

# WORDS



A Quarterly Bulletin for Technical Writers & Communicators

Volume 1 | Issue 1 | February 2009

## The beginning

This is the first issue of *Words*, a quarterly bulletin for technical writers and communicators—indeed, for anyone with a passion for words and for what you can do with them.

Each issue of *Words* will include articles on language, authoring tools and authoring methodologies. Occasionally, other topics will be covered, such as document design, usability and audience profiling. There will also be reviews of books and software from time to time.

In this inaugural issue, we kick off with a consideration of why good writing matters. We often hear the retort that no one reads documentation, with the often unsubtle implication that fussing about quality technical writing is a waste of time.

But people do read documentation, although not as they read a novel. They dip into it when they need to, and they mostly need to when they are frustrated with the product they are trying to use. Mostly they would rather be doing something else, a fact that should colour how technical writers write if their role is to be the users' advocate, as it should be. And it follows from this that there is an ethical dimension that informs, or should inform, how technical writers write. The first article in this issue looks at aspects of this ethical dimension.

The next issue will carry a special feature on the Macquarie Dictionary, looking at how language usage is collected and assessed, and how you can contribute content to the teams of lexicographers who labour behind the scenes to ensure the longevity of a distinctly Australian English. (See page 5 of this issue for a generous offer from Macquarie Dictionary Online: a free three-month subscription.)

*Words* is for sharing information, and we are happy to share useful information whatever the source. So if you would like to contribute articles (or even suggest articles) feel free to make contact. Indeed, all comments are welcome. We will endeavour to publish all the correspondence we receive, so long as it is relevant to a topic that has been discussed in an issue of *Words*.

**Geoffrey Marnell**

Editor [geoffrey@abelard.com.au]

## Contents

Why good writing matters .....	1
Getting metadata into a PDF .....	3
Creating custom web styles with Dreamweaver CS4 ....	5
Client profile.....	8
Quotation marks: over-used and under-done .....	9
Localising documents with FrameMaker (Part 1) .....	11
Miscellany .....	14
Fun for numerophiles.....	15

## Why good writing matters

Why should technical writers fuss over the quality of their documentation? It's just plain old instructional writing for most part, and no one expects that sort of writing to be as polished as, say, a novel, surely?

### Professional ethics

For a start, good technical writing is writing that is easily, effortlessly, *understood*. And *understanding* is important for a number of reasons. First, we want the sick to *understand* the documentation that accompanies their medicines, just as we want the machinist to *understand* how to operate the machine they have to use and not injure themselves and others around them. It is difficult to argue otherwise, for this inherent respect for the reader of our documentation—this respect for their safety and wellbeing—goes to the very heart of our professional ethics. No, it goes to the very heart of ethics itself.

### Ethics pure and simple

Second, how we write can affect how others are able to spend their time. We all have a finite number of hours in a lifetime, and we all want—and need—time away from the office: time to spend with our friends and family, and to engage in activities that give our lives richness and significance. So every additional hour someone spends struggling to decipher the meaning of a document their job requires them to understand is, in this era of immovable deadlines, likely to be an hour of personal time lost (for no doubt the struggle, or the work thereby usurped, will continue after-hours and

perhaps on weekends). They would not have lost that hour if the meaning of the document had been crystal clear. Put another way, if we, as technical writers, do not strive to achieve maximum communicative efficiency—by writing in ways that impart our meaning with the least effort and time—then we may be *depriving* our readers of time that they could better spend. And deprivation of freedom is undoubtedly an ethical issue. Indeed, this type of deprivation can be considered a form of theft: theft of another's time.

Third, a person's self-esteem, and general mental wellbeing, can be affected by the degree of confidence they have in their ability to understand material they need to understand. For example, a young apprentice having difficulty understanding operating instructions may give up their chosen trade because of a mistaken belief that they don't have the cognitive wherewithal to succeed in it.

Mistaken? If their difficulty arises not from poor language skills on their part but from poor language skills (or laziness) on the part of the writer of those instructions, then the apprentice's belief that they are not up to it may very well be mistaken.

So writers have the potential to shatter the confidence of readers—not all readers, surely, but certainly some—and, again, this is an ethical issue.

## The law

Fourth, measures of *understandability* (also known as *readability*) are gathering popularity, and in places that matter:

"Today, reading experts use ... formulas as standards for readability. They are widely used in education, publishing, business, health care, the military, and industry. Courts [in the USA] accept their use in testimony."<sup>1</sup>

Organisations have been successfully sued by plaintiffs claiming that they have been disadvantaged by an inability to understand certain public documents. To protect themselves against such litigation, many bodies—commercial and government—now have specific guidelines on the minimum level of readability required of their public documents. For example, numerous US insurance companies demand this of their customer insurance policies, and the US Federal Drug Administration demands a readability level no greater than that to be expected of an eighth-grader on medicinal labels and on medical consent forms. The National Cancer Institute and the Office of Human Research Protection (both in the US) make similar demands on literature destined for public consumption.

So, if we are working on documentation to accompany a product that will be exported to the US, we may have to subject our documentation to readability tests in order to ensure that it can legally be imported into the US.

Further, recent changes to tort law in Europe mean that organisations can be held liable for faulty documentation (that is, documentation difficult to understand) just as they can for faults in the product that it accompanies.<sup>2</sup>

So if readability itself is important, going as it does to the very heart of our professional ethics, and if ensuring a minimum readability is increasingly becoming a legal requirement, then writing well should be a fundamental concern of all technical writers.

### Geoffrey Marnell

Geoffrey Marnell teaches technical writing and editing in the English Department at the University of Melbourne. He is also the founder and managing director of Abelard Consulting.

1. W DuBay, *Smart Language*, Impact Information, Costa Mesa (USA), 2007, p. 5.

2. S Burton, "A Worldwide Phenomenon", *Intercom*, Sept./Oct. 2007, p. 3.

Don't miss this great professional development and networking opportunity!

**20th to 22nd May 2009  
Melbourne, Australia**



20 Sessions... 12 Expert Speakers... Pre-Conference Workshops... Vendor Exhibits... Networking...

Wikis, User-Centered Design, eLearning, DITA, Structured Documents, Case Studies, more...

Keep your skills up-to-date, and learn new techniques, tips and technologies. If you missed a previous AODC, don't make the same mistake in 2009!

The biggest annual event for technical communicators from Australia and New Zealand

Register now at [www.aodc.com.au](http://www.aodc.com.au)

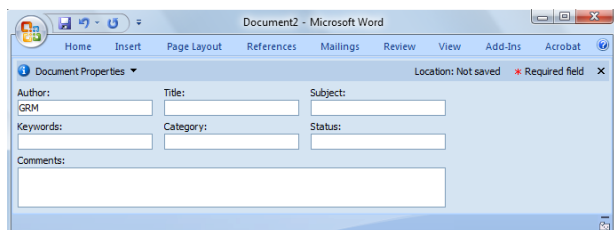
Proudly presented by



# Getting metadata into a PDF document

Most authoring tools these days offer a means of recording *metadata* inside the document you are authoring. Metadata is data about the document. It can include the name of the author, the title of the document, the subject the document is about, keywords for indexing, a copyright notice, and so on.

