



By Amy Cowen  
(9/7/99)

Before you close your next <HEAD> element, you should spend some time describing your page using meta tags. Style sheets work to separate the formatting of your page from the structure of the page. But when you turn to meta data, you will have to step back even further from the page and look at your document as a whole. What can you say *about* the document?

Whether you're [hoping to talk to a search engine](#) or a local script, or you just want to prepare your pages for future technologies that may take advantage of standardized element sets such as the [Dublin Core](#), you can start using meta tags today.

Because these tags don't appear on the browser screen, they tend to be ghostlike elements that you catch a glimpse of here and there. In response to common questions about meta data and meta tags, we've gathered answers designed to help you polish your meta speak.



1. [What Is a Meta Tag?](#)
2. [How Do I Use Meta Tags?](#)
3. [What Meta Tags Do Search Engines Read?](#)
4. [Can I Control What Pages Are Indexed?](#)
5. [How Do I Set Up a Redirect?](#)
6. [Can I Stop Cached Pages?](#)
7. [What Is the Resource Description Language?](#)
8. [What Is the Dublin Core?](#)
9. [Can I Rank My Site for PICS?](#)
10. [Is There More?](#)



## What Is a Meta Tag?

*Meta tags* are used to define meta data. So, what is meta data? In the most general terms, *meta data* is information about a document. The [World Wide Web Consortium's](#) definition is a bit more techno and Net specific: "Meta data is machine understandable information for the Web."

Don't worry, you don't have to practice your R2-D2 speak to work with meta tags. They are actually very easy to use.

`<META name="pizza" content="x-large cheese">` is a meta tag. It means something to me. That's why I decided to include it in my file--a mental note to myself. It's not, however, a meta tag that will have meaning to most automatons looking at my HTML file. This points to a problem with meta tags: the lack of standardization.

Right now, there are no set standards defining what meta tags can be used, what attributes are viable, what property terms mean, and how the information should be used. You can create any meta tags you want and write your own set of scripts to take advantage of that information.

If you tend to prowl the Web in View Source mode, you may spot homegrown meta tags on corporate sites where a localized search engine has likely been configured to read company-specific meta information. Even if you don't plan to do anything with the meta data, you can still use the tags, such as the pizza attribute above. They won't change the way the page renders, so your meta tags can give you a way to annotate your documents. For example, the following information might be useful to me and my team but won't have meaning for robots that drop by my site:

```
<META name="author" content="Amy Cowen">
<META name="team" content="Blue Marketing">
<META name="owner" content="Dibbert Q">
<META name="revision" content="1; June 1 1999">
<META name="revision" content="2; August 14 1999">
```

## A Philosophical Approach

Traditionally, the information we put on the page is content. Meta data is information used to describe or define that content. So, it's not content in and of itself, and it doesn't appear on users' screens. Instead, meta data sits silently in the HEAD element, adding commentary to the page for anyone--or anything--who cares to look.

If you've been poking around [XML](#), then this snapshot of the META element declaration from the [HTML 4 Specification](#) should be relatively easy to decode:

```
<!ELEMENT META - O EMPTY -- generic metainformation -->
<!ATTLIST META
%i18n; -- lang, dir, for use with content --
http-equiv NAME #IMPLIED -- HTTP response header name --
name NAME #IMPLIED -- meta information name --
content CDATA #REQUIRED -- associated information --
scheme CDATA #IMPLIED -- select form of content -->
```

You can see that the META element's attribute list (ATTLIST) includes the following attributes: `http-equiv`, `name`, `content`, and `scheme`. In order to speak the meta lingo to any robots you encounter, you'll want to familiarize yourself with [http-equiv](#), [name](#), and [content](#).

To have such a small number of possible attributes, you may be surprised at how much you can do with meta tags. From setting your page up to [roll to another URL](#) to [rating your site's content on the PICS scale](#), meta

tags offer some nice--and easy to apply--functionality.

### **A Hodgepodge of Tags**

The HTML specification does not define what values can be used within the META element. Instead, values are defined in a profile, and individual sites reference the profile to which their tags correspond in the HEAD element. More than one site can, theoretically, share the same profile and take advantage of the same schema for applying meta tags. Turning to a centralized source in this way resembles XML's attention to namespaces and projects such as the [Dublin Core](#). The Dublin Core is, in effect, a profile for meta data.

