

</div>

Le dossier Templates regroupe les fichiers qui gèrent toutes les fonctionnalités de bases de Shopify. La modification d’un fichier comme collection.liquid appliquera les modifications sur toutes le collections.

Généralement on va gérer en Html et Css les comptes, order etc de l’utilisateur, mais pour les autres fichiers on va simplement avoir :



Lorsque l’on veut créer un nouveau modèle de page ou d’article on crée le template, rajoute une section ou un snippet à l’intérieur avec :

{% include ‘mon-nom-de-snippet’ %} ou {% section ‘ma-section’ %}

Une image contenant texte

Description générée automatiquementLiquid utilise des conditions et des boucles comme n’importe quel langage :

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Il faut savoir que le langage Liquid est exécuté côté serveur il est donc impossible de récupérer une variable Css ou Js en Liquid mais l’inverse est faisable.

On crée une variable texte en Liquid quelque part {% assign mon\_Titre = product.title %}

Pour Css : $ma\_Variable\_css : {{ mon\_Titre }} ;

Pour Js : var ma\_Variable = {{ mon\_Titre | json }}

Pour Html : <div> {{ mon\_Titre }} </div>

On peut également faire les déclarations {% assign %} dans les zones Css et script.