

Numerals and tables

Take just as much care over numerical and tabular content as you would do over text. Use the correct form of numeral for the situation, and design tables to be read, not just looked at.

Use old-style numerals in running text

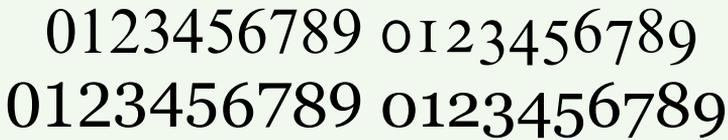
In ‘[Ligatures and abbreviations](#)’ we established that writing systems based on the Latin alphabet, in addition to Greek and Cyrillic, use a bicameral script, with each letter represented by two different forms – uppercase and lower (or majuscule and minuscule, to use more formal terms). The same is true of numerals. We have at our disposal ‘uppercase’ numbers 0123456789 called *lining* or *titling* numerals, and ‘lowercase’ numerals 0123456789 called *old-style* or *text* numerals.

Unlike capital and lowercase letters, different forms of numbers do not convey different meanings; they are, however, an essential component of the typographer’s palette. Just as a string of capital letters in the middle of a sentence SHOUTS at your reader, so numbers set in lining numerals call undue attention to themselves. Are pounds, dollars, dates and quotas really more important than the words and ideas which give them context and meaning?

Treat numbers as you would letters, making sure they don’t stand out unnecessarily. Do this by using old-style numerals in all your running text. Most modern, professional fonts will include both old-style and lining numerals as OpenType features. One or other of these styles will be used for the default numbers. More often it will be the old-style numerals, but there is no strict rule or consistency, and the choice of default is down to the type designer.

It’s also the case that the vast majority of fonts are neither modern nor professional, if modern means OpenType-enabled and professional means designed with both sets of numerals. Take Georgia, for example. Designed

by Matthew Carter in 1993 as a screen font for Microsoft, it is extremely well put together, elegant and appealing, and one of the most popular and widely distributed fonts in the world. But it is not packaged as an OpenType font and so only contains one set of numbers, in this case old-style numerals. Times New Roman, which is similarly widespread but, again, not as an OpenType font, is packaged only with lining numerals. Georgia and Times New Roman are so widely distributed because they are bundled free of charge with Windows and Mac operating systems. However, both these fonts – like many others – are available to buy in professional versions, which do come as OpenType fonts complete with both sets of numerals, small caps and many other features.



Top: Numerals in Times New Roman Pro. *Bottom:* Numerals in Georgia Pro.

To specify old-style numerals, set the `font-variant-numeric` property with a value of `oldstyle-nums`. If most of what you're designing on a page is running text, then your best approach is to set old-style numerals so that they are inherited from the `<body>`.

```
body { font-variant-numeric: oldstyle-nums; }
```

For legacy browsers requiring `font-feature-settings`, use the `onum` OpenType feature tag. As explained in ‘[Ligatures and abbreviations](#)’, you can add an `@supports` rule to cater for legacy browsers that only support `font-feature-settings`:

```
body { font-feature-settings: "onum" 1; }

@supports (font-variant-numeric: oldstyle-nums) {
  body {
    font-feature-settings: normal;
    font-variant-numeric: oldstyle-nums;
  }
}
```

Many sans serif fonts of the early to mid-twentieth century, including Helvetica, were never designed with anything other than lining numerals. This is one of the reasons why Helvetica is rarely your best choice for body text. That said, the lining numerals are less of a problem in Helvetica than they are in some other typefaces.

As we saw in ‘[Designing paragraphs: line spacing](#)’, Helvetica has a large x-height. A consequence of this is that its lowercase letters are closer in stature to its lining numerals when compared to other sans serif fonts such as Futura and Avenir, which have far smaller x-heights.

For nearly 50 years the Swiss type designer Adrian Frutiger, born in 1928, was a hugely influential figure in typography.

For nearly 50 years the Swiss type designer Adrian Frutiger, born in 1928, was a hugely influential figure in typography.

Compared with old-style numerals, lining numerals shout loudly in Avenir.

Clearly Paul Renner and Adrian Frutiger, the designers of Futura and Avenir respectively, recognised the need for old-style numerals in their fonts as both these typefaces were designed with them from the start. Sadly, the versions of Futura and Avenir widely distributed with Apple devices have lining numerals as the default, and do not include old-style numerals as OpenType features (the macOS version of Avenir Next, however, does include them).

Use lining numerals in headings

Old-style numerals are your go-to glyphs for making numbers sit well in running text. For the same reason they are at ease with lowercase letters, so old-style numerals feel uncomfortable in close proximity to capital letters. If you set headings in anything other than sentence case, in particular ALL CAPS, or Title Case, then don't use old-style numerals. Lining numerals will sit far more naturally in the company of uppercase letterforms.

1912 Ditchling Sussex

Edward Johnston moves to the village, where Eric Gill already lived and an artistic community was forming.

1912 Ditchling Sussex

Edward Johnston moves to the village, where Eric Gill already lived and an artistic community was forming.

1912 DITCHLING SUSSEX

Edward Johnston moves to the village, where Eric Gill already lived and an artistic community was forming.

1912 DITCHLING SUSSEX

Edward Johnston moves to the village, where Eric Gill already lived and an artistic community was forming.

Lining numerals sit more naturally in headings than old-style numerals.

On those occasions when numbers are the star attraction, lining numerals are the way to give them the attention they crave. Old-style numerals have a wavy rhythm to them, with some numbers reaching upwards towards the capitals, some squatting at the x-height, and others ducking down below the baseline: 1234567890. This is why they work so well in continuous reading – they replicate the patterns of words in running text. However, if your reader is scanning a sequence of numbers, looking for patterns, making comparisons, or hunting for data in a list, table or other setting, they will find it far easier to do so with the familiarity and evenness of lining numerals. To specify lining numerals, set the `font-variant-numeric` property with a value of `lining-nums`:

```
h1 { font-variant-numeric: lining-nums; }
```

For legacy browsers requiring `font-feature-settings`, use the `lnum` OpenType feature tag.