Because the current state of meta data could culminate in a Tower of Babble--give ten people the same information and ask them to assign either meta or XML tags to see how quickly meaning spirals out of control when there are no overarching schematics to follow--[the W3C Resource Description Framework \(RDF\)](#) is currently under development and promises a common framework for meta data.



[Back to intro](#)

## How Do I Use Meta Tags?

Meta tags are easy to implement on your Web pages. They look just like regular HTML tags, but they follow a few rules:

- All begin with the word *META*, the element name.
- All are located within the HEAD element.
- No closing tag is used; the META element is an empty element.

When you use a meta tag, you define attribute/value sets. For each attribute, you assign a value. If you look back at the element [declaration](#), you'll see there are several available attributes, only one of which is required. So, the format might be diagrammed as: `<META attributeName="value" attributeName="value">`.

And, since content is required, it will always appear in your meta tags.

This structure can be seen in all of the following meta tags:

```
<META name="author" content="Amy Cowen">
<META name="title" content="Mastering the Meta Tag">
<META http-equiv="refresh" content="5;http://www.cnet.com/">
<META name="Generator" content="Microsoft FrontPage 4.0">
<META name="description" content="Answers to Your Top Questions Regarding Meta Tags.">
```

All of these tags provide different information about the page. None of them contribute to how the content is formatted or looks.

## Automating Meta Tag Creation

Although meta tags are not hard to write, a number of tools exist to make writing them even easier. Tools such as the [Meta Tag Generator](#) and [META builder](#) allow you to fill in the information related to various commonly used meta values (e.g., author, keywords, description, robots). After you've filled in the blanks, you'll be presented with (or emailed, depending on the tool) a full set of plug-and-play meta data that you can simply paste onto your page. See the list at right for more online meta tag-building tools.

For an online evaluation of your meta tags, try visiting the [Meta Medic](#). Enter your URL, and the program analyzes your meta tags. You can pick up some good pointers by periodically testing your meta data with a resource like this.



## What Meta Tags Do Search Engines Read?

Although there are no concrete rules defining what information a search engine, spider, or other creepy Web crawler pulls from your page, many search engines do make use of two primary meta tags: description and keywords.

Both of these meta tags let you further define your site's content and function so that those searching for your site's content have a better chance of finding you and can get an accurate sense of what your site offers.

Although some search engines index all the words on a Web page, often the first few words get the most exposure. For example, in the absence of a description meta tag, [AltaVista](#) uses the first few words on the page when it returns a set of search results.

By stepping back just a bit and using two targeted tags to summarize your site and pinpoint the most relevant terms, you hone in on what your site is really about. It's possible your site is really about growing herbs, but if terms and strings such as *herb gardening*, *herbs*, and *garden* don't appear on the main page, then your presence in a search engine isn't going to be overly accurate. If your main page, instead, talks about weather, BLT sandwiches, recipes using basil, and the newest insect threat, you may find your site coming up for all kinds of only remotely related searches.

Using the description tag, you'll enter a sentence or two describing the page. Many search engines use this information when a list of search results is returned.

The description tag for our gardening site might look like this:

```
<META name="description" content="Find out everything you need to know about growing herbs in your backyard.">
```

This description is short and to the point. It doesn't try to account for every single page on the site. Instead, it's a solid summarization of the site's primary content. As you can imagine, the description for your home page will be broader than the ones you might use on subpages.

[WebPromote](#) suggests that you keep your description at less than 25 words. Thinking that you'll really get noticed if your description is the longest one returned, you might be tempted to put in a whole paragraph or more, but don't waste the kilobytes. Search engines use a set amount of characters in the descriptions they return. If yours is too long, it will be cut off abruptly.

The rule of thumb is to try and create a sentence that will make users want to click through to your site. Crafting a simple sentence that can do that effectively will serve you better than rambling your way through several lines in an attempt to list everything, including the kitchen sink, about your site.