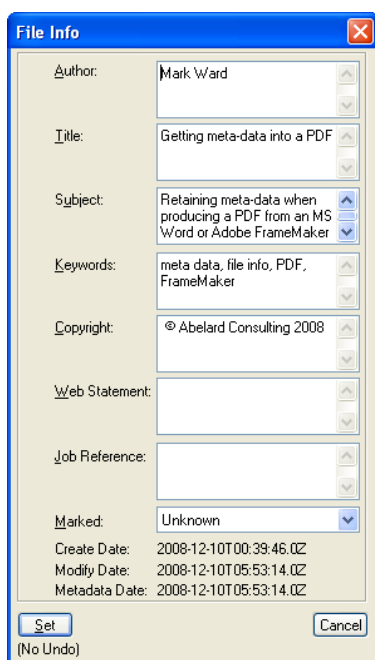
To enter metadata in Microsoft Word 2007, click the Microsoft Office button (top left corner of the window) and then select **Prepare > Properties**. A panel appears above your document for you to enter information about the document (see below).



In Adobe FrameMaker, metadata is entered in the **File Info** window, displayed by selecting **File > File Info**. An example is shown at the right.

There are a number of reasons why you might want to add metadata to your document, and one important one is that many search engines will record some of the metadata they find when they index an internet or intranet site. And the more of your document that is indexed, the more likely it is that an online search will find it.

Because a PDF document can be opened on any computer whatever its operating system, PDF is an ideal format for documents destined for the web (or for an intranet in an organisation with diverse computing equipment). Metadata can also be added to a PDF file. You can do this manually (if you have the full Adobe Acrobat product, not just Adobe Reader). But this would be a time-consuming task if your PDF documents are updated often. It would be far better if the metadata you entered in the source document—in Microsoft Word, Adobe FrameMaker and so on—appeared automatically in a PDF document generated from that source.



## Microsoft Word

There are three main ways to create a PDF from a Microsoft Word 2007 document:

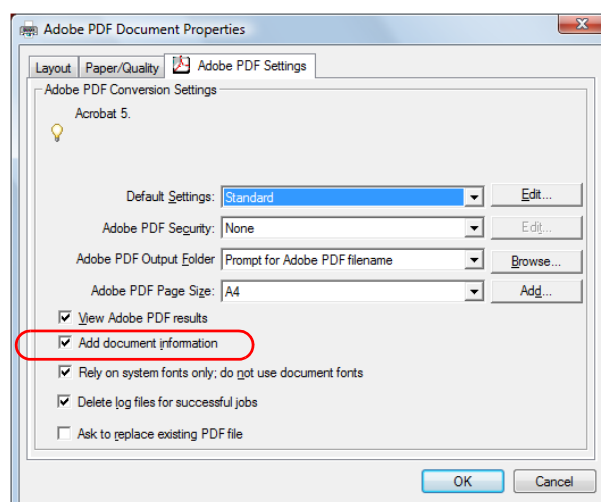
- by choosing the **Adobe PDF** printer driver from the **Print** window
- by choosing to save your document as a PDF file
- by selecting an option from the **Acrobat** menu

These options are not available unless you have installed separate PDF-creation software. The discussion that follows assumes that you have installed the latest version of Adobe Acrobat (version 9). The process may be slightly different with earlier versions.

## Create a PDF by printing

When you install Adobe Acrobat, a special printer driver is also installed. This driver—called **Adobe PDF**—launches Adobe Distiller, the utility that does the work of generating a PDF. The **Adobe PDF** printer driver enables you to generate a PDF document by printing your source document just as you print a document to an external printer. But instead of printing the document on a printer, the **Adobe PDF** printer driver generates a PDF document.

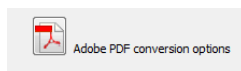
However, this method of generating a PDF does not copy metadata from the source document to the PDF. Even if you select the option to add document information when you are specifying your print options (see figure below), your metadata is not copied to the resulting PDF document. You will get some default information—such as the source document name and author—but not every item of metadata you specified. You will have to add this information yourself, using Adobe Acrobat or some other PDF editor.



## Create a PDF by saving as a PDF

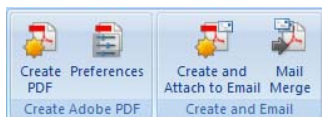
In Word 2007, you can click the Microsoft Office button and then select **Save As > Adobe PDF**.

On the **Save Adobe PDF File As** window, click **Adobe PDF conversion options** and select **Convert document information**. When you proceed to save the file as a PDF, all the metadata is copied to the resulting PDF.



### Create a PDF from the Acrobat menu

When you install Adobe Acrobat, a new menu—titled **Acrobat**—is added to Microsoft Office products. When you select this option, the PDF ribbon appears.



Choose **Preferences** and make sure that the **Convert document information** option is selected.

When you proceed to create the PDF file, all the metadata in the Word document is copied to it.

### Summary

To ensure that all the metadata recorded in your Microsoft Word 2007 document is copied to a PDF version of the document, choose the **Save As** option or an option from the **Acrobat** ribbon.

### Adobe FrameMaker

There are two main ways to create a PDF from a FrameMaker document:

- by choosing the **Adobe PDF** printer driver from the **Print** window
- by choosing to save your document as a PDF file

### Create a PDF by printing

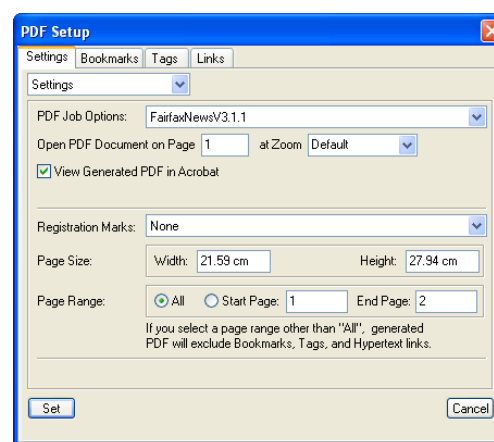
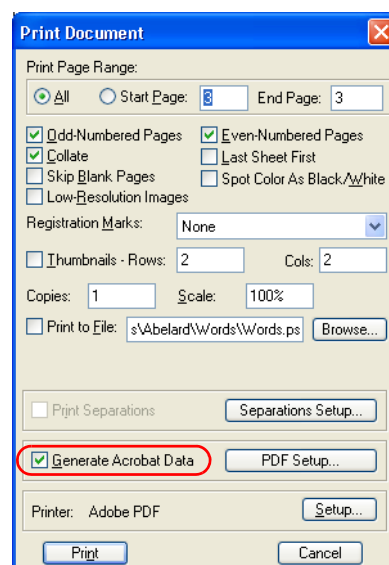
If you create PDFs by selecting **File > Print** and choosing the **Adobe PDF** printer driver, your metadata is copied to the resulting PDF only if you also select:

- **Generate Acrobat Data** in the **Print Document** window (see figure at top right) and
- **Add Document Information** in the **Adobe PDF Document Properties** window.

