Note для «ДокуВики»

—- plugin —- description: Этот плагин позволяет вам создавать на страницах DokuWiki такие красивые «заметки» author : Anael Mobilia email : contrib@anael.eu type : syntax lastupdate : 2024-04-15 compatible : Hogfather, Igor, Jack Jackrum, Kaos depends : conflicts : similar : bootswrapper, box, wrap, admnote tags : boxes, icons, toolbar

downloadurl: https://github.com/AnaelMobilia/dokuwiki-plugin-note/zipball/master bugtracker : https://github.com/AnaelMobilia/dokuwiki-plugin-note/issues sourcerepo : https://github.com/AnaelMobilia/dokuwiki-plugin-note donationurl :

screenshot_img : https://trello.com/1/cards/659ed9c093bc8f60bf5251a9/attachments/659eda3333beaab793b64a14/download/note-plugin-screenshot.png

---

Совместимость

- 2024-02-06 «Kaos» да
- 2023-04-04 «Jack Jackrum» да
- 2022-07-31 «Igor» да
- 2020-07-29 «Hogfather» да

Примечания

- В «пригодном для использования» шаблоне над примечанием имеется слишком много пробелов. Я смог решить эту проблему, закомментировав «clear: Both;» в «style.css». Ваш пробег может отличаться.
- С шаблоном «DokuCMS» и с «clear: Both;» в «style.css» примечания «смещаются» полем оглавления , а не перекрываются с ним [откровенный]

Монтаж

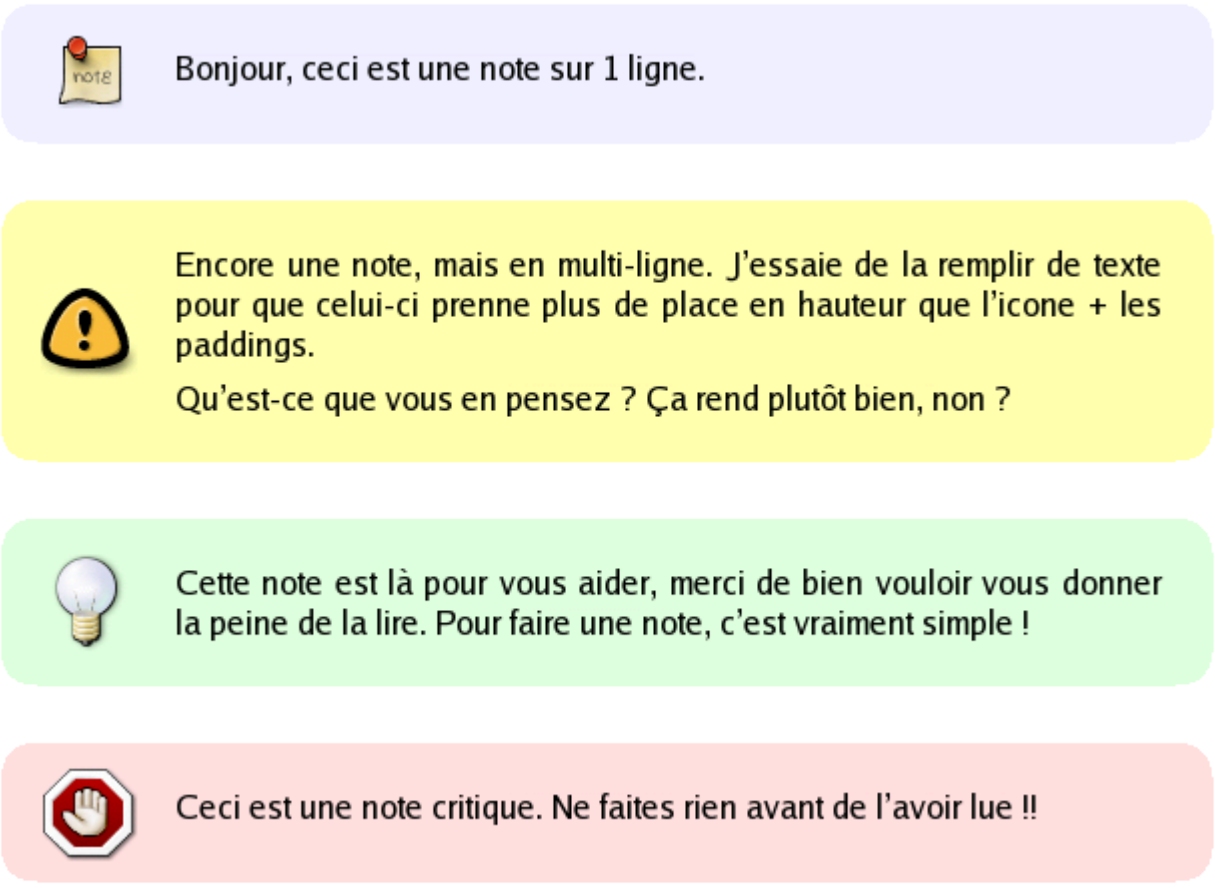
Используйте менеджер плагинов для установки или…