After carefully considering the site's function and audience, the keywords tag for our gardening site might resemble the following:

```
<META name="keywords" content="herbs, herb garden, growing herbs, gardening, garden, herbology">
```

You can always revise and add to your keywords listing. By filling in the relevant terms, phrases, and concepts related to your site, you increase your chances that people searching for the information you have will find you. Because many search engines are case sensitive, WebPromote also suggests that you use lowercase keywords, as this is how most users enter search terms.

Not all search engines use all of these meta tags. For example, [Excite](#) does not use meta keywords in indexing

sites. It does, however, use the meta description for generating the summary of the site.

The title returned by many search engines comes from the TITLE container on your HTML page, so don't overlook this element. In addition to the description and keywords values, some search engines honor the robots tag.

Since keeping track of which search engines partake of meta data and which ones ignore it can be difficult, your best bet is to use these tags effectively. This is especially true on your first (or kome) page, but good use of meta tags on all subpages is something you should consider.



[Back to intro](#)

## Can I Control What Pages Are Indexed?

Creating a *robots* text file lets you specify whether robots, spiders, and other crawlers will index your site. But, you can also talk to the robots via the robots meta tag.

This tag uses the name value *robots* and has two properties for content: *index* and *follow*.

The following sample robots tag tells automatons a) to not index this page, and b) to not follow links from this page to other pages:

```
<META name="robots" content="noindex, nofollow">
```

The available values for each pair are:

index *or* noindex  
follow *or* nofollow

According to the [HTML Author's Guide to the Robots META tag](#), you can also use *all* or *none* to further condense the line. So, if you want the related values index and follow, you could use content="all". If you want noindex and nofollow, you can use content="none".

AltaVista also recognizes two additional values: noimage and noimageclick. See AltaVista's [Excluding Pages with META Tags](#) for more information.



## How Do I Set Up a Redirect?

Meta tags are very commonly used to redirect one page to another page or to refresh the page. In fact, you have probably visited numerous pages that use a meta refresh, and you may not have even realized that's what was happening.

The refresh meta tag contains a few core elements and follows this format:

```
<META http-equiv="refresh" content="time;URL=url">
```

Notice that the type of meta tag here is *http-equiv* rather than *name*. This indicates some information needs to be passed along to the server, in a format equivalent to an HTTP header. The value for this specific meta tag is refresh.

For the content section, you are specifying both the time interval (in seconds) and the URL that the page should load after the time has elapsed. If you list the same URL as the source page, the page will simply reload. If you list a different URL, the browser will access and load the specified page automatically.

The refresh meta tag offers a very low-tech method for loading another page. Three of the most common uses for this tag are splash screens, "we've moved" pages, and slide shows.

Most sites that use a splash screen generally roll automatically to another page after a set amount of time. So, you might see an announcement, a cool graphic, or some other welcome screen and then be transported to the home page for the site after about ten seconds.

Whether you've switched to another ISP or just rearchitected your site, you may find yourself with a set of old URLs that people will be trying to access but that don't point to your current site. You want to bring those visitors with you, but it can take a while for everyone to update their bookmarks, and your old site may stick in search engine listings for quite some time. The refresh tag can help. You can leave a page there that explains you've moved on and provides a link to the new address, which contains a refresh meta tag. The visitor can then click through to the new URL or be transported there automatically.

Here's a sample of how the code for a "we've moved" page might look:

```
<HTML>
<HEAD>
<META http-equiv="refresh" content="10;URL=http://www.theflap.com/">
<TITLE>theFLAP Has Moved!</TITLE>
</HEAD>
```

```
<BODY>
TheFLAP has moved to a new home at http://www.theflap.com/. You'll be taken there automatically in a few
seconds. If nothing happens, please use the link above. Don't forget to update your bookmarks! </BODY>
</HTML>
```

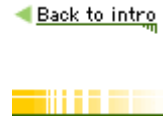
You can customize the message to say whatever you want. You may, in fact, not want to use a message at all and instead just set the refresh to 0 so that it happens automatically. But, if you're worried that someone may get stuck on a blank page, the message and alternate link is a good safety measure. Customers who get stuck on a blank site tend not to be repeat customers!

An innovative use of the refresh tag is to link a series of pages together creating a "tour" of a set of content.

