Liquid uses a combination of **tags**, **objects**, and **filters** to load dynamic content. They are used inside Liquid **template files**, which are a group of files that make up a theme. For more information on the available templates, please see [Building themes](https://docs.shopify.com/themes/development).

Liquid使用一系列标签，对象和过滤器的组合来加载动态内容。通常被用来制作Liquid模板文件，一个主题由多个文件组成。如果想知道更多关于可使用模板的信息，请查阅模板设计。

## Tags

Tags make up the programming logic that tells templates what to do. [Read more ›](https://docs.shopify.com/themes/liquid/tags)

{% **if** user.name **==** 'elvis' %}

Hey Elvis{% **endif** %}

标签构成了程序的逻辑，告诉模板要做什么

## Objects

Objects contain attributes that are used to display dynamic content on a page. [Read more ›](https://docs.shopify.com/themes/liquid/objects)

{{ product.title }} <!-- Output: Awesome T-Shirt-->

对象有很多属性，他们是网页的内容

## Filters

Filters are used to modify the output of strings, numbers, variables, and objects. [Read more ›](https://docs.shopify.com/themes/liquid/filters)

{{ 'sales' | **append**: '.jpg' }} <!-- Output: sales.jpg -->

过滤器用来修改输出的字符串，数字，变量或者对象。

**Handles** are used to access the attributes of Liquid objects. By default, a handle is the object's title in lowercase with any spaces and special characters replaced by hyphens (-). Most objects in Shopify (products, collections, blogs, menus) have handles.

For example, a page with the title "About Us" can be accessed in Liquid via its handle about-us as shown below:

句柄用来访问Liquid对象的属性。Liquid默认一个句柄就是一个文件的标题（由小写字母和-组成）。大部分Liquid对象都有句柄。例如，一个用“About Us”标题的网页能够给被“about-us”句柄以各种方式访问。

<!-- the content of the About Us page -->

{{ pages.about-us.content }}

## Accessing handle attributes

访问句柄元素的方法

In many cases you may know the handle of a object whose attributes you want to access. You can access its attributes by pluralizing the name of the object, then using either the square bracket ( [ ] ) or dot ( . ) notation.

在很多情况下你想通过句柄来访问某个对象的属性，可以在两对大括号中间输入对象的句柄与某个属性的名字。

{{ pages.about-us.title }}{{ pages["about-us"].title }}

About Us

About Us

Notice that the example uses pages instead of page.

注意是pages.

You can also pass in **Customize theme** page objects using this notation. This is handy for theme designers who want to give the users of their themes the ability to select which content to display in their theme.

{% for product in collections[settings.home\_featured\_collection].products %}

{{ product.title }}{% endfor %}

在一对大括号与百分号之间输入特殊符号，可以通过访问主题页面中的文件来方便的控制页面中动态内容的显示。

Awesome Shoes

Cool Shirt

Wicked Socks

# Logical and comparison operators

Liquid has access to all the logical and comparison operators. These can be used in tags such as [if](https://docs.shopify.com/themes/liquid/tags/control-flow-tags/" \l "if) and[unless](https://docs.shopify.com/themes/liquid/tags/control-flow-tags/" \l "unless).

逻辑与比较操作符

Liquid可以使用所有逻辑和比较操作，他们在标签中被使用，比如if和unless.

## Basic operators 基本操作符

|  |  |
| --- | --- |
| == | equals |
| != | does not equal |
| > | greater than |
| < | less than |
| >= | greater than or equal to |
| <= | less than or equal to |
| or | condition A **or** condition B 条件A和条件B |
| and | condition A **and** condition B |

For example:

{% **if** product.title **==** "Awesome Shoes" %}

These shoes are awesome!{% **endif** %}

You can use multiple operators in a tag:

{% **if** product.type **==** "Shirt" or product.type **==** "Shoes" %}

This is a shirt or a shoe.{% **endif** %}

ConditionA OR ConditionB

## The contains operator contains 操作符

contains checks for the presence of a substring in a string.

用来检查在一个字符串中子串的存在

{% **if** product.title **contains** "pack" %}

This product's title contains the word Pack.{% **endif** %}

contains can also check for the presence of a string in an array of strings.

也可以被用来检查在一个字符串数组中某个字符串的存在

{% **if** product.tags **contains** "Hello" %}

This product has been tagged with 'Hello'.{% **endif** %}

contains can only search strings. You cannot use it to check for an object in an array of objects.

只能用来寻找字符串，不能用来寻找对象数组中某个对象的存在

# Data types

Liquid objects can be one of six types. You can initialize Liquid variables using [assign](https://docs.shopify.com/themes/liquid/tags/variable-tags/" \l "assign) or [capture](https://docs.shopify.com/themes/liquid/tags/variable-tags/" \l "capture) tags.