Use proper subscripts and superscripts

Subscripts and superscripts are tiny numerals which are lowered or raised. They are used in chemical and mathematical formulas, as footnote indicators, and other specialist situations. For example: ‘Caffeine’ is $C_8H_{10}N_4O_2$. Mark this up meaningfully in HTML using the `<sup>` and `<sub>` elements:

```
Caffeine<sup>1</sup> is C<sup>8</sup>H<sup>10</sup>N<sup>4</sup>O<sup>2</sup>.
```

Browsers' default styling for `<sup>` and `<sub>` is to take a regular numeral, make it a bit smaller, and raise or lower it accordingly:

Caffeine¹ is C₈H₁₀N₄O₂

This works fine up to a point, but the numerals are still a little too big aesthetically and they affect the line spacing, causing unevenness in the rhythm:

Caffeine is a central nervous system stimulant of the methylxanthine class. Its chemical formula is C₈H₁₀N₄O₂. It is a bitter, white crystalline purine, a methylxanthine alkaloid, and is chemically related to the adenine and...

Most professional fonts contain properly designed subscripts and superscripts built in as OpenType features. These numerals are smaller and squatter than regular numbers, and because their position relative to other characters is part of their design, the line spacing is unaffected:

Caffeine¹ is C₈H₁₀N₄O₂

To use proper subscripts and superscripts, use the `font-variant-position` property, like this:

```
sub { font-variant-position: sub; }
sup { font-variant-position: super; }
```

Unfortunately, this still leaves us with a problem: the browser's default styling is still applied. Our special subscript character is being made smaller and it's getting moved downwards, affecting the line spacing:

Caffeine is a central nervous system stimulant of the methylxanthine class. Its chemical formula is $C_8H_{10}N_4O_2$. It is a bitter, white crystalline purine, a methylxanthine alkaloid, and is chemically related to the adenine and...

Caffeine is a central nervous system stimulant of the methylxanthine class. Its chemical formula is $C_8H_{10}N_4O_2$. It is a bitter, white crystalline purine, a methylxanthine alkaloid, and is chemically related to the adenine and...

Caffeine is a central nervous system stimulant of the methylxanthine class. Its chemical formula is $C_8H_{10}N_4O_2$. It is a bitter, white crystalline purine, a methylxanthine alkaloid, and is chemically related to the adenine and...

Top: Default <sub> styling. *Middle:* Proper subscripts with browser default styling.

Bottom: Proper subscripts only.

The styles the browser applies to our subscript characters are these:

```
vertical-align: sub; font-size: smaller;
```

We need to remove those styles to get the desired effect, so our rule now becomes:

```
sub { vertical-align: inherit;
      font-size: inherit;
      font-variant-position: sub; }
```

That will work fine for browsers that support OpenType. But browsers that don't will get C8H10N4O2, a degraded rendering compared with the browser defaults. To address this we can use an @supports rule to check if the browser supports font-variant-position and only override the browser's default <sub> styling if that's the case:

```
sub { font-variant-position: sub; }

@supports (font-variant-position: sub) {
  sub { vertical-align: inherit;
        font-size: inherit; }
}
```

For legacy browsers requiring `font-feature-settings`, use the `sup`s OpenType feature tag for superscripts, and `sub`s for subscripts. If we factor these in, we get comprehensive, backwards-compatible style rules, but where two simple properties should have done the job, we now have a far more verbose proposition:

Subscripts

```
sub { font-feature-settings: "subs" 1; }

@supports (font-variant-position: sub) {
  sub { font-feature-settings: normal;
        font-variant-position: sub; }
}
@supports ((font-variant-position: sub) or (font-feature-
settings: "subs" 1)) {
  sub { vertical-align: inherit;
        font-size: inherit; }
}
```

Superscripts

```
sup { font-feature-settings: "sups" 1; }

@supports (font-variant-position: super) {
  sup { font-feature-settings: normal;
        font-variant-position: super; }
}
@supports ((font-variant-position: super) or (font-
feature-settings: "sups" 1)) {
  sup { vertical-align: inherit;
        font-size: inherit; }
}
```

Reference notes with superscripts

One particular use of superscripts is for footnotes. When you reference notes using numbers, use true superscripts in the text but full-size numbers in the notes themselves.

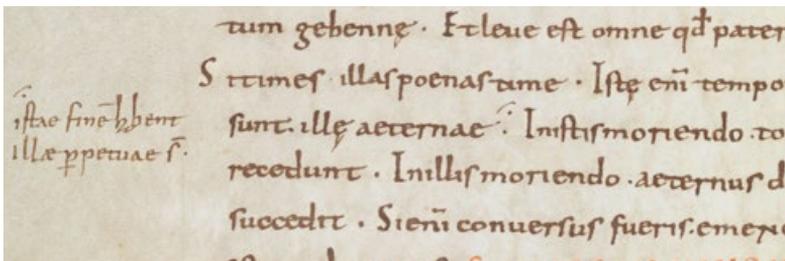
Show footnotes in context

While we're on the subject of footnotes, it's worth making a brief diversion into how the web improves their usability compared with the limitations of paper.

Many forms of writing, including academic papers, historical novels, detailed journalism and non-fiction books such as this one, contain additional citations, explanations and thoughts referred to within the text itself. A symbol is used to connect the note to the relevant location in the text. The symbols employed as references to annotations are either superscripted numbers or an esoteric series of devices starting with asterisks* and processing through daggers† to double daggers‡ and beyond.

Since the advent of mass printing in the Victorian era, the notes themselves have typically been positioned either at the bottom of the referring printed page (*footnotes*), or at the end of a chapter or the entire work (*endnotes*). However, this approach means the notes are located away from their position within the body of text. This can disturb the reader who wishes to refer to the annotation as they proceed through the text. The connected point in the text may well be halfway through a sentence in the middle of a paragraph at some point higher up the page, or on a different preceding page altogether, and attempting to return to it disrupts the reader's rhythm.

