# Sharing Common Functionality across Properties

Agility’s page, module, and content structure allow you to build a bespoke content schema that addresses your specific sites’ requirements. Usually, this means that each Agility instance varies in terms of what modules and functionality are available.

This is all great, but what happens when you have another website with similar requirements and you want to re-use functionality that was built for an existing site? Do you need to re-create the module from scratch just for this new website, or with proper planning, can we design modules to be website agnostic?

## How

Generally speaking, a web page in Agility is composed of one or many Modules that encapsulate functionality.

Therefore, to share common functionality across web properties, you need to make those same Modules available for use where you need them.

Modules are comprised of an input form of *fields* that editors use to create and manage content. Modules also have corresponding *code* that gets executed when the Module is rendered on a page. All you need to do is ensure that each web property has the same module, and corresponding code to be executed.

Content, Module, and Page Template Definitions can be copied between instances or re-used across channels within the same instance, allowing you to keep a consistent content schema across web properties. This is all done within the Agility Content Manager.

Code for your modules can be copied between websites as well. This is up to the developer. With the advent of modern package managers such as Nuget and NPM, this has become much easier to implement and maintain.

## Benefits of Sharing Common Functionality

Having the same modules available to your editors across web properties will not only increase their familiarity with your CMS instance(s), it will also make them more productive.

For developers, this means you can likely limit the number of modules you develop on a regular basis, as you re-use existing modules with minimal effort. This often leads to an increase in quality as well as developers and QA can spending more time testing and enhancing features, all while increasing the value of these modules across your properties.

## Guiding Principals

Whether you have multiple Agility instances or an instance with multiple channels, the general philosophy is the same.

1. You have Modules that have the same fields across all web properties and can be added to a page.
2. The source code for these Modules are centralized and distributed using a package manager.
3. Keep these Modules as generic and simple as possible. Tailoring the modules too much to a specific use case can affect its ability to be used in other web properties.
4. Keep all functionality, including HTML, JS and CSS self-contained so it will work on any website *out-of-the-box*.