<?php /** * DokuWiki Plugin html5video (Syntax Component) * * @license GPL 2 http://www.gnu.org/licenses/gpl-2.0.html * @author Jason van Gumster (Fweeb) jason@monsterjavaguns.com * * Parts borrowed from the videogg plugin written by Ludovic Kiefer, * which is based on Christophe Benz' Dailymotion plugin, which, in turn, * is based on Ikuo Obataya's Youtube plugin. Whew... * * Currently only supports webm videos */

must be run within Dokuwiki if (!defined('DOKU_INC')) die(); if (!defined('DOKU_LF')) define('DOKU_LF', "\n"); if (!defined('DOKU_TAB')) define('DOKU_TAB', "\t"); if (!defined('DOKU_PLUGIN')) define('DOKU_PLUGIN',DOKU_INC.'lib/plugins/'); require_once DOKU_PLUGIN.'syntax.php'; class syntax_plugin_html5video_video extends DokuWiki_Syntax_Plugin { public function getType() { return 'substition'; } public function getPType() { return 'block'; } public function getSort() { return 159; } public function connectTo($mode) { Kind of a nasty regex, but it allows us to avoid using a

```
      //   plugin-specific prefix. We can use syntax directly from Media
      //   Manager.
      // http://gskinner.com/RegExr is a big help

$this->Lexer->addSpecialPattern('\{\{[^}]*(?:(?:webm)|(?:ogv)|(?:mp4))(?:\|(?
:\d{2,4}x\d{2,4})?(?:\|(?:loop)?,?(?:autoplay)?(?:,loop)?)?)?
?\}\}',$mode,'plugin_html5video_video');
  }
```

```
  public function handle($match, $state, $pos, &$handler){
      $params = substr($match, strlen('{{'), - strlen('}}')); //Strip markup
      if(strpos($params, ' ') === 0) { // Space as first character
          if(substr_compare($params, ' ', -1, 1) === 0) { // Space a front
and back = centered
              $params = trim($params);
              $params = 'center|' . $params;
          }
          else { // Only space at front = right-aligned
              $params = trim($params);
              $params = 'right|' . $params;
          }
      }
      elseif(substr_compare($params, ' ', -1, 1) === 0) { // Space only as
last character = left-aligned
          $params = trim($params);
          $params = 'left|' . $params;
      }
      else { // No space padding = inline
          $params = 'inline|' . $params;
      }
      return array(state, explode('|', $params));
  }
```

```
  public function render($mode, &$renderer, $data) {
      if($mode != 'xhtml') return false;
```

```
      list($state, $params) = $data;
```

```
      list($video_align, $video_url, $video_size, $video_attr) = $params;
```

```
      if($video_align == "center") {
          $align = "margin: 0 auto;";
      }
      elseif($video_align == "left") {
          $align = "float: left;";
      }
      elseif($video_align == "right") {
          $align = "float: right;";
      }
      else { // Inline
          $align = "";
      }
```

```
      if(!substr_count($video_url, '/')) {
          $video_url = ml($video_url);
      }
```

```
      if(substr($video_url, -4) != 'webm' && substr($video_url, -3) != 'ogv'
&& substr($video_url, -3) != 'mp4') {
          $renderer->doc .= "Error: The video must be in webm, ogv, or mp4
format.<br />" . $video_url;
          return false;
      }
```

```
      if(is_null($video_size) or !substr_count($video_size, 'x')) {
          $width  = 640;
          $height = 360;
      }
      else {
          $obj_dimensions = explode('x', $video_size);
          $width  = $obj_dimensions[0];
          $height = $obj_dimensions[1];
      }
```

```
      if(is_null($video_attr)) {
          $attr = "";
      }
      else {
          $arr_attr = explode(',', $video_attr);
          if(count($arr_attr) == 1) {
              if($arr_attr[0] == "loop") {
                  $attr = 'loop="loop"';
              }
              elseif($arr_attr[0] == "autoplay") {
                  $attr = 'autoplay="autoplay"';
              }
          }
          elseif(count($arr_attr) == 2) {
```

```
                if($arr_attr[0] != $arr_attr[1]) {
                    $attr = 'loop="loop" autoplay="autoplay"';
                }
                else {
                    $renderer->doc .= "Error: Duplicate parameters.<br />";
                    return false;
                }
            }
            else {
                $renderer->doc .= "Error: Wrong number of parameters.<br />";
                return false;
            }
        }
```

```
        $obj = '<video src="' . $video_url . '" width="' . $width . '"
height="' . $height . '" controls="controls" ' . $attr . '></video>';
        if($align != "") {
            $obj = '<div style="width: ' . $width . 'px; ' . $align . '">' .
$obj . '</div>';
        }
```

```
        $renderer->doc .= $obj;
        return true;
    }
    private function _getAlts($filename) {
        return false;
    }
```

```
}
```