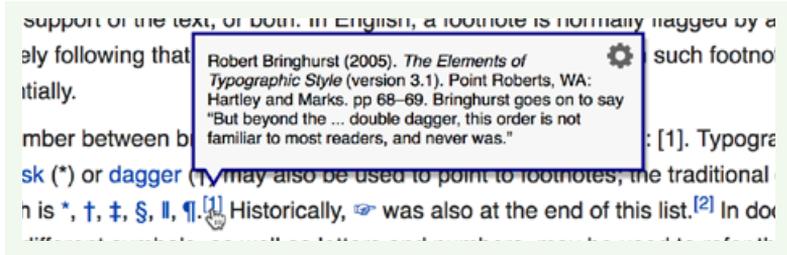
An earlier approach by medieval scribes and Renaissance printers placed notes in the margins (*side notes*) rather than at the bottom of the page. By including notes as marginalia, the annotations are present where needed and can be read with little more than a glance away from the main text.



A side note in a 9th-century manuscript. Source: Einsiedeln, Stiftsbibliothek, Codex 172(1128), p.20.

Although side notes are an improvement on footnotes, both solutions are designed within the confines of the two-dimensional printed page. The web is an interactive medium and provides us with at least three dimensions in which to work, implying you can use the z-axis to place the note *on top* of the main text.

Enable your reader to reveal the note on demand in the very place they are reading. Put simply, link to the footnote using a conventional symbol, but have it pop up in the vicinity of the link, thus providing a thoroughly modern solution impossible within the limitations of a printed page.



Clicking a superscript could pop up a footnote in situ.

Set tables as text to be read

Tables come in many forms. Some contain simple numbers, others are complex with mixtures of numeric data and textual information. Some require reading row by row, others are scanned vertically. The potential use for tables is as varied as the written word. They can be financial statements, bus timetables, multilanguage dictionaries, tables of contents, numerical conversions, pricing options, feature comparisons, technical specifications, and so on.

Despite the huge variation in table size, complexity, contents and purpose, every table shares two simple design principles: they should be readable and support a sense of the data held within. They should not be prettied up to satisfy a sense of aesthetic when simply looked at. That said, a well-designed table can still be a thing of beauty but with the form following the function. Tables are not pictures of data: they are catalogues of data to be perused, parsed, referenced and interrogated. A well-designed table will enable the information to be read and understood, and will reveal the patterns and correlations inherent in the data. As Jan Tschichold, the virtuoso of typography design, put it in *Asymmetric Typography*¹:

Tabular matter need no longer be a rather unpleasant job to design: on the contrary, it can become a really charming and artistic exercise, in no way less interesting than any other area.

1 *Asymmetric Typography* by Jan Tschichold (1967, after 1935 original).

Wherever possible plan the readability of every table in advance. Your design process should be an investigation into making the data undemanding to read, simple to follow and easy to extract.

Just as you wouldn't design body text with the aim of fitting as many words as possible on the screen, so you shouldn't treat designing a table as an exercise in cramming as much data as possible into one space. You might be tempted to reduce the text size of your table – and if the data is entirely numeric you might be able to get away with it. Your reader should still be able to be comfortably read and interpret the table from their normal position, without needing to lean in.

Don't stretch tables

Many designers will instinctively apply a width to their tables – just as they might an image – stretching them to fill the text column or page. And that is the appeal of setting tables full-width: you can make them look somewhat image-like when viewed from afar. However, while a table spread across the screen might look preferable from a distance, on closer inspection it will be harder to read as the data will be unnecessarily separated. To add insult to injury, tables set full-width are often replete with background colours and borders to give the table further the texture of an image, when what your reader really requires is the texture of text. For the sake of your readers, avoid these temptations.

You might consider making all the columns an even width. This too does nothing for the readability of the contents. Some table cells will be too wide, leaving the data lost and detached from its neighbours. Other table cells will be too narrow, cramping the data uncomfortably. Table columns should be sized according to the data they contain. Columns of small numbers should be narrow, and columns of paragraphs should be relatively wide. This sounds like a lot of effort, and for a print designer it would be, as they would have to size each column manually in their layout software. Fortunately, web browsers are very clever when it comes to laying out tables and will do all that hard work for you. Browsers have been laying out tables automatically according to complex algorithms since long before CSS came along – just let them do their thing.

Keep table furniture and fills to a minimum

The statistician and information designer Edward Tufte introduced the concept of *data-ink* in his 1983 classic, *The Visual Display of Quantitative Information*. He defines data-ink as 'the non-erasable core of the graphic', whereas non-data-ink is the ink used in the graphic, not to directly represent data

but for scales, labels, fills and edges. Tufte goes on to define the *data-ink ratio* as the proportion of ink that is used to present actual data compared to the total amount of ink used in the entire graphic. The goal is to design a graphic with the highest possible data-ink ratio (tending towards 1.0) without eliminating what is necessary for effective communication.

Where Tufte talks about graphics he includes charts, diagrams and tables, and where he uses 'ink' we can think of pixels. In terms of tables, he's saying that we should remove almost everything in the design which is not data or white space. Minimise furniture, maximise information. This is an ideal first principle to bear in mind when considering the *typographic* design of a table.

As a starting point, avoid any border or frame surrounding the table. This is a Victorian embellishment which is entirely unnecessary as text alignment will shape the table just fine.

Try to achieve a readable table using just alignment, spacing and grouping. Avoid zebra striping, tints and fills, and any other backgrounds. These can be superficially pretty but are usually a distraction. They serve to distort the meaning of the data by highlighting every other row to the detriment of neighbouring rows. Only use tints as a subtle means of guiding your reader's eyes, and then only if you cannot arrange the data to that end. If you choose to tint, do so only in the primary direction of reading: down if lists, across otherwise.