Imagine, for example, that you have coded a photo journal of a recent trip (the techie version of "My Summer Vacation"). You may have created all the HTML pages, each with a photo and some text chronicling the image. Using the refresh tag, you can automatically take the user through your tour, page by page, without her ever having to click a Next button.

Be sure to set the refresh for enough time to allow the image to load and for the visitor to both look at the image and read any text. You don't want anyone getting seasick trying to keep up with your pictures, especially if your summer vacation involved a trip to the ocean!

Because motion can be a serious problem for some users, an implementation of the refresh tag on a series of pages might warrant a warning that the pages will change automatically. You would probably want to include a note to this effect on your welcome page, anyway. After all, the novelty of your automated tour is that your users can sit back and relax while they surf your site.



## Can I Stop Cached Pages?

There is a meta tag that specifies when a page expires. If all search technologies understood this tag, you could, in effect, ensure that a browser loads the page live rather than using a cached copy. However, not all browsers read the tag. Proxies, too, are often immune to your anticache attempts.

However, if you've ever had to fend off calls from anxious clients who can't understand why the page they're seeing doesn't reflect the changes they sent you, using an *expires* tag may be worth a few extra keystrokes. It won't solve all your problems, and you may still have to explain the principles of caching and the reload button, but a simple expiration line might cut down on most of your cache problems.

Like redirecting to another page, this meta tag involves the http-equiv attribute:

```
<META http-equiv="expires" content="Sat, 1 Jan 2000 00:00:00 GMT">
```



[Back to intro](#)

## What Is the Resource Description Framework?

While meta data seems to offer lots of power, it holds more promise than functionality right now because of the lack of standards. Just like with XML, you could, in theory, create an almost unlimited number of meta tags. But, if no one else is using those tags, or if browsers are not being created to perform specific functions when presented with certain tags, then meta tags remain largely in a vacuum.

As the W3C points out in its [Metadata Activity Statement](#):

Missing is a part of the Web which contains information about information--labeling, cataloging and descriptive information structured in such a way that allows Web pages to be properly searched and processed in particular by computer. In other words, what is now very much needed on the Web is metadata. W3C's Metadata Activity is concerned with ways to model and encode metadata.

The Resource Description Framework (RDF) is a W3C initiative designed to help standardize and define how meta data should be used. As it is right now, while there are some meta tags that have become commonly used and a variety of meta-related languages and technologies underway (e.g., XML, Dublin Core), no specification encompasses and standardizes meta data for the Web.

The RDF promises a structured method for creating and defining meta data terms in a "schema," which can then be used by others. Because the same term can have a different meaning in different circumstances and when applied to different kinds of content, schemas and namespaces are invaluable.

Variations in the meaning of a term such as *address* are commonly used to illustrate the problem posed by meta data that is not tethered to a particular schema.

Multiple sites might use an address meta tag:

```
<META name="address" content="123 Sunshine Road"> <META name="address"
content="http://www.cnet.com/">
```

As you can see, both of these content values are addresses, but they have very different meanings.

In a grocery list of potential uses for RDF, the W3C lists the following: thesauri and library classification schemes, Web site maps, description of the contents of Web pages describing the formal structure of privacy practice descriptions, rating systems, expressing meta data about meta data, and digital signatures.



## What Is the Dublin Core?

If you're talking meta speak these days (whether it's [RDF](#), [meta](#), or [XML](#)), you won't get far before you run into the Dublin Core, a set of 15 elements designed to "facilitate discovery of electronic resources" by providing a "common core of semantics for resource description." Don't be put off by the linguistics of it all. Meta data is, after all, data *about* data.

In establishing a core semantics, the Dublin Core is creating a structured system for talking about resources. To make your pages talk the Dublin Core talk, all you have to do is use meta tags or XML tags that take advantage of the 15 core elements.

