"Teach Yourself ASP.Net in 21 Days" is an introduction to Microsoft’s new ASP.Net approach to developing web-based applications. Microsoft is placing the .Net label on just about everything it produces these days, so trying to define .Net can be tricky. Luckily ASP.Net is easier to pin down. ASP.Net is Microsoft’s successor to its very successful Active Server Pages (ASP) framework for developing applications which run on the web.

Some examples of those applications include: On-line questionnaire or registration forms, checking of the shipping status for orders, on-line catalogs tied to inventory databases which automatically reflect any changes to the inventory database, and the creation of electronic shopping carts to make it easy to select and order items from the on-line catalog.

As these types of applications became more prevalent and grew more complex, the limitations of ASP became apparent. The good news is that ASP.Net seems like an answer to many of the criticisms raised against ASP, such as: a tendency towards a “spaghetti code” mixing of ASP and HTML in the same file, a lack of object orientation, difficulties associated with creating and updating business components and a number of scalability issues, particularly when storing information in session variables. ASP is also not particularly well suited for working with important new technologies such as XML and Web Services

In addition, the development tools available to ASP programmers are a far cry from the rich set of editors and debugging tools available to desktop application developers. The good news with ASP.Net is that the new Visual Studio.Net finally provides the type of integrated development environment desktop developers expect, but with the capability to easily create desktop, Web based and Web services applications.

Chapters 1 – 7 cover the basics of using ASP.Net. Of particular interest is chapter 5 which covers something new to ASP.Net – Web Server Controls. Server Controls are the new ASP.Net way to hide much of the complexity of developing applications in the stateless environment of the Web, with a look and feel much like the rich development environments that desktop developers have come to expect.
The focus of Chapters 8 – 14 is on interacting with data sources with ADO.Net. These chapters really do not do justice to the richness of ADO.Net, but in fairness to the authors, entire books now exist just on this topic. These chapters do an adequate job of hitting the high points and suggest the key role that XML plays in ADO.Net. XML is the foundation which ADO.Net is built upon and it is fairly effortless for a developer to both read from and write to XML data files. Chapter 14 covers another new and interesting aspect of ASP.Net: caching. Caching provides another tool for the developer interested in temporary storage of data in a scalable manner.

Chapters 15 – 21 cover a variety of interesting topics such as: the new ASP.Net way to create business objects, creating and consuming Web Services, debugging ASP.Net pages and adding security to your ASP.Net pages. Each of these topics is of great interest to Web developers and in each case ASP.Net offers new or improved tools to accomplish the task.

ASP.Net was officially released only a few months ago so it is still too early to tell how successful it will be. But as someone who has been working with and teaching ASP for as long as it has existed, ASP.Net looks like a very promising successor to ASP. If you are interested in exploring this new technology, “Teach Yourself ASP.Net in 21 Days” should provide a pleasant introduction to this promising new technology.

Dan Clapper is an associate professor and chair of the Business Computer Information Systems and Economics Department in the College of Business at Western Carolina University. He teaches application development for both the desktop and World Wide Web environments.