When it comes to lines and borders between rows and columns – typographically referred to as *rules* – the same applies: use them judiciously and preferably not at all. In *Asymmetric Typography* Jan Tschichold sums this up wonderfully:

Tables should not be set to look like nets with every number enclosed. Try to do without rules altogether. They should be used only when they are absolutely necessary. Vertical rules are needed only when the space between columns is so narrow that mistakes will occur in reading without rules. Tables without vertical rules look better. Thin rules are better than thick ones.

Avoid using row or column borders unless the data alignment, spacing and grouping are not sufficient to guide your reader's eye. If you do need to use rules for this purpose, use them in one direction only and employ a lighter colour to reduce the impact of the lines: you are making a distinction, not constructing a barricade.

Left-align text, right-align numbers, and align headings with data

In the spirit of treating tables as artefacts to be read, don't centre-align text within tables. Align table text as you would anywhere else; that is, aligned left. As text in tables tends to end up in narrow columns, don't justify the text either – leave it ragged-right – or you will end up with rivers flowing down the tables, potentially causing confusion and certainly harming readability. You can hyphenate, however, particularly if the table columns would otherwise have a pronounced rag.

Right-align numbers to help your reader make easier comparisons of magnitude when scanning down columns. To aid scanning in this manner you will need consistent precision of your numeric data; that is, use the same number of decimal places.

For consistency and ease of understanding, match the alignment of headings to the alignment of the data. Right-align headings of numeric data and left-align headings of columns with text, for example:

Country	Area	Population	GDP	Capital
Austria	83,858	8,169,929	339	Vienna
Belgium	30,528	11,007,000	410	Brussels
Denmark	43,094	5,564,219	271	Copenhagen
France	547,030	66,104,000	2,181	Paris
Germany	357,021	80,716,000	3,032	Berlin
Greece	131,957	11,123,034	176	Athens
Ireland	70,280	4,234,925	255	Dublin
Italy	301,230	60,655,464	1,642	Rome
Luxembourg	2,586	448,569	51	Luxembourg
Netherlands	41,526	16,902,103	676	Amsterdam
Portugal	91,568	10,409,995	179	Lisbon
Spain	504,851	47,059,533	1,075	Madrid
Sweden	449,964	9,090,113	447	Stockholm
UK	244,820	65,110,000	2,727	London

Align to the decimal point

You may find yourself not having control of numerical precision, or perhaps the data you're working with is rounded to the same *significant number* rather than adhering to the same precision. In this case, simply right-aligning a column of numbers will not help your reader scan down the column: small, high-precision numbers will look at first glance like a large number. Instead, align numbers to the decimal point. This will enable your reader to more readily compare magnitudes among a wider variety of data:

```
+-----+
| Call charge |
+-----+
|    $1.30    |
|    $2.50    |
|   $10.80    |
|  $111.01    |
|    $85      |
|    N/A      |
|    $.05     |
|    $.06     |
+-----+
```

Aligning to the decimal point was theoretically possible by using the HTML4 `char` attribute on a `<td>` tag, but in reality it was never supported. The modern way to align numbers to a decimal point (or to any character, in fact) is through a new value of the `text-align` property, although at the time of writing this is languishing in the CSS Text Level 4 Module² and support is patchy at best.

The syntax of the new value is simple. You include the alignment character (usually a full stop or comma) in quotes, followed by a space and your desired alignment keyword, which defaults to `right` if you omit it. For example, the following will centre the data and align to a decimal point as in our prior example:

```
td { text-align: "." center; }
```

By specifying different alignment characters you can lay out more complex tables in a useful way; in this example, aligning digits to 'x' and 'y':

² [Character-based Alignment in a Table Column](http://wbtyp.net/103) (<http://wbtyp.net/103>) in the CSS Text Module Level 4.

Video standard	Resolution	Pixels	Aspect ratio
QQVGA	160 × 120	19k	4:3
HQVGA	240 × 160	38k	3:2
QVGA	320 × 240	76k	4:3
WQVGA	480 × 272	130k	16:9
VGA	640 × 480	307k	4:3
SVGA	800 × 600	480k	4:3
XGA	1024 × 768	786k	4:3
HD	1260 × 768	967k	16:9
WXGA	1280 × 800	1,024k	16:10
SXGA	1280 × 1024	1,310k	5:4
UXGA	1600 × 1200	1,920k	4:3
FHD	1920 × 1080	2,073k	16:9
DCI 2K	2048 × 1080	2,211k	19:10
WQXGA	2560 × 1600	4,096k	16:10
4K UHD	3840 × 2160	8,294k	16:9
8K UHD	7680 × 4320	33,177k	16:9

Selected display standards

Use tabular lining numerals in tables of numbers

Many tables, such as financial statements or timetables, are made up mostly of numbers. Generally speaking, their purpose is to provide the reader with numeric data, presented in either columns or rows, and sometimes in a matrix of the two. Your reader may use the table by scanning down the columns, either searching for a data point or by making comparisons between numbers. Your reader may also make sense of the data by simply glancing at the column or row. It is far easier to compare numbers if the ones, tens and hundreds are all lined up vertically; that is, all the digits should occupy exactly the same width.

Digits of the same width can inherently be found in monospaced fonts, and there is nothing wrong with choosing a suitable monospaced font to present a table of data (see ‘[Combining typefaces](#)’). However, many proportionally spaced fonts (those where a *l* is narrower than an *8*, and a *W* is wider than an *I*) also come with additional sets of figures which are monospaced.

These are called *tabular numerals*. As well as being of equal width, tabular numerals will be subtly designed differently from the standard proportional numerals. For example, a 1 will normally have a bar for its base, and a 0 (zero) may be designed slightly narrower to better fit the chosen number width. Tabular numerals are usually available in old-style and lining variations. Use tabular lining numerals to provide your reader with the most effective way to reference vertically and horizontally in tables of data.

	Lining	Old-style
Proportional	409,280	409,280
	367,112	367,112
	155,068	155,068
	171,792	171,792
Tabular	409,280	409,280
	367,112	367,112
	155,068	155,068
	171,792	171,792

Different numeral styles compared.