### Create a PDF by saving as a PDF

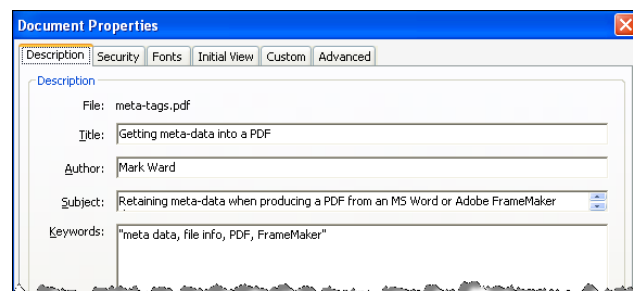
This method of creating a PDF from FrameMaker always copies the metadata from the **File Info** window to the resulting PDF.

1. Select **File > Save As PDF**. The **Save As** window appears.
2. Specify a name and location for the file and click **OK**. The **PDF Setup** window appears (see figure at right).



3. Choose a **PDF Job Options** setting and any other settings you need and click **Set**. The PDF is created.

With the PDF open, select **File > Properties**. The **Document Properties** window opens. On the **Description** tab, the metadata you entered in FrameMaker will be displayed (see below).



### Mark Ward

Mark Ward is an independent documentation consultant, adviser to Abelard Consulting, and delivers Abelard Consulting's *Adobe FrameMaker* and *Structured Authoring with Adobe FrameMaker* training courses.



# Creating custom web styles with Dreamweaver CS4

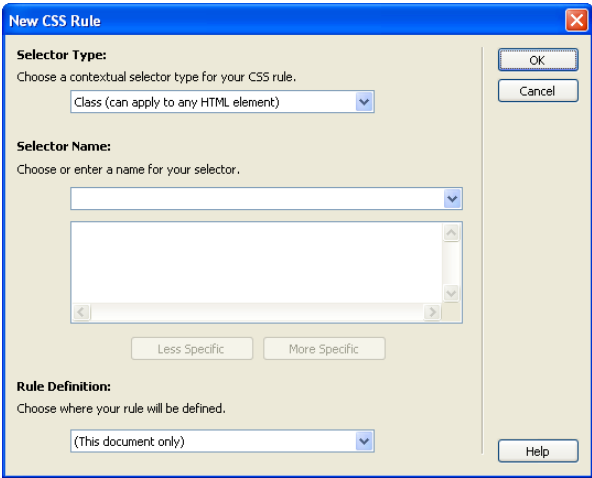
Without any explicit styling, the appearance of the contents of a web page—fonts, font size and so on—is determined by the default settings of your browser. The default appearance in most browsers is not especially inspiring, so you will no doubt want to modify it.

You can change the appearance of:

- every instance of an HTML element
- a particular instance of an HTML element
- a selected word or string of words within an HTML element

Changes made to the appearance of an HTML element can be limited to just the page you are working on, or applied to a set of web pages. In the first case, the formatting specifications are embedded in the page you are working on; in the latter case, the formatting specifications are added to an external document called a *cascading style sheet*. Each page that is to adopt the formatting specifications is then linked to that cascading style sheet.

## Reformatting an HTML element

1. With the HTML file open in Dreamweaver, place your cursor between the opening and closing tags of the element you want to reformat. For example, if you want to reformat all <h1> content, place your cursor inside a set of <h1> and </h1> tags.
  2. Select **Format > CSS Styles > New**. The **New CSS Rule** window appears.
- 
3. From the **Selector Type** list, choose **Tag (redefines an HTML element)**. The HTML element enclosing your cursor appears in the **Selector Name** field. (If it is some other HTML element you want to reformat, you can choose it from the menu of elements that appears if you click the arrow at end of the **Selector Name** field.)
  4. You now need to choose whether the new style is:
    - ◆ to apply just to the current page or



**MACQUARIE  
DICTIONARY  
ONLINE**

[www.macquariedictionary.com.au](http://www.macquariedictionary.com.au)

Subscribe to the complete Macquarie Dictionary online, updated annually with new words and definitions.

Also available online is the full Macquarie Thesaurus - that perfect word is just a click away.

**Try it out now for FREE!**

Macquarie Online is offering free extended trial access. Simply contact Macquarie Online to set up your 3 months free access. Quote code: 3mfTrialAC

Macquarie Online Support  
phone: 1800 645 349  
email: [support@macquarieonline.com.au](mailto:support@macquarieonline.com.au)



*Australia's national dictionary*



commercial  
translation  
centre

**For Effective  
Translation/Localisation  
Of Technical Documentation  
In 80 Languages**

Australia Wide Service Since 1988  
Call 1800 655 224 (Australia Toll Free)  
Email: [mail@ctc.com.au](mailto:mail@ctc.com.au)

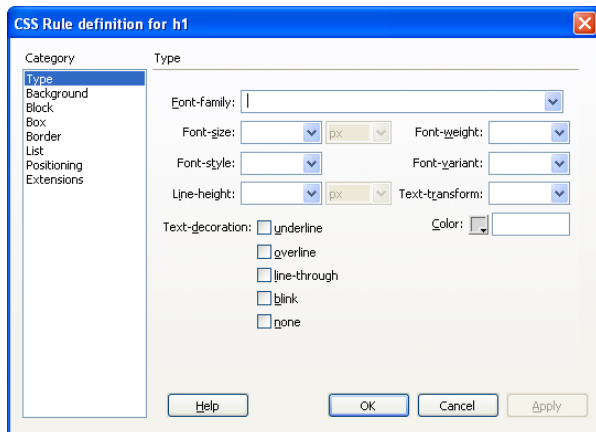
commercial translation centre

- ◆ added to a cascading style sheet (and thus apply to whatever web page you link to that cascading style sheet)

From the **Rule Definition** list, choose either:

- ◆ **(This document only)** or
- ◆ **(New Style Sheet File)**

For the sake of this demonstration, select **(This document only)** and click **OK**. The **CSS Rule definition for [the HTML element you chose]** window appears. In the following example, the `<h1>` element was chosen.



5. Specify the attributes you want the chosen HTML element to have (such as a particular font family, size, colour, bolding and so on).

**NOTE:** Any attribute you don't explicitly specify will retain its default appearance when the element is displayed in a browser.

6. Click **OK**. Note that the new format is immediately applied to all HTML elements of the type you chose.
7. Scroll to the top of the code panel and note that a new block of code has been added: the `<style>` element. An example is shown below.

```
<style type="text/css">
<!--
h1 {
  font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
  font-size: 24px;
  font-weight: bold;
  color: #900;
}
-->
</style>
```

The fact that this style specification has been added to the current page means that it affects only the current page. In the example above, all content tagged `<h1>` in the current document will be restyled according to style specification.

