



Links can look and work better, can they?

In the beginning: CSS and CSS 2

Links have been on the web since the beginning as a representation of a connection to another document. In the beginning you could only change the color of the link itself and its underline. By default, text links are blue and underlined. This underlining strikes through descenders - letters that dip below the baseline, such as a lowercase j - making some links more difficult to read.

with CSS level 2 we got the `text-decoration` keyword that allowed to change the way underlines in links and other elements looked like. With it you could do something like this to create lines above and below a matching element:

```
.over {  
  text-decoration: overline;  
}  
  
.regular {  
  text-decoration: underline;  
}
```

This was good but limited. The values available were:

- none to remove any underline
- underline
- overline to create a line above the content
- line-through to strike through the text
- blink
- inherit from the parent element

you could change the color of the link (and the text) using the `color` keyword and the position of the link using `text-decoration` but that was the extent of it. The underline would still run through the descenders of letters and number

CSS 3

CSS has now developed an entire module dedicated to text decorations. [CSS Text Decoration Module Level 3](#) has been sitting at the candidate recommendation stage for almost 4 years yet it provides very interesting properties concerning underlines and other text decorations. We will only concentrate on the decorations part of the spec.

text-decoration-color

This rule defines the color of the underline. It can be any of the CSS color formats: color names, hexadecimal, rgb, rgba, hsl or the transparent andcurrentColor keywords. The example below shows some possibilities for the text-decoration-color keyword.

```
.demo {  
  /* <color> values */  
  text-decoration-color: currentColor;  
  text-decoration-color: red;  
  text-decoration-color: #00ff00;  
  text-decoration-color: rgba(255, 128, 128, 0.5);  
  text-decoration-color: transparent;  
}
```

text-decoration-line

This rule defines the type of line that will be used with text decoration. It is the successor of the CSS 2 property with a different name. The rule also allows two or more selectors but be careful not to go overboard like the last example of the CSS block below shows.

```
.demo2 {  
  /* Possible text-decoration-line values */  
  text-decoration-line: none;  
  text-decoration-line: underline;  
  text-decoration-line: overline;  
  text-decoration-line: line-through;  
}
```

```

text-decoration-line: blink;
/* Two decoration lines */
text-decoration-line: underline overline;
/* Multiple decoration lines */
text-decoration-line: overline underline line-through;
}

```

The table below explains the different values for the text-decoration-line keyword.

Explanation of the values for text-decoration-style

Value	Description
underline	Each line of text is underlined.
overline	Each line of text has a line above it.
line-through	Each line of text has a line through the middle.
blink	The text blinks (alternates between visible and invisible). This value is deprecated in favor of Animations.

text-decoration-style

The text-decoration-style keyword defines what type of underline we use.

```

.demo3 {
/* Possible text-decoration-style values */
text-decoration-style: solid;
text-decoration-style: double;
text-decoration-style: dotted;
text-decoration-style: dashed;
text-decoration-style: wavy;
}

```

Definition of the values for text-decoration-style

Keyword	Description
solid	Draws a single line

Keyword	Description
double	Draws a double line
dotted	Draws a dotted line
dashed	Draws a dashed line
wavy	Draws a wavy line

text-decoration

text-decoration is a shorthand for text-decoration-color, text-decoration-line, and text-decoration-style CSS properties. Like for any other shorthand property, it means that it resets their value to their default if not explicitly set in the shorthand.

text-decoration-skip

The text-decoration-skip CSS property specifies what parts of the element's content any text decoration affecting the element must skip over. It controls all text decoration lines drawn by the element and also any text decoration lines drawn by its ancestors.

```
.demo4 {  
  /* Possible values for text-decoration-skip */  
  text-decoration-skip: none;  
  text-decoration-skip: objects;  
  text-decoration-skip: spaces;  
  text-decoration-skip: ink;  
  text-decoration-skip: edges;  
  text-decoration-skip: box-decoration;  
  
  /* We can also combine values */  
  text-decoration-skip: ink spaces;  
  text-decoration-skip: ink edges box-decoration;  
}
```

Value	Description
none	Nothing is skipped, i.e. text decoration is drawn for all text content and across atomic inline-level boxes.
objects	The entire margin box of the element is skipped if it is an atomic inline such as an image or inline-block.
spaces	All spacing is skipped, i.e. all Unicode white space characters and all word separators, plus any adjacent letter-spacing or word-spacing.
ink	<p>The text decoration is only drawn where it does not touch or closely approach a glyph. I.e. it is interrupted where it would otherwise cross over a glyph.</p> <p>example of decoration-skipink</p>
edges	<p>The start and end of the text decoration is placed slightly inward (e.g. by half of the line thickness) from the content edge of the decorating box. E.g. two underlined elements side-by-side do not appear to have a single underline. (This is important in Chinese, where underlining is a form of punctuation).</p> <p>example of decoration-skip-edges</p>
box-decoration	The text decoration is skipped over the box's margin, border and padding areas. This only has an effect on decorations imposed by an ancestor; a decorating box never draws over its own box decoration.

Support and managing multiple

browsers

Support for the level 3 text decoration properties is not even. The specification has been sitting at candidate recommendation stage for almost 4 years and only now different browsers have come out with implementations.

How do we handle working with multiple browsers? We use the cascade and rely on the fact that browsers will not work with rules they don't understand. For example we can do something like this:

```
a {  
  /* CSS 1 */  
  color: blue;  
  /* CSS 2 */  
  text-decoration: underline;  
  /* CSS 3 */  
  text-decoration: underline navy solid;  
}
```

- Browsers who only support CSS 1 will change the color to blue.
- Browsers that support CSS 2 will ignore the first rule and apply the second one and use underline as the text decoration
- Browsers that support the CSS level 3 specification will use the last rule where it controls the decoration color and style in one declaration

Further ideas

There are other features that we can use to further refine links.

If you're working with languages other than English you can use [text-underline-position](#) to control where the underline will appear on the text.