How to publish code examples in WordPress

If you are a web developer who often publishes his demos and examples, you've surely noticed that we usually have to upload our examples as static HTML files or use a third-party service like JSFiddle. Fortunately, there's another solution: we can publish them as WordPress custom post types. Let's see how.

First off all, you need two simple plugins to make things easier:

1. Advanced Custom Fields
2. Custom Post Type UI

These two plugins are a killer solution when it comes to productivity: they allow you to create custom fields and custom post types with ease. Once you've installed them, you need to create a custom post type for your examples.

We can call it `examples`. Then you need two custom fields to be associated with you new post type, one for the JavaScript code and the other one for the CSS code. Both fields must be of the type `textarea` and, most of all, should be parsed without additional formatting. When you've created these two fields, you have to associate them with the `examples` custom post type.

Now you can create your examples but you still need to publish them. You have to create a `single-examples.`php file in your theme with the following structure:

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>
<?php
global $post;
echo get_the_title($post->ID);
?>
</title>
<meta charset="utf-8" />
<style type="text/css" media="screen">
<?php
$css = get_post_meta($post->ID, 'styles', true);
echo $css;
?>
</style>
</head>

<body>


<?php if ( have_posts() ) while ( have_posts() ) :
the_post();

                the_content();

endwhile;
?>
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquer
y.min.js"></script>
<script type="text/javascript">
<?php
$js = get_post_meta($post->ID, 'js', true);
echo $js;
?>
</script>
```

```
</body>
</html>
```

We don't need the `get_header()` and `get_footer()` functions because our examples don't share the same document structure used in the rest of our site.

All we need are the post content (within the Loop), the title of the post and the two custom fields used to insert our JavaScript and CSS code. As you can see, I've used jQuery here but you can insert and use any kind of JavaScript library.