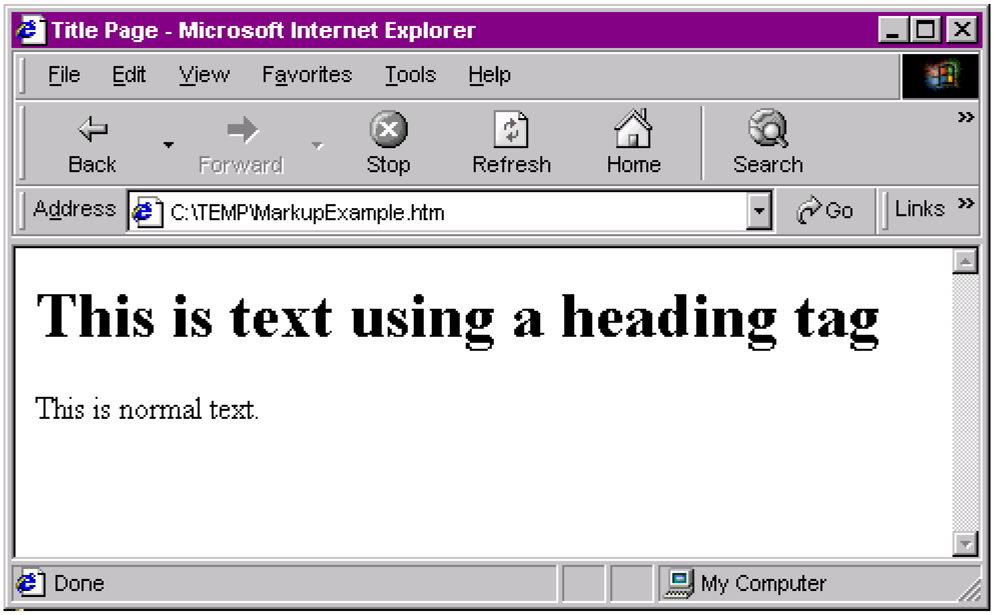
What is a Markup Language?

Before we talk about the features of a markup language, let's define what a markup language is. In simple terms, a *markup* refers to the use of characters within a piece of information that can be used to process or identify that information in a particular way. A good example of a markup can be seen in an HTML document. The following example uses greater-than and less-than symbols to identify markup elements, or tags, that have specific purposes.

```
<HTML>
  <HEAD>
    <TITLE>Title Page</TITLE>
  </HEAD>
  <BODY>
    <H1>This is text using a heading tag</H1>
    This is normal text.
  </BODY>
</HTML>
```

If the above HTML document were viewed in a browser, the browser would interpret the markup elements and display the content to reflect the author's intentions. For example, the <H1> and </H1> tags are used to display the text in a large font, whereas the text immediately following it is displayed in the browser's standard font.



C:\TEMP\MarkupExample.htm

This is text using a heading tag

This is normal text.

There are a number of different markup languages and types.
We are going to review three of the most common.

SGML

Standard Generalized Markup Language (SGML) was designed as a standard way to store data independent of any software application or platform. SGML is often referred to as a *meta language*. Meta languages are languages that are used for describing *markup* languages. HTML is a derivative of SGML and is therefore called an SGML application. There are a number of languages based on SGML. There are also a number of standard data formats based on SGML.

The real power behind SGML is its ability to declare *Document Type Definitions,* or DTD's, which we will discuss in detail later. For now, it is only important to understand that SGML provides the ability to define the contents of the document, its markup characteristics, and its information model.

The downside to SGML is that it is all encompassing, and has a lot of rules. It has so many aspects to it that it is almost impossible to implement all of them. For this reason, SGML is rarely used by itself. Instead, subsets have been created that target niche applications and needs.

HTML

Hyper Text Markup Language (HTML) is the first internationally accepted derivative of SGML. HTML is really the document language of the World Wide Web. However, HTML is rather limiting. It was originally designed to represent document based data within a browser in very basic form. It has evolved over time to support application features through the use of JavaScript, Java Applets, and the inclusion of client-side plug-ins, but as a whole it is still very basic. Furthermore, it is not a good language for applications to store and share data. One of the primary reasons for this is its lack of support for DTD's. Remember DTD's are external elements that define the contents and structure of the data. HTML's structure is extremely limited compared to its predecessor SGML.

Then again, that is not what HTML was designed for. It was designed as a document language, not a data language. This is where XML enters the picture.

XML

XML incorporates many of the features of SGML while learning from the limitations of HTML. Like SGML, XML utilizes DTD's, making it flexible and extensible. The goals of XML as defined by its creators were more focused than those of SGML making it much easier to implement. These goals included:

XML could be used with existing Internet protocols (HTTP, MIME, etc.). This makes it the ideal format for sharing information on the Internet.

XML support is application independent. Any application can utilize and support XML documents.

XML is platform independent. Its use of technologies such as Unicode make it portable across machine types.

XML is license free. It is controlled by an international standards organization. This means that it isn't going to cost you anything to use it.

XML is compatible with SGML.

The feature set of XML was kept to a minimum so that applications could support it. Compare this goal with that of SGML.

XML is a family of technologies. XML has already evolved to include support for such things as style sheets, hyperlinks, and the Document Object Model (DOM).

XML takes the best of SGML (structured data definition capabilities) and the best of HTML (web addressing) . The result is a portable, highly usable, markup language that can be used by any number of applications to store and share structured data. Applications that will benefit or are already benefiting from XML include:

Office applications (word processors, spreadsheets, etc.)

Web applications (browsers, e-mail, etc.)

Server applications (database servers, e-mail servers, etc.)

At its core, XML appears very simple. However, the implications of its use are very complex. It is already changing the way that people store information and build applications. Microsoft, Netscape,
Sun, and many others are already using XML today in their applications, database servers, and e-commerce platforms.