## Cascading style sheets

Let's now create a cascading style sheet so that a restyling can be applied to several HTML pages automatically.

1. Place your cursor between the opening and closing tags of the element you want to reformat.
2. Select **Format > CSS Styles > New**. The **New CSS Rule** window appears.
3. From the **Selector Type** list, choose **Tag (redefines an HTML element)**. The HTML element enclosing your cursor appears in the **Selector Name** field.
4. From the **Rule Definition** list, choose **(New Style Sheet File)** and click **OK**. The **Save Style Sheet File As** window appears.
5. Enter a name for the cascading style sheet and specify a location for it. When you click **OK**, the **CSS Rule definition for [the HTML element you chose]** window appears.
6. Specify the attributes you want the chosen HTML element to have (such as a particular font family, size, colour, bolding and so on).
7. Click **OK**. Note that the new format is immediately applied to all HTML elements of the type you chose.
8. Scroll to the top of the code panel and note that a new line of code has been added between the `<head>` and `</head>` tags. An example is shown below:

```
<link href="trial.css" rel="stylesheet" type="text/css" />
```

This line of code tells a web browser to use the style definitions in an external file (called here `trial.css`). If there are HTML elements in the document without a style definition in the referred-to cascading style sheet, the browser will use its default formatting.

**NOTE:** To have another web page appear according to the style specification in the cascading style sheet, you need to copy this line of code to that page, placing it anywhere between the `<head>` and `</head>` tags.

## Reformatting instances of an HTML element

You may have need to change the appearance of particular instances of a standard HTML element. For example, you might want a version of the `<p>` element for quotations, styled so that the text is indented and the font slightly smaller than the standard `<p>` style. You can achieve this by defining a style *class*.

**TIP:** Once defined, a class can be applied to any HTML element. That element will then take on the styling specified in the class definition.



1. With the HTML file open in Dreamweaver, select **Format > CSS Styles > New**. The **New CSS Rule** window appears.

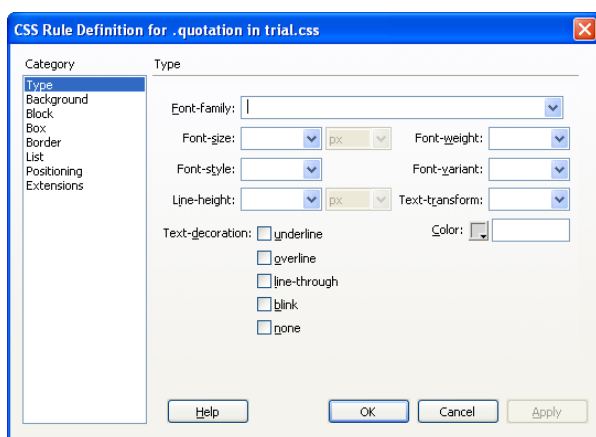
In this case, it doesn't matter where your cursor is. This is because you are not actually modifying an element, but creating a style class that can be applied to *any* element.

2. From the **Selector Type** list, choose **Class (can apply to any HTML element)**.
3. Tab to the **Selector Name** field and enter a name for the class. Enter any name you like (but to help you remember its purpose, give it a name that suggests its purpose: quotations, small\_type, body\_indented and so on).
4. Now choose whether the new style is:
  - ◆ to apply just to the current page or
  - ◆ added to a cascading style sheet (and thus apply to whatever web page you link to that cascading style sheet)

From the **Rule Definition** list, choose:

- ◆ **(This document only)**
- ◆ **(New Style Sheet File) or**
- ◆ the cascading style sheet you created in the previous exercise

In this exercise, select the cascading style sheet you created in the previous exercise and click **OK**. The **CSS Rule definition for [class name in .css file]** window appears. In the following example, a class called **quotation** is to be created and saved in `trial.css`.



Note that Dreamweaver gives class names a leading full stop: **.quotation** in our example. This will help distinguish default HTML elements from classes should you wish to call a class by the same name as a standard HTML element.

5. Specify the attributes you want the class to have (such as a particular font family, size, colour, bolding and so on).

6. Click **OK**. Note that nothing changes in your web page. All we have done so far is added a class definition to a cascading sheet. We now have to reference that class in the code for the web page we want to change.

## Applying a class to an element

A class is applied by making it an *attribute* of an HTML tag, just as **href** is an attribute of the `<a>` tag and **src** is an attribute of the `<img>` tag. You specify a class in just the same way as you specify any other attribute:

```
class = "[name of class]"
```

1. In the code panel, place your cursor inside the opening tag of an element whose appearance you want modified by the class you just created. Place the cursor just before the *closing* bracket.
2. Press the **SPACEBAR** and note that a menu of attributes relevant to the HTML element appears.
3. Double-click **class** in the list of attributes. Note that the cursor is now automatically positioned between a pair of inverted commas and a menu of classes appears.
4. Double-click the name of the class. It is placed within the inverted commas. The tag will now look something like this:

```
<p class="quotation">
```

**TIP:** You can also type the attribute and its value directly from the keyboard instead of selecting them from menus.

5. Refresh the design panel by pressing **F5**. Then view your page in your default browser by pressing **F12** (and saving the file in the process). Notice how your new class style modifies the appearance of the text within the tags to which the class was applied. Close the browser.
6. Add the class attribute to other elements in your document and note the result. Note too that the class can be added to *any* HTML element.

## Reformatting text within an HTML element

In the cases we have just considered, we have explored ways of modifying the appearance of *all* the text within an element. However, sometimes you will want to modify the appearance of just some of the text within an element. The `<strong>` and `<em>` elements only give you bold and italic formatting, but with the `<span>` element you can style characters, words and strings of words in any way you like.

1. Create a new class called **BoldRed** and leave the **Font-size** field on the **CSS Rule Definition...** window blank. (Leaving a field on this window

blank means that this particular characteristic will not change when the class is applied. In other words, the size of the font will remain unchanged when the class is applied.)

If necessary, follow the steps in “Reformatting instances of an HTML element” on page 6. You can add the new class to the cascading style sheet you created earlier.

Make sure that the **Font-weight** field is bold and that the **Color** field is some form of red. Choose other characteristics as you please.

2. With the **BoldRed** class defined, place your cursor immediately to the *left* of the text you want to reformat according to the **BoldRed** style specification. Do this in the code panel, not the design panel.
3. Enter < (that is, the opening bracket for an HTML tag). Note that a menu of tags appears.
4. Scroll through the menu of tags until you see **span** and then double-click it.
5. Press the SPACEBAR. Note that a menu of relevant attributes appears.

6. Double-click **class**. Note that the cursor is now automatically positioned between a pair of inverted commas and a menu of classes appears.
7. Double-click **BoldRed** and enter > (that is, the closing bracket for an HTML tag).
8. Place your cursor immediately to the right of the *last* character in the text that you want to have the **BoldRed** style.
9. Enter < / (that is, the opening bracket for an HTML tag followed immediately by the forward slash).
10. Refresh the design panel by pressing F5. Then view your page in your default browser by pressing F12 (and saving the file in the process).