To specify tabular lining numerals, set the `font-variant-numeric` property with a value of `lining-nums` and `tabular-nums`:

```
table { font-variant-numeric: lining-nums tabular-nums; }
```

The equivalent properties for legacy browsers requiring `font-feature-settings`, use the `lnum` and `tnum` OpenType feature tags.

Proportional numerals

If you need to specify proportional numerals, set the `font-variant-numeric` property with a value of `proportional-nums`. For legacy browsers requiring `font-feature-settings`, use the `pnum` OpenType feature tag.

Put white space to work to group and separate

Having eliminated rules and fills (borders and backgrounds) from your table, you will need to apply white space to your table so your reader can make sense of it. It is at this point that you should remove from your mind's eye all visions of spreadsheets and other such uniform grids, and think instead in terms of typography and simple gestalt grouping principles.

You will primarily need to separate the data so that each element can be individually identified and read as separate from the others. To have more control over the spacing, first collapse the spacing between borders:

```
table { border-collapse: collapse; }
```

In traditional HTML tables, adjacent cells each have their own distinct borders which are separated from each other, with the separation still present even if the borders are not. In the collapsed border model, adjacent table cells share borders. As we are removing (almost) all cell borders, and any we retain will be single key lines, the collapsed border model is the most appropriate.

Now apply padding to the table cells to separate the data. You'll find that adding a smaller amount of padding to the top of the cell is a useful way to provide a visually balanced separation from the rows above and below. To ensure everything lines up nicely, apply the same padding to heading cells as to data cells. Because line lengths are often very short in tables, you can reduce the line height right down. In the following example, we've removed all additional line spacing, but you may need more depending on your choice of font and the amount of text in the table cells.

```
td, th {
  padding: 0.125em 0.5em 0.25em 0.5em;
  line-height: 1;
}
```

The gestalt grouping principles most useful in tables are those of proximity and similarity. Move related data closer together to be distinct from other data; in other words, space apart groups of rows or columns. A by-product of grouping rows is that the data becomes much easier to scan and refer to than if the table consisted of a succession of undifferentiated rows. Ensure data of a similar content or meaning look similar at a glance, which you can do through alignment, colour and font style.

Table captions

We will attend to the typographic specifics of table captions in 'Choosing typefaces for functional text' but it's worth noting now how to mark up captions for tables. If you are choosing to place your table inside a `<figure>` element, which is a perfectly reasonable thing to do, then use a `<figcaption>` element before or after the table. If your table is not inside

a `<figure>` element, or you have multiple items in the figure, use the aptly named `<caption>` element, which HTML provides specifically for tables. Always write the `<caption>` tag immediately after the opening `<table>` tag and before any table data, like this:

```
<table>
  <caption>
    Imperial to metric conversion factors
    <p><i>Values are given to three significant figures</i></p>
  </caption>
  <thead> ... </thead>
  <tbody> ... </tbody>
</table>
```

You can position the caption either above or below the table using the `caption-side` property and a corresponding value of either `top` or `bottom`.

```
caption { caption-side: bottom; }
```

The following table shows a caption and demonstrates gestalt grouping principles by separating the data into related rows:

	<i>To convert</i>	<i>into</i>	<i>multiply by</i>
Length	inches	millimetres (mm)	25.4
	feet	centimetres (cm)	30.48
	yards	metres (m)	0.914
	miles	kilometres (km)	1.61
Area	square inches	sq. millimetres (mm ²)	645
	square feet	square metres (m ²)	0.093
	square yards	square metres (m ²)	0.836
	acres	hectares	2.47
Volume	cubic inches	millilitres (ml)	16.4
	cubic feet	litres	28.3
	imperial gallons	litres	4.55
	US barrels	cubic metres (m ³)	0.159

Imperial to metric conversion factors
Values are given to three significant figures unless exact

Note that in this example, the numbers do not align to the decimal point. This is because the purpose of the table is for the reader to easily identify and extract a multiplication factor. In this instance there is no obvious use case for comparing the relative magnitudes of the factors, which is when decimal alignment would be useful.

Do not over-stylise tables

The French writer-aviator Antoine de Saint-Exupéry wrote³ ‘perfection is attained not when there is nothing more to add, but when there is no longer anything to take away.’ Quoting de Saint-Exupéry may have become a cliché, but his idiom is entirely apt when applied to table design.

There is no need to make a table look like a spreadsheet. A spreadsheet is a tool unto itself; a table is for presenting data and information that can be read. Spreadsheet software offers a multitude of options for table styles, which add text formatting, borders, background fills and all manner of ornament. They may make pretty pictures but do nothing for table readability, so do not try to emulate them. Tables can be beautiful but they are not works of art. Instead of painting and decorating them, design tables for your reader.

Director	Film	Released	Budget	Gross	Rating
John Ford	The Informer	May 1935	\$243,000	\$950,000	8.3/10
	The Grapes of Wrath	Jan 1940	\$800,000	\$2,500,000	9/10
	How Green Was My Valley	Oct 1941	\$800,000	\$2,800,000	7.9/10
William Wyler	The Quiet Man	Jul 1952	\$1,750,000	\$3,200,000	8/10
	Mrs. Miniver	Jun 1942	\$1,340,000	\$8,878,000	7.8/10
	The Best Years of Our Lives	Nov 1946	\$2,100,000	\$23,700,000	8.6/10
Frank Capra	Ben-Hur	Nov 1959	\$15,200,000	\$146,900,000	7.9/10
	It Happened One Night	Feb 1934	\$325,000	\$2,500,000	8.9/10
	Mr. Deeds Goes to Town	Apr 1936	\$845,710	\$1,000,000	8.4/10
Billy Wilder	You Can't Take It With You	Aug 1938	\$1,644,736	\$5,295,526	7.6/10
	The Lost Weekend	Nov 1945	\$1,250,000	\$11,000,000	8.2/10
	The Apartment	Jun 1960	\$3,000,000	\$24,600,000	8.5/10
David Lean	The Bridge on the River Kwai	Oct 1957	\$2,800,000	\$30,600,000	9.2/10
Steven Spielberg	Lawrence of Arabia	Dec 1962	\$15,000,000	\$70,000,000	9/10
	Schindler's List	Nov 1993	\$22,000,000	\$321,200,000	9/10
	Saving Private Ryan	Jul 1998	\$70,000,000	\$481,800,000	8.6/10
Fred Zinnemann	From Here to Eternity	Aug 1953	\$2,500,000	\$30,500,000	8.2/10
	A Man for All Seasons	Dec 1966	\$2,000,000	\$28,400,000	7.8/10

A typical spreadsheet-styled table set full-width with borders, fills and centre-alignment.

