

Looking Up Template Files

The `find_template` helper is used to find template files for rendering:

```
find_template settings.views, 'foo', Tilt[:haml] do |file|
  puts "could be #{file}"
end
```

This is not really useful. But it is useful that you can actually override this method to hook in your own lookup mechanism. For instance, if you want to be able to use more than one view directory:

```
set :views, ['views', 'templates']

helpers do
  def find_template(views, name, engine, &block)
    Array(views).each { |v| super(v, name, engine, &block) }
  end
end
```

Another example would be using different directories for different engines:

```
set :views, :sass => 'views/sass', :haml => 'templates', :default => 'views'

helpers do
  def find_template(views, name, engine, &block)
    _, folder = views.detect { |k,v| engine == Tilt[k] }
    folder ||= views[:default]
    super(folder, name, engine, &block)
  end
end
```

You can also easily wrap this up in an extension and share with others!

Note that `find_template` does not check if the file really exists but rather calls the given block for all possible paths. This is not a performance issue, since `render` will use `break` as soon as a file is found. Also, template locations (and content) will be cached if you are not running in development mode. You should keep that in mind if you write a really crazy method.