数据类型 Liquid对象可以是六种中的任何一种，可以初始化变量用assign或者capture

## String

Strings are declared by wrapping a variable's value in single or double quotes.

String变量用单引号或双引号中间的内容声明

{% assign my\_string **=** "Hello World!" %}

## Number

Numbers include floats and integers.

包括浮点和整形

{% assign my\_int **=** 25 %}{% assign my\_float **=** 39.756 %}

## Boolean

Booleans are either true or false. No quotations are necessary when declaring a boolean.

Booleans 取值true或者false ，不能用引用声明一个BOOL型变量

{% assign foo **=** **true** %}{% assign bar **=** **false** %}

## Nil

Nil is a special empty value that is returned when Liquid code has no results. It is **not** a string with the characters "nil".

Nil是一个特殊的空值，作为Liquid代码没有结果的时候的返回值。”nit”不是一个字符串

Nil is treated as false in the conditions of if blocks and other Liquid tags that check the truthfulness of a statement.

Nil在if模块和其他用来检测statement的真实性中被看做false.

In the following example, if a tracking number does not exist (that is, fulfillment.tracking\_numbers returnsnil), Liquid will not print the text:

在接下来的例子中，Liquid不会打印其中的文本

{% **if** fulfillment.tracking\_numbers %}

There is a tracking number.{% **endif** %}

Tags or outputs that return nil will not print anything to the page.

标签或者输出返回nil将会没有任何输出

Tracking number: {{ fulfillment.tracking\_numbers }}

Tracking number:

## Array

Arrays hold lists of variables of any type.

数组保存任何形式变量的列表

### Accessing items in arrays

访问数组的方法

To access all of the items in an array, you can loop through each item in the array using a [for](https://docs.shopify.com/themes/liquid/tags/iteration-tags/" \l "for) or [tablerow](https://docs.shopify.com/themes/liquid/tags/iteration-tags/" \l "tablerow)tag.

用for或者tablerow标签循环访问数组中的每个元素

{% for tag in product.tags %}

{{ tag }}{% endfor %}

sale summer spring wholesale

### Accessing specific items in arrays

访问数组中特定的元素

You can use square bracket [ ] notation to access a specific item in an array. Array indexing starts at zero.

可以使用方括号标记来访问数组中特定的元素。数组的下表从0开始。

<!-- if product.tags = "sale", "summer", "spring", "wholesale" -->{{ site.users[0] }}{{ site.users[1] }}{{ site.users[3] }}

sale

summer

spring

### Initializing arrays

You cannot initialize arrays using only Liquid.

只能用Liquid初始化数组

You can, however, use the [split](https://docs.shopify.com/themes/liquid/filters/string-filters/" \l "split) filter to break a single string into an array of substrings.

可以使用split过滤器把一个字符串分成一个有字串组成的字符串

## EmptyDrop

An EmptyDrop object is returned if you try to access a deleted object (such as a page or post) by its[handle](https://docs.shopify.com/themes/liquid/basics/handle). In the example

EmptyDrop对象在尝试访问一个已经删除的对象的句柄时被返回，比如下面的例子

below, page\_1, page\_2 and page\_3 are all EmptyDrop objects.

Page\_1,page\_2,page\_3都是EmptyDrop对象

{% assign variable **=** "hello" %}{% assign page\_1 **=** pages[variable] %}{% assign page\_2 **=** pages["does-not-exist"] %}{% assign page\_3 **=** pages.this-handle-does-not-exist %}

EmptyDrop objects only have one attribute, empty?, which is always true.

EmptyDrop对象永远只有一个empty属性而且它的值永远为真。

Collections and pages that do exist do not have an empty? attribute. Their empty? is “falsy”, which means that calling it inside an if statement will return false. When using an unless statement on existing collections and pages, empty? will return true.

集合和网页没有empty属性，他们的empty属性是false，

### Checking for emptiness

You can check to see if an object exists or not before you access any of its attributes.

检查网页是否为空

在访问他的属性之前可以检查他是否存在

{% **unless** pages **==** empty %}

<!-- This will only print if the page with handle "about" is not empty -->

<h1>{{ pages.frontpage.title }}</h1>

<div>{{ pages.frontpage.content }}</div>{% **endunless** %}

If you don't check for emptiness first, Liquid might print empty HTML elements:

<h1></h1><div></div>

You can check for emptiness with collections as well:

{% **unless** collections.frontpage **==** empty %}

{% for product in collections.frontpage.products %}

{% include "product-grid-item" %}

{% **else** %}

<p>We have a "frontpage" collection, but it's empty.</p>

{% endfor %}{% **endunless** %}