3 *Terre des Hommes* (translated into English as *Wind, Sand and Stars*) by Antoine de Saint-Exupéry (1939).

Director	Film	Released	Budget	Gross	Rating
John Ford	The Informer	May 1935	\$243,000	\$950,000	8.3/10
	The Grapes of Wrath	Jan 1940	\$800,000	\$2,500,000	9/10
	How Green Was My Valley	Oct 1941	\$800,000	\$2,800,000	7.9/10
	The Quiet Man	Jul 1952	\$1,750,000	\$3,200,000	8/10
William Wyler	Mrs. Miniver	Jun 1942	\$1,340,000	\$8,878,000	7.8/10
	The Best Years of Our Lives	Nov 1946	\$2,100,000	\$23,700,000	8.6/10
	Ben-Hur	Nov 1959	\$15,200,000	\$146,900,000	7.9/10
Frank Capra	It Happened One Night	Feb 1934	\$325,000	\$2,500,000	8.9/10
	Mr. Deeds Goes to Town	Apr 1936	\$845,710	\$1,000,000	8.4/10
	You Can't Take It With You	Aug 1938	\$1,644,736	\$5,295,526	7.6/10
Billy Wilder	The Lost Weekend	Nov 1945	\$1,250,000	\$11,000,000	8.2/10
	The Apartment	Jun 1960	\$3,000,000	\$24,600,000	8.5/10
David Lean	The Bridge on the River Kwai	Oct 1957	\$2,800,000	\$30,600,000	9.2/10
	Lawrence of Arabia	Dec 1962	\$15,000,000	\$70,000,000	9/10
Steven Spielberg	Schindler's List	Nov 1993	\$22,000,000	\$321,200,000	9/10
	Saving Private Ryan	Jul 1998	\$70,000,000	\$481,800,000	8.6/10
Fred Zinnemann	From Here to Eternity	Aug 1953	\$2,500,000	\$30,500,000	8.2/10
	A Man for All Seasons	Dec 1966	\$2,000,000	\$28,400,000	7.8/10

1. Remove stretch and size columns to data.

Director	Film	Released	Budget	Gross	Rating
John Ford	The Informer	May 1935	\$243,000	\$950,000	8.3/10
	The Grapes of Wrath	Jan 1940	\$800,000	\$2,500,000	9/10
	How Green Was My Valley	Oct 1941	\$800,000	\$2,800,000	7.9/10
	The Quiet Man	Jul 1952	\$1,750,000	\$3,200,000	8/10
William Wyler	Mrs. Miniver	Jun 1942	\$1,340,000	\$8,878,000	7.8/10
	The Best Years of Our Lives	Nov 1946	\$2,100,000	\$23,700,000	8.6/10
	Ben-Hur	Nov 1959	\$15,200,000	\$146,900,000	7.9/10
Frank Capra	It Happened One Night	Feb 1934	\$325,000	\$2,500,000	8.9/10
	Mr. Deeds Goes to Town	Apr 1936	\$845,710	\$1,000,000	8.4/10
	You Can't Take It With You	Aug 1938	\$1,644,736	\$5,295,526	7.6/10
Billy Wilder	The Lost Weekend	Nov 1945	\$1,250,000	\$11,000,000	8.2/10
	The Apartment	Jun 1960	\$3,000,000	\$24,600,000	8.5/10
David Lean	The Bridge on the River Kwai	Oct 1957	\$2,800,000	\$30,600,000	9.2/10
	Lawrence of Arabia	Dec 1962	\$15,000,000	\$70,000,000	9/10
Steven Spielberg	Schindler's List	Nov 1993	\$22,000,000	\$321,200,000	9/10
	Saving Private Ryan	Jul 1998	\$70,000,000	\$481,800,000	8.6/10
Fred Zinnemann	From Here to Eternity	Aug 1953	\$2,500,000	\$30,500,000	8.2/10
	A Man for All Seasons	Dec 1966	\$2,000,000	\$28,400,000	7.8/10

2. Remove fills, gridlines, border and bolding.

Director	Film	Released	Budget	Gross	Rating
John Ford	The Informer	May 1935	\$243,000	\$950,000	8.3/10
	The Grapes of Wrath	Jan 1940	\$800,000	\$2,500,000	9/10
	How Green Was My Valley	Oct 1941	\$800,000	\$2,800,000	7.9/10
	The Quiet Man	Jul 1952	\$1,750,000	\$3,200,000	8/10
William Wyler	Mrs. Miniver	Jun 1942	\$1,340,000	\$8,878,000	7.8/10
	The Best Years of Our Lives	Nov 1946	\$2,100,000	\$23,700,000	8.6/10
	Ben-Hur	Nov 1959	\$15,200,000	\$146,900,000	7.9/10
Frank Capra	It Happened One Night	Feb 1934	\$325,000	\$2,500,000	8.9/10
	Mr. Deeds Goes to Town	Apr 1936	\$845,710	\$1,000,000	8.4/10
	You Can't Take It With You	Aug 1938	\$1,644,736	\$5,295,526	7.6/10
Billy Wilder	The Lost Weekend	Nov 1945	\$1,250,000	\$11,000,000	8.2/10
	The Apartment	Jun 1960	\$3,000,000	\$24,600,000	8.5/10
David Lean	The Bridge on the River Kwai	Oct 1957	\$2,800,000	\$30,600,000	9.2/10
	Lawrence of Arabia	Dec 1962	\$15,000,000	\$70,000,000	9/10
Steven Spielberg	Schindler's List	Nov 1993	\$22,000,000	\$321,200,000	9/10
	Saving Private Ryan	Jul 1998	\$70,000,000	\$481,800,000	8.6/10
Fred Zinnemann	From Here to Eternity	Aug 1953	\$2,500,000	\$30,500,000	8.2/10
	A Man for All Seasons	Dec 1966	\$2,000,000	\$28,400,000	7.8/10