Dreamweaver automatically adds whatever text is needed to complete the end tag (which should now read </span>).

Notice how your new style class modifies the appearance of the text between the SPAN tags. Close the browser.

**Geoffrey Marnell**

## Client profile: Hunter Valley Research Foundation



*Base your decisions on reliable evidence, not guesswork*

Australia's longest-serving regional research organisation, the Hunter Valley Research Foundation has provided independent, reliable economic, demographic and social information to guide decision-makers for more than 50 years.

We welcome your enquiries.  
Contact our free information service  
(02) 4969 4566 (12-3pm Mon to Fri)

**[www.hvrf.com.au](http://www.hvrf.com.au)**





# Quotation marks: over-used and under-done

Quotation marks (aka *inverted commas*) are put to a great many uses, although most of these can be taken care of by other means. With good reason, many language manuals recommend reserving quotation marks for marking out quoted speech or text.

## Mark out quoted text

The primary use of quotation marks is to enclose or mark out words spoken or written by someone:

The doctor said, "You really must stop drinking so much".

As Ayer wrote, "Only if a statement can be verified does it have meaning".

Where the quoted material straddles more than one paragraph, a quotation mark is commonly placed:

- at the start of the quote
- at the start of each subsequently quoted paragraph and
- at the end of the quote

In other words, no quotation mark is needed at the end of a paragraph other than the very last paragraph quoted.

## Quotation marks in references

A well-established practice is to enclose in quotation marks the titles of:

- a chapter of a book
- an article in a journal or newspaper
- an essay
- a lecture
- an unpublished document

## Other uses

According to *Style manual*, that trusty Government-sponsored guide to Australian English, quotation marks can also be used:

- for a technical term in a non-technical document  
The exploding star eventually becomes a "black hole".
- for a word or phrase that the writer has invented  
This government has infected the economy with a "tax cancer".
- to show irony  
Their "policy" on the environment never looked like winning support.<sup>3</sup>

However, other language manuals recommend that quotation marks not be used in these ways:

"Using *quote marks* to highlight words for such a range of different purposes is not ideal. The *Chicago Manual of Style* (2003) comments that it is irritating if overused. In any case *quote marks* cannot be relied on to express a particular attitude or form of irony. Ideally the intended emphasis or meaning is conveyed by the choice of words, appropriately arranged. If something is still needed for the individual word, you can resort to bold or italic type. Bold and italics are also the answer for technical or foreign terms ... Quote marks can then be reserved purely for quoted material, and for translations ... of foreign words".<sup>4</sup>

It is certainly the case that the more uses allowed of a punctuation mark the more likely it is that

there will be ambiguity. Look back at the three sample sentences given above. Without the introduction to each, is it immediately clear what the writer intended by the quotation marks?

Are they being used to quote, to introduce a technical term, to indicate that the term is invented or to indicate irony? No; the reader will need to work this out, if that is indeed possible in any particular case.

Quotation marks only became a regular feature of the English language in the late eighteenth century.

On Peters's advice, we should rewrite the three sample sentences above, avoiding the use of quotation marks. Here are some possibilities:

The exploding star eventually becomes what scientists call a black hole.

This government has infected the economy with what could be called a tax cancer.

Their so-called policy on the environment never looked like winning support.

Now the meaning of each sentence clearly matches the writer's intent. The reader no longer has to work out what particular use the writer has put the quotation marks to. And we have achieved this without recourse to bold or italics (the purpose of which the reader might also need to work out).

## Spelling out technical terms

The practice of putting a technical term in quotation marks in a non-technical document is common but rather odd. Consider again the black-hole example introduced in the previous section:

The exploding star eventually becomes a "black hole".

To make the writer's intention clear, we rewrote the sentence as:

“ ? ”

3. *Style manual for authors, editors and printers*, 6th edn, John Wiley & Son, Canberra, 2002, p. 114.

4. P Peters, *The Cambridge guide to Australian English usage*, Cambridge University Press, Cambridge (UK), 2007, p. 670.

The exploding star eventually becomes what scientists call a black hole.

But does this really help the reader? Informing them that this is a technical or specialist term—a jargon term—does not help them understand what the term means. And this is important given that the document is addressed to a non-specialist audience. (If the intended audience were, say, astrophysicists, there would be no necessity to use quotation marks.) It is far better not to waste words telling the reader that this is a term scientists or astrophysicists use—which is probably of no interest to them—but rather to *explain* what the term means:

The exploding star eventually becomes a black hole, that is, a body with such a strong gravitational force that no light can escape from it.

### Quotation marks and block quotation

Most language manuals will tell you that quotation marks are not needed around a *block quotation*, that is, a quotation set apart and set differently from the text that introduces it.

A block quotation is often indented from the left and often in a different font (or a smaller font size) than body text.

However, in a document where you have material other than quotations that you want to set similarly—such as examples—you need to make it clear which blocks are quotation blocks and which are not. You could do this with typography: a different font, different indent, and the like. You could also do it by making sure that each quote had a citation. However, readers might not discern the significance of the differing typographical styles, and you might not want to include a citation for every quotation you include. Also, readers shouldn't have to wait until the end of a paragraph to learn that what they are reading is a quote rather than something else (which, for a lengthy passage, might not be apparent until the page is turned).

**Recommendation:** it will help the reader detect immediately that a block of text is a quote and not something else if you enclose it in quotation marks. No meaning is lost if quotation marks are used, or omitted; so this is case of discretionary punctuation. But from a readability perspective, quotation marks can be useful, especially where blocks of text with different purposes are styled similarly.

### Lazy quotes

Some writers use quotation marks to indicate that the words enclosed within them *almost* express the meaning they have in mind but are not to be taken literally.

Here are two examples:

the current 'seeped' through the substrate  
a record 'holds' data from each field

Scientific and technical writing aims for precision. (Indeed, that should be a goal of all factual writing.) The advancement of scientific and practical knowledge is not helped if its main vehicle—written words—is peppered with expressions that are not meant to be taken literally, for such non-literal expressions are literally imprecise.

Moreover, the writer who uses what might be called *near-enough* quotation marks is expecting the *reader* to work out what the real meaning is. Rather than using precise words that would get the message across effortlessly, the writer is lazily using words that will require unnecessary effort on the part of the reader as

they try to work out what exactly the writer had in mind. Hence, a better term for these quotations marks is *lazy quotes*.

**Recommendation:** a good writer—that is, a writer who writes so as to be understood with the least effort—shuns lazy quotes. Instead, they attempt to find the right word

to use and, failing that, use a near-synonym and explain the particular meaning they have in mind:

- × everything in the 'universe'
- ✓ everything in the universe (understood to mean everything that we can currently detect)

### Single or double?

In American English, quotations are almost always enclosed in double quotation marks:

Jackie said, "That is a difficult question."

