# Annotations in PDFKit

Annotations are interactive features of the PDF format, and they make it possible to include things like links and attached notes, or to highlight, underline or strikeout portions of text. Annotations are added using the various helper methods, and each type of annotation is defined by a rectangle and some other properties. Here is a list of the available annotation methods:

```
note(x, y, width, height, contents, options)
link(x, y, width, height, url, options)
goTo(x, y, w, h, name, options)
highlight(x, y, width, height, options)
underline(x, y, width, height, options)
strike(x, y, width, height, options)
lineAnnotation(x1, y1, x2, y2, options)
rectAnnotation(x, y, width, height, options)
ellipseAnnotation(x, y, width, height, options)
textAnnotation(x, y, width, height, text, options)
fileAnnotation(x, y, width, height, file, options)
```

Many of the annotations have a **`color`** option that you can specify. You can use an array of RGB values, a hex color, or a named CSS color value for that option.

If you are adding an annotation to a piece of text, such as a link or underline, you will need to know the width and height of the text in order to create the required rectangle for the annotation. There are two methods that you can use to do that. To get the width of any piece of text in the current font, just call the **`widthOfString`** method with the string you want to measure. To get the line height in the current font, just call the **`currentLineHeight`** method.

You must remember that annotations have a stacking order. If you are putting more than one annotation on a single area and one of those annotations is a link, make sure that the link is the last one you add, otherwise it will be covered by another annotation and the user won't be able to click it.