3. Left-align text, right-align numbers and align headings with data.

Director	Film	Released	Budget	Gross	Rating
John Ford	The Informer	May 1935	\$243,000	\$950,000	8.3/10
	The Grapes of Wrath	Jan 1940	\$800,000	\$2,500,000	9/10
	How Green Was My Valley	Oct 1941	\$800,000	\$2,800,000	7.9/10
	The Quiet Man	Jul 1952	\$1,750,000	\$3,200,000	8/10
William Wyler	Mrs. Miniver	Jun 1942	\$1,340,000	\$8,878,000	7.8/10
	The Best Years of Our Lives	Nov 1946	\$2,100,000	\$23,700,000	8.6/10
	Ben-Hur	Nov 1959	\$15,200,000	\$146,900,000	7.9/10
Frank Capra	It Happened One Night	Feb 1934	\$325,000	\$2,500,000	8.9/10
	Mr. Deeds Goes to Town	Apr 1936	\$845,710	\$1,000,000	8.4/10
	You Can't Take It With You	Aug 1938	\$1,644,736	\$5,295,526	7.6/10
Billy Wilder	The Lost Weekend	Nov 1945	\$1,250,000	\$11,000,000	8.2/10
	The Apartment	Jun 1960	\$3,000,000	\$24,600,000	8.5/10
David Lean	The Bridge on the River Kwai	Oct 1957	\$2,800,000	\$30,600,000	9.2/10
	Lawrence of Arabia	Dec 1962	\$15,000,000	\$70,000,000	9/10
Steven Spielberg	Schindler's List	Nov 1993	\$22,000,000	\$321,200,000	9/10
	Saving Private Ryan	Jul 1998	\$70,000,000	\$481,800,000	8.6/10
Fred Zinnemann	From Here to Eternity	Aug 1953	\$2,500,000	\$30,500,000	8.2/10
	A Man for All Seasons	Dec 1966	\$2,000,000	\$28,400,000	7.8/10

4. Put white space to work to group and separate.

Director	Film	Released	Budget (\$)	Gross (\$)	Rating
John Ford	The Informer	May 1935	243,000	950,000	8.3
	The Grapes of Wrath	Jan 1940	800,000	2,500,000	9.0
	How Green Was My Valley	Oct 1941	800,000	2,800,000	7.9
	The Quiet Man	Jul 1952	1,750,000	3,200,000	8.0
William Wyler	Mrs. Miniver	Jun 1942	1,340,000	8,878,000	7.8
	The Best Years of Our Lives	Nov 1946	2,100,000	23,700,000	8.6
	Ben-Hur	Nov 1959	15,200,000	146,900,000	7.9
Frank Capra	It Happened One Night	Feb 1934	325,000	2,500,000	8.9
	Mr. Deeds Goes to Town	Apr 1936	845,710	1,000,000	8.4
	You Can't Take It With You	Aug 1938	1,644,736	5,295,526	7.6
Billy Wilder	The Lost Weekend	Nov 1945	1,250,000	11,000,000	8.2
	The Apartment	Jun 1960	3,000,000	24,600,000	8.5
David Lean	The Bridge on the River Kwai	Oct 1957	2,800,000	30,600,000	9.2
	Lawrence of Arabia	Dec 1962	15,000,000	70,000,000	9.0
Steven Spielberg	Schindler's List	Nov 1993	22,000,000	321,200,000	9.0
	Saving Private Ryan	Jul 1998	70,000,000	481,800,000	8.6
Fred Zinnemann	From Here to Eternity	Aug 1953	2,500,000	30,500,000	8.2
	A Man for All Seasons	Dec 1966	2,000,000	28,400,000	7.8

5. Use tabular lining numerals, consistent precision, and remove repetition.

Adapt tables to small screens

Tables regularly require a fair bit of horizontal space to display the information they contain. Even when judiciously designed and edited, a typical table may need to be wider than the 45–75 characters we normally allow for paragraphs of text. For small screens, such as phones, designing readable tables which work under such cramped conditions presents us with a serious challenge. The best approaches always deal with each table on case-by-case basis, but that's not always possible if we need to generically style whatever comes out of a CMS database.

One immediate approach is to use either a condensed font or a slightly smaller size (but not both smaller and condensed). In both cases, readability must remain paramount and other options should also be explored.

Consider setting oblique headings to save space

One way to save horizontal space, particularly when you have short pieces of data but long headings, is to set the headings at an oblique angle.

Celcius (°C)	Fahrenheit (°F)
0	32
10	50
20	68
30	86
40	104
50	122
60	140
70	158
80	194
90	194
100	212

Using oblique headings to save space.

You can use a simple CSS translation to achieve the effect. You will also need to absolutely position the headings so the original width of the columns isn't retained and they shrink to wrap the data instead.

```
th {
  transform-origin: bottom left;
  position: absolute;
}

th.degC {
  transform: translate(2.58em,-2em) rotate(-60deg);
}

th.degF {
  transform: translate(5.14em,-2em) rotate(-60deg);
}
```

Let the browser handle tables with horizontal scrolling

The simplest solution to help tables of any size and complexity is to let the browser lay out the table as best it can and render part of the table off-screen as necessary. Provided you then enable your reader to scroll the table sideways independently of the rest of the text, the table can be relatively easily brought into view.