In British and Australian English, both single and double quotations are widely used. *Style manual* recommends single quotations marks—soley because it is "in keeping with the trend towards minimal punctuation"<sup>5</sup>—but this may not match majority local practice.<sup>6</sup>

A readability issue arises if you use single quotation marks. This is due to the fact that the closing single quotation mark and the apostrophe are identical. Hence, if you have a plural possessive apostrophe within a quotation enclosed within single quotation marks, there is some chance that the reader will mistakenly think that the apostrophe is indicating the end of the quotation.

Consider the following sentence fragment:

And then Lesley said, 'I'm tired of all the academics' fighting to be heard ...



5. op. cit. p. 112.

6. Peters op. cit. p. 670

When the reader encounters the word *academics'* it is not immediately clear whether the punctuation mark at the end of it is a possessive apostrophe or a closing quotation mark. And it may not be until the end of the sentence that the reader eventually works out which it is:

And then Lesley said, 'I'm tired of all the academics' fighting to be heard above the noise. [Here, Lesley is fighting to be heard above the noise.]

And then Lesley said, 'I'm tired of all the academics' fighting to be heard above the noise'. [In this case, Lesley said the words *fighting to be heard above the noise*.]

Here is an example from *The Age* newspaper, reproduced exactly as it was printed:

... would accept anything as legitimate scholarship as long as it sounded good and "flattered the editors' ideological preconceptions".<sup>7</sup>

Note that if *The Age* hadn't used double quotes, readers would have encountered 'flattered the editors' at the end of a line and many, no doubt, would have interpreted the possessive apostrophe as the end of the quotation. In the context, those three words—*flattered the editors*—make a sensible quotation. Only when those readers shift their eyes to the start of the next line would they realise that something is amiss. They will either ignore or reread what they have just read; that is, there will either be communication breakdown or inefficient communication, both undesirable outcomes and all because of a poor choice of quotation marks.

---

7. *The Age*, 8 January 2009, p. 11.

**Recommendation:** if you have a choice, use double quotes. If you must use single quotes, avoid using them for any other purpose, and be alert to possessive apostrophes that might confuse the reader. You can't rewrite the sentence to remove that confusion—since you are quoting—but you need to make your meaning obvious by supplying additional text or rearranging the sentence. For example, if *academics'* is the end of the quotation in the example above, the sentence would be better written as:

And then, fighting to be heard above the noise, Lesley said, 'I'm tired of all the academics'.

### Quotes within quotes

If you opt to use single marks for quotations, then any quotation within a quotation should be enclosed within double quotation marks. Alternatively, if you opt to use double marks for quotations, then any quotation within a quotation should be enclosed within single quotation marks.

According to Krutz, "what Hegel really said was 'societies are continuously changing' and this has been misinterpreted ever since".

This is one case where you are free to change the punctuation within a quotation: the author you are quoting may have used double quotes inside the text you are quoting whereas your system, if it is to be applied consistently, calls for single quotes. If you are quoting several authors and their quotation practices differ, obviously the quotation practice of at least one author will need to be changed.

**Geoffrey Marnell**

## Localising documents with Adobe FrameMaker (Part 1)

Most, perhaps all, versions of Adobe FrameMaker have provided ways for authors to customise a file or set of files so that a number of versions can be created from that file or set of files. This is one form of *single sourcing*—that is, generating many deliverables from one source.

Single sourcing is particularly useful where you need to localise a document. For example, you might be working for a company that sells a product into a number of overseas markets, and each market uses different terminology, different spellings, different date formats and so on.

For example, what Australians call a *supplier*, Americans call a *vendor*. We talk of *anticlockwise* movement, whereas the Americans would call this *counterclockwise* movement. We call it *petrol*; the Americans call it *gas*. Our *aluminium* is their *aluminum*. Our *public holidays* are American *national holidays* and British *bank holidays*. We walk on

*footpaths*, Americans on *sidewalks* and the British on *pavements*. And so on and so on.

The linguistic variances between the different English languages are quite substantial. There are even regional differences in Australian English. What most Australians call a *power pole*, South Australians call a *stobie pole*.

If you have to localise your documentation for two audiences each of which adopts a distinct English—American English, Australian English, British English and so on—you could always create two sets of documentation: one for each audience. This would be a tedious approach, especially when it came to maintenance (that is, producing a new version of the documentation to match a new version of the accompanying product). All the common material would have to be duplicated in each documentation set.



---

**Localisation:** making user documentation appear as if it was written by a member of the intended audience for that audience

---

For two audiences, this might not be too much of a burden. But for three or more audiences, copying common material across multiple documentation sets is simply not the most efficient approach. A far better approach is single sourcing.

## Single-sourcing features

Adobe FrameMaker provides two features to help you localise your deliverables from one set of files:

- variables and
- conditional text

### Variables

A variable is just a name that stands in for a character, word or string of words. When you want to enter that character, word or string of words, you just insert the variable instead. You can do this throughout the entire documentation. Then when you need to change the character, word or string of words, you simply change the character, word or string of words associated with the variable. Every instance of that variable throughout the entire documentation is then automatically updated.

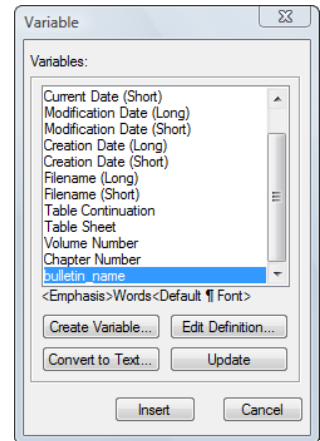
Suppose you need to produce a user guide for an Australian audience and another user guide (for the same product) for an American audience. You want to minimise your effort by generating both user guides from one set of FrameMaker files.

Suppose further that your user guides will often refer to the fuel used to power a car: *petrol* in Australia and *gas* in America. With a single-sourcing methodology, you would create a variable—called, perhaps, *fuel*, although you can call it anything you like—and, when you need to insert the word *petrol* or *gas*, you insert the variable *fuel* instead. Whatever text is currently associated with that variable is automatically inserted. If you were preparing the Australian version, that text would be *petrol*. When you come to produce the American version, you just override *petrol* with *gas* as the definition of the variable *fuel*. All instances of the variable *fuel* immediately change from *petrol* to *gas*.

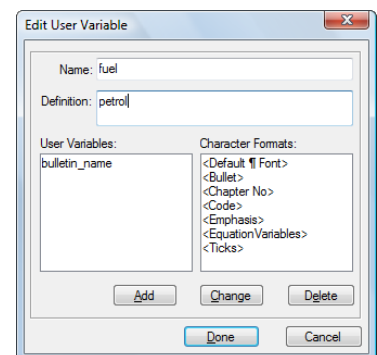
Here is how it's done in FrameMaker.

### Step 1: Creating a variable

1. Select **Special > Variable**. The **Variable** window appears.
2. Click **Create Variable**.
3. In the **Edit User Variable** window, enter a name for the variable. This can be any name you like, but give it a name that will help you remember its purpose (such as *fuel* in this example).