## The Dublin Core in Action

In XML, you would specify you were using the Dublin Core schema by using this namespace declaration:

```
<?xml:namespace ns="http://www.purl.org/metadata/dublin_core">
```

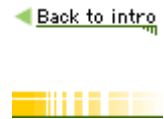
In HTML, your meta tags may use the Dublin Core schema or profile (particularly if they are generated by an online tool such as [META builder](#). You specify the profile you'll be using within the HEAD element:

```
<HEAD PROFILE="http://purl.org/metadata/dublin_core">
```

The format for each individual meta tag follows the general meta tag structure with one exception: each name is preceded by *DC:*. Those familiar with XML will recognize using an abbreviation this way as part of working with any namespace. The *DC:* abbreviation specifies the attributeName as one tied to the DC. Thus, the meaning of the content value you specify will be assumed to fall under the realm of meaning encompassed by the *DC:* element.

The 15 established Dublin Core properties that can be set via meta tags or used as XML elements include: TITLE, CREATOR, SUBJECT, DESCRIPTION, PUBLISHER, CONTRIBUTOR, DATE, TYPE, FORMAT, IDENTIFIER, SOURCE, LANGUAGE, RELATION, COVERAGE, and RIGHTS.

These properties are defined [here](#). Remember, in XML, case matters, so it's important to note that the Dublin Core elements are all expressed in uppercase.



## Can I Rank My Site for PICS?

The [Platform for Internet Content Selection](#) (PICS) specification allows labeling based on the kind of content included on a specific site. Many familiar Web-filtering software packages (such as Cybersitter) use PICS labels to help filter content.

Within your HTML page, PICS labels are embedded as meta tags. Unlike most of the other meta tags discussed in this article, PICS labels are not ones you write yourself.

Some sites and services rate material independently. In other words, you can't control the ratings assigned to your site, and your site may have been "ranked" even though you didn't add any tags or request a ranking.

Other services, however, let you self-rate your content. Generally, you answer specific questions or make selections from drop-down menus, and a label is generated that you can then paste into your pages. PICS labels--and the vocabularies used to generate the labels--vary, so it is acceptable to use more than one service and house more than one tag on your page. View the source of the [W3C](#) home page to see two PICS labels representing different self-labeling services.

You can choose to create either generic or specific PICS labels. Specific labels cover only certain pages, letting you label some sections of your site differently than others. For example, if you publish movie reviews, some of them may require different labeling than others. By using specific labels where necessary, you can maintain a wide audience rating on the rest of your site.

If you just use a generic label, it only needs to be applied to the first page of your site and will encompass all the other pages.

The PICS specification shows the format for a PICS label as follows:

```
(PICS-1.1 <service URL> [option...] labels [option...] ratings (<category> <value> ...) [option...] ratings (<category> <value> ...) ... <service URL> [option...] labels [option...] ratings (<category> <value> ...) [option...] ratings (<category> <value> ...) ... ..)
```

Here is a sample meta tag containing a PICS label generated at [SafeSurf](#):

```
<META http-equiv="PICS-Label" content='(PICS-1.1 "http://www.classify.org/safesurf/" l gen true for "http://www.theflap.com/" r (SS~~000 1))'>
```

(This tag is shown here for demonstration purposes only.)

So that you can see how PICS labels can vary, here is the meta tag that appears on [CNET.com's home page](#):

```
<META HTTP-EQUIV="PICS-Label" content=(PICS-1.1 "http://www.rsac.org/ratingsv01.html" L GEN TRUE COMMENT "RSACi North America Server" BY "chris_barr@cnet.com" FOR "http://home.cnet.com/" on "1997.07.14T16:27-0800" r (n 0 s 0 v 0 l 0))>
```

If you look closely, you can see the PICS structure in both of these examples, but, even so, the format isn't obvious or intuitive. Trying to write or alter your own PICS labels isn't advised!



## Is There More?

The following meta tags can help you further describe your content, the languages, and the technologies you've used to create your content:

- Use `META name="author" content="authorName"` for the document creator.
- Use `META name="date" content="date"` for the date created.
- Use `META http-equiv="Content-Script-Type" content="type"` to specify the scripting language used.
- Use `META http-equiv="Content-Style-Type" content="text/css"` to specify the default style sheet language.
- Use `META http-equiv="Content-Type" content="text/html; charset=characterSetName"` to specify the document character set in use on the page.

As work on the [RDF](#) progresses, the world of meta data will continue to become more robust and more important in your day-to-day HTML coding.