	MONDAY	TUESDAY
Soup	Pea and mint topped with shredded spring onion	Tomato art with parm
Bistro dish	Chicken stroganoff, with mushrooms and cream	Teriyaki sa coriander vegetable
Comfort food	Butchers pork sausages, caramelised red onion jus	Beef bolog garlic brea
Desert	Apple and cranberry crumble, custard	Syrup spo with crèm

Using a crawl bar to scroll a table into view.

To do this, first wrap your table in a `<figure>` element:

```
<figure class="fig-table">
  <table> ... </table>
</figure>
```

Then apply the following simple rules to hide the portion of the table off-screen and enable your reader to scroll the table without affecting the rest of the text:

```
.fig-table {
  max-width: 100%;
  overflow-x: scroll;
}
```

It is important not to set a width on your table; the browser can then compress the table as far as it can before overflowing off the screen. To preserve readability, make good use of non-breaking spaces and white-space: nowrap to limit the amount the data wraps in the cells. It's better to have a readable table that requires scrolling than an unreadable one which doesn't.

Linearise simple tables into lists

You can safely linearise simple data tables when space is limited. The tables most suitable for this treatment are lists of structured data; for example, an employee directory:

Name	Email	Title	Phone
Jones, Claire	claire.jones@domain.com	Managing Director	01234 567890
Smith, Darren	darren.smith@domain.com	Head of Sales	01234 567891

When there is not enough room for the table to render comfortably, we can set it with a completely different layout. This is less compact overall, and takes more space vertically, but it succeeds in fitting the table into a much narrower viewport:

Name:	Jones, Claire
Email:	claire.jones@domain.com
Title:	Managing Director
Phone:	01234 567890
Name:	Smith, Darren
Email:	darren.smith@domain.com
Title:	Head of Sales
Phone:	01234 567891

The two renderings of our employee directory table use exactly the same markup, comprising the conventional HTML elements you would expect in any table. The one addition is a `data-title` attribute on each cell enabling us to repeat the label in the list view, should we need to.

```
<th data-title="Name">Jones, Claire</th>
<th data-title="Email">claire.jones@domain.com</th>
<th data-title="Title">Managing Director</th>
<th data-title="Phone">01234&nbsp;567890</th>
```

There are four simple steps to turning the table into a list, using a media query and CSS (no JavaScript is required).

1. Identify the viewport width at which the table starts to render poorly.
2. Apply `display: block` to all table-related elements so they align vertically instead of as a table.
3. Hide the header row and any empty cells.
4. Display labels for each data item (optional).

You will need to apply some additional styling for aesthetics and readability, but the responsiveness described can be accomplished in these few lines of CSS:

```
@media (max-width: 25em) {  
  table, caption, tbody, tr, th, td {  
    display: block;  
    text-align: left;  
  }  
  thead, th:empty, td:empty {  
    display: none;  
    visibility: hidden;  
  }  
  th[data-title]:before, td[data-title]:before {  
    content: attr(data-title) ": ";  
    display: inline-block;  
    width: 3.5em;  
  }  
}
```

This technique was first popularised by Aaron Gustafson⁴.

Make tables responsive according to their purpose

There are many different techniques⁵ available for making data tables responsive. Some are simple CSS-only methods (we've covered two already); others are complex, enhanced by JavaScript. When considering which technique to use, ask yourself how your reader will use the table. In particular, consider if your reader is likely to compare either rows or columns – these kinds of tables need extra attention owing to the way they are used.

When being able to compare columns is important, one method is to hide non-essential fields and provide an option to turn them back on.

4 'Responsive Tables' (<http://wbtyp.net/16>) by Aaron Gustafson on *Easy Designs blog* (2013).

5 See CSS-Tricks' 'Responsive Tables' (<http://wbtyp.net/148>) for the latest options.

This technique was popularised by Filament Group⁶ using a stocks table as an example:

Company	Last Trade	Trade Time	Change	Prev Close	Open	Bid	Ask	1y Target Est
GOOG Google Inc.	597.74	12:12PM	14.81 (2.54%)	582.93	597.95	597.73 x 100	597.91 x 300	731.10
AAPL Apple Inc.	378.94	12:22PM	5.74 (1.54%)	373.20	381.02	378.92 x 300	378.99 x 100	505.94
AMZN Amazon.com Inc.	191.55	12:23PM	3.16 (1.68%)	188.39	194.99	191.52 x 300	191.58 x 100	240.32
ORCL Oracle Corporation	31.15	12:44PM	1.41 (4.72%)	29.74	30.67	31.14 x 6500	31.15 x 3200	36.11
MSFT Microsoft Corporation	25.50	12:27PM	0.66 (2.67%)	24.84	25.37	25.50 x 71100	25.51 x 17800	31.50
CSCO Cisco Systems, Inc.	18.65	12:45PM	0.97 (5.49%)	17.68	18.23	18.65 x 10300	18.66 x 24000	21.12
YHOO Yahoo! Inc.	15.81	12:25PM	0.11 (0.67%)	15.70	15.94	15.79 x 6100	15.80 x 17000	18.16

A data-rich table rendered on a large screen.

Company	Last Trade	Change	1y Target Est
GOOG Google Inc.	597.74	14.81 (2.54%)	731.10
AAPL Apple Inc.	378.94	5.74 (1.54%)	505.94
AMZN Amazon.com Inc.	191.55	3.16 (1.68%)	240.32
ORCL Oracle Corporation	31.15	1.41 (4.72%)	36.11
MSFT Microsoft Corporation	25.50	0.66 (2.67%)	31.50
CSCO Cisco Systems, Inc.	18.65	0.97 (5.49%)	21.12

The same table with hidden columns and options to toggle.

Tables are a frequently overlooked aspect of reading, sometimes overstyled, sometimes poorly thought out. Responsiveness is a particularly thorny issue as the best solutions depend very much on the utility of the table. Tables can be packed with data, rich in content and meaning. Give them the attention they deserve.

⁶ Filament Group's 'Tablesaw' (<http://wbtyp.net/15>) responsive table plug-ins.