4. In the **Definition** field, enter one of the values this variable will take; for example, *petrol*.
5. Click **Add**, **Done** and **Done** again.



The variable is now available for you to insert in your text whenever you need to specify the substance that fuels a car.

### Step 2: Inserting a variable

When you need to specify in your document the substance that fuels a car, you simply insert the variable, not the substance:

1. Select **Special > Variable**. The **Variable** window appears (shown above).
2. Click once on the name of the variable (*fuel* in our example) in the list of variables.
3. Click **Insert**. The value (that is, the definition) of that variable—*petrol* in our example—is inserted at the location of the cursor.



## In the next issue

- Font choice: Colin Wheildon's research revisited
- Entering uncommon characters as unicode
- Localisation using conditional text
- Behind the scenes at Macquarie Dictionary Online by Sue Butler (publisher of the Macquarie Dictionary)
- Unnecessary singular-plural composites ... and more

**austechwriter**

The online discussion forum for local technical writers

Unmoderated and ad-free

- ◀ To read previous postings, visit [www.freelists.org/archives/austechwriter/](http://www.freelists.org/archives/austechwriter/)
- ◀ To subscribe, send a blank email to [austechwriter-request@freelists.org](mailto:austechwriter-request@freelists.org) with SUBSCRIBE as the subject
- ◀ For more information, contact [austechwriter-admins@freelists.org](mailto:austechwriter-admins@freelists.org)

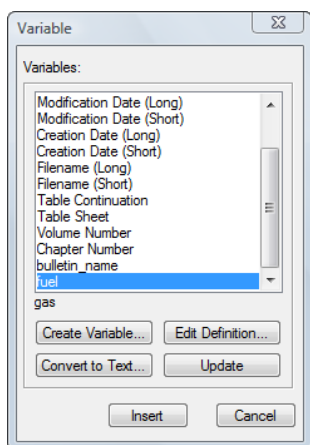


### Step 3: Localising

Suppose now that you need to produce a version of the documentation for the American audience. In this version, all instances of the word *petrol* need to be changed to *gas*. If you had entered *petrol* as direct text rather than as a variable, you would have to search for every instance of *petrol* and replace it with *gas*. This would be a tedious exercise in a large document or set of documents.

However, if *petrol* has been inserted as a variable, all you need to do is change the definition of that variable. All instances of that variable, throughout the entire document, are then automatically changed to the new definition. Here is how it is done:

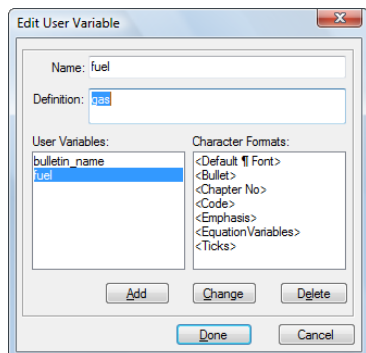
1. Select **Special > Variable**. The **Variable** window appears.
2. Click once on the name of the variable (*fuel* in our example) in the list of variables.
3. Click **Edit Definition**.



4. In the **Edit User Variable** window, change the definition of the variable. In our example, change *petrol* to *gas*.

5. Click **Change**, **Done** and **Done** again.

Every instance of the *fuel* variable you inserted in your document now changes from *petrol* to *gas*.



If you have other documents that require the same localisation, you simply import the variable into those documents: **File > Import > Formats > Variable Definitions**. And you can do this in one step if all those documents are assembled in the one FrameMaker book.

### Conditional text

In the next issue of *Words* we'll look at how *conditional text* can be used to localise documents. With conditional text, you select text, apply a user-defined conditional text tag to it (say **US\_audience\_only**) and then, at production time, choose which tags to show and which to hide. Text so tagged is then shown or

hidden. This is an excellent way of ensuring that a particular audience only gets the material they need to see without you having to delete the material that is not relevant to them, for text tagged as hidden is not deleted. It can be revealed again simply by changing its show-hide settings. More next issue.

**Geoffrey Marnell**

## Style manual

for authors, editors and printers

The seventh edition of this authoritative guide to Australian style and usage will be published in late 2009 or early 2010.

The publisher will be holding focus groups with interested readers to uncover what information is needed in this edition. If you want to be part of such a focus group, send an email to [govstyle@finance.gov.au](mailto:govstyle@finance.gov.au).

## Abelard courses

Coming up in Q1 2009

Abelard Consulting offers public and custom courses in technical writing, business writing, scientific writing and Adobe FrameMaker (structured and unstructured). We also offer half-day intensive courses on punctuation.

In the first three months of 2009, the two-day **technical writing course** will be delivered in:

- Melbourne [February 5 & 6]
- Sydney [February 9 & 10]
- Canberra [February 10 & 11]
- Perth [February 23 & 24]
- Adelaide [February 25 & 26]
- Brisbane [March 2 & 3]
- Auckland [March 23 & 24]
- Geelong [March 30 & 31]

For details of all Abelard courses, visit <http://www.courses.abelard.com.au> or call 1800 601 116.

Early-bird discounts apply



## The Words team

Artwork: Christine Weaver

Copy-editor: Marcia Bascombe

Adviser: Mark Ward

Editor: Geoffrey Marnell

© Abelard Consulting Pty Ltd 2009

## Miscellany

### FrameMaker 9 released

Predictions of the demise of Adobe FrameMaker were rampant not so many years ago. Indeed, the topic had been popping up for a very long time. After the long, static life of version 5.5, Adobe surprised its doubters with relatively quick releases of versions 6, 7 and 8.

And now, less than two years after the release of version 8, version 9 has hit the streets.

Version 9 offers a substantially redesigned user interface, a workspace you can customise (with floating tool bars and dockable palettes), additional DITA tools (to provide full support for the current DITA standard), the ability to import comments from PDF files, and quite a bit more that Miscellany hasn't had a chance to explore.

If your copy of version 8 is part of the Adobe Technical Communication Suite (TCS) version 1, you can upgrade to TCS version 2 and get, in addition to FrameMaker 9, RoboHelp 8, Captivate 4, Photoshop CS4, Acrobat 9 and Presenter 7—all in the one bundle.

To explore further, visit <http://www.adobe.com/ap/products/framemaker/>

### New MS Word 2007 resources

Christine Kent is a long-time educator, writer, editor and user of Microsoft products. After 20 years as a technical and education writer, there's not much she doesn't know about producing a good document in Microsoft Word.

Christine has just released three new books:

- *Enjoy... Microsoft Word 2007*, for newcomers to Word
- *Enjoy... Microsoft Excel 2007*, for newcomers to Excel
- *Enjoy... Upgrading to Microsoft Word 2007*, for users with some skills in previous versions of Word who are finding the new version a challenge

Christine also has a blog devoted to Word 2007 (at <http://christinekent.blogspot.com/>) which is regularly updated with new *how-to* articles, tips and tricks. She is also offering remote and on-site support for newcomers to Word 2007.

To learn more about the full range of her offerings and see a preview of her books, visit <http://christinekent.blinkweb.com/>

## TRAINING COURSES

Technical, general & scientific writing, FrameMaker, Structured authoring

Abelard Consulting is offering a number of courses and workshops designed to benefit technical writers, would-be technical writers, and those whose job requires them to be good communicators. The following courses are available:

- ▶ Technical Writing: An Introduction and Refresher [2 days]
- ▶ Scientific Writing: An Introduction and Refresher [2 days]
- ▶ Using Adobe FrameMaker: An Introduction and Refresher [2 days]
- ▶ Structured Authoring with Adobe FrameMaker [2 days]
- ▶ Business Writing: The Fundamentals of Clear Communication [2 days]
- ▶ Punctuation: An Intensive Overview of Best Practice [4 hours]

#### Where?

Scheduled classes are held in Melbourne, Sydney, Canberra, Perth, Brisbane, Adelaide, Geelong, Ballarat, Christchurch and Auckland. Some instructor-led online courses are planned for 2009.

#### What?

To see the topics covered in a course, download a brochure from [www.courses.abelard.com.au](http://www.courses.abelard.com.au)

#### When?

Download a brochure or visit [www.abelard.com.au/course\\_booking.html](http://www.abelard.com.au/course_booking.html) to see dates and locations.

#### By whom?

The FrameMaker and structured authoring courses are conducted by Mark Ward. Mark is a long-time user of FrameMaker and an expert in creating FrameMaker templates and structured applications.

The writing courses are designed and conducted by Dr Geoffrey Marnell. Geoffrey is the founder and principal consultant of Abelard Consulting. He also teaches Technical Writing and Editing in the English Department at the University of Melbourne and is accredited by IPED (Institute of Professional Editors). Geoffrey has more than 20 years experience as a technical writer, documentation consultant, documentation project manager and educator.

#### Discounts?

Early-bird and student discounts apply. Further, on-site training is charged at a flat rate regardless of the number of participants.

**ABELARD** consulting Pty Ltd

Freephone: 1800 601 116

Telephone: +61 3 9596 3456

Email: [courses@abelard.com.au](mailto:courses@abelard.com.au)

Web: [courses.abelard.com.au](http://courses.abelard.com.au)

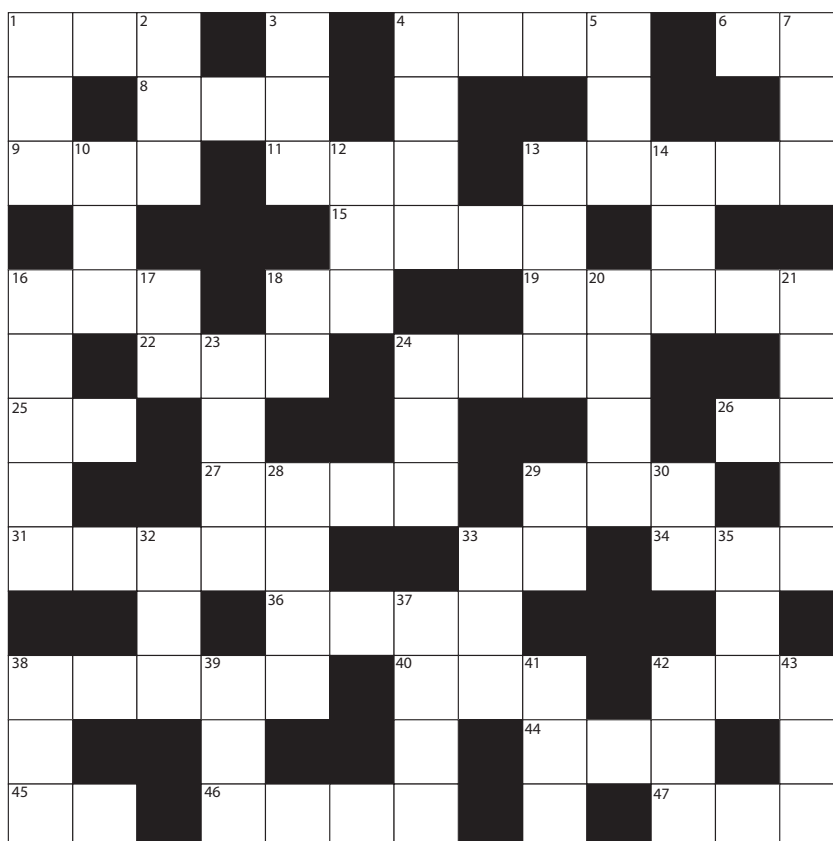
# Fun for numerophiles

In the following puzzle, every answer is a number. The clues, though, are not esoterically mathematical—you don't need any special mathematical training to solve them. You will, however, need to apply a little logic, for some clues do not yield the answer on their own but depend on the answer to another clue, itself dependent on the answer to a further clue, and so on. You will sometimes need to eliminate possible solutions by

trial and error, to run with possibilities until some—maybe many—steps later you discover which fits and which lead to contradiction or absurdity.

NOTE: No answer begins with a zero, all answers are positive, all digits are whole numbers and only one digit per square is allowed.

The solution will appear in the next issue of *Words*.



## ACROSS

- |  |   |
|--|---|
| 1. Equally ascending digits                                  | 26. Factor of 43D                         |
| 4. Cube  | 27. Equally ascending digits              |
| 6. Product of the digits equals the sum of the digits in 16A | 29. Same as 27A                           |
| 8. Anagram of a square                                       | 31. Multiple of 38A                       |
| 9. Same as 8A  | 33. Square                                |
| 11. 12 D less 10D  | 34. Square                                |
| 13. Number followed by its triple                            | 36. Anagram of the 4 unique digits in 38A |
| 15. Consecutive squares                                      | 38. Factor of 31 A                        |
| 16. Square   | 40. Cube                                  |
| 18. Square   | 42. Cube                                  |
| 19. Difference between adjacent digits is 1                  | 44. Cube                                  |
| 22. Cube   | 45. Half of 32D                           |
| 24. Cube   | 46. Multiple of 14D                       |
| 25. Third of 45A   | 47. Reverse of a cube                     |

## DOWN

- |                                   |                              |
|-----------------------------------|------------------------------|
| 1. Square                         | 21. Consecutive primes       |
| 2. Twice 1A                       | 23. Palindrome               |
| 3. Twice 2 D                      | 24. Square                   |
| 4. Quadruple                      | 28. Cube                     |
| 5. Cube                           | 29. Factor of 43D            |
| 7. Reverse of a cube              | 30. Square                   |
| 10. 12D less 11 A                 | 32. Twice 45A                |
| 12. 10D +11A                      | 33. Digits sum to 14         |
| 13. Cube                          | 35. Square                   |
| 14. Square                        | 37. Cube                     |
| 16. Multiple of 36A               | 38. Quadruple                |
| 17. Prime number                  | 39. Cube                     |
| 18. Reverse of a prime number     | 41. Square                   |
| 20. Number followed by its double | 42. Equally ascending digits |
|                                   | 43. Product of 26A and 29D   |