1. скачать zip-файл
2. распаковать его в<dokuwiki>/lib/plugins/
3. войдите как администратор и измените плагин в менеджере конфигурации

Применение

Когда он у вас установлен, используйте это:

```
<note>
This is my note! Remember it!!
</note>

<note important>
Warning! You're about to lose your mind ;-)
</note>

<note tip>
The clues are in the images.
</note>

<note warning>
Beware of the cat when you open the door!!
</note>
```



Вы можете использовать ключевые слова примечаний importantи warning. tipИли некоторые французские синонимы: importante(важно), bloquante, critique(предупреждение), tuyau, idée(совет), classique(классический). Добавлять новые типы заметок довольно легко, если вы немного разбираетесь в PHP и CSS .

Ограничения

Плагин не будет работать внутри нумерованного списка

~~Похоже, что он также не работает в сложенном разделе при использовании плагина « Fold ».~~

Исправление: Синтаксис следующий:

```
=== Note ===

++++Note dans un Folded|
<note critique>
Ça fonctionne seulement avec le //div// pas le //span//.
</note>
++++
```

Модификация

Основной файл внешнего вида плагина, это style.php. Отредактированно:

- Количество заметок увеличенно до 9 шт.
- Убраны иконки.
- Добавленна цветовая полоса слева.
- Уменьшино скругление углов.
- Увеличена ширена на длину страницы.

style.php

```
.plugin_note {
  margin: 2em auto; /*Отступ от левого края элемента*/
  max-width: 100%; /* Максимальная ширина страницы в процентах */
  min-height: 10px; /* Минимальная высота */
  text-align: justify; /*горизонтальное выравнивание текста в пределах элемента (justify - по ширине)*/
  vertical-align: middle;/*Выравнивание средней точки элемента по базовой линии родителя плюс половина высоты родительского элемента.*/
  border-collapse: collapse; /* отображать границы вокруг ячеек таблицы*/
  padding: 12px 17px 12px 27px; /*Устанавливает значение полей вокруг содержимого элемента*/
  /*background-position: 20px 50%;*/
  background-repeat: no-repeat;/*Определяет, как будет повторяться фоновое изображение (no-repeat
Устанавливает одно фоновое изображение)*/
  -moz-border-radius: 3px; /*Устанавливает набор цветов для нижней границы элемента*/
  -khtml-border-radius: 3px;
  border-radius: 3px;
  color: black;
  overflow: hidden;
}

.plugin_note .li {
  color: black !important;
```

```
}

.noteclassic {
  /*серый grey #708090  #dcdcdc*/ /*набор цветов*/
  /*background-image: url(images/note.png);*/ /*путь к иконкам - не
используется*/
  background: linear-gradient(
            to right,
            #708090 0%,
            #708090 1%,
            #dcdcdc 1%,
            #dcdcdc 100%
        );

}

.noteblue {
  /*синий blue #0000ff  #eeeeff*/
  /*background-image: url(images/note.png);*/
  background: linear-gradient(
            to right,
            blue 0%,
            blue 1%,
            #eeeeff 1%,
            #eeeeff 100%
        );
}

.noteyellow {
  /*желтый yellow #ffff00 #ffffcc*/
  /*background-image: url(images/important.png);*/
    background: linear-gradient(
            to right,
            #ffff00 0%,
            #ffff00 1%,
            #ffffcc 1%,
            #ffffcc 100%
        );
}

.notegreen {
  /*зеленый green #008000  #ddffdd*/
  /*background-image: url(images/warning.png);*/
    background: linear-gradient(
            to right,
            #008000 0%,
            #008000 1%,
            #ddffdd 1%,
            #ddffdd 100%
        );
}
```

```
.notered {
  /*красный red #ff0000* #ffdddd/
  /*background-image: url(images/tip.png);*/
    background: linear-gradient(
            to right,
            #ff0000 0%,
            #ff0000 1%,
            #ffdddd 1%,
            #ffdddd 100%
        );
}

.noteorange {
  /*оранжевый orange #ff8c00 #ffe2bd*/
  /*background-image: url(images/tip.png);*/
    background: linear-gradient(
            to right,
            #ff8c00 0%,
            #ff8c00 1%,
            #ffe2bd 1%,
            #ffe2bd 100%
        );
}

.notepurple {
  /*фиолетовый purple #800080 #ebc7eb*/
  /*background-image: url(images/tip.png);*/
    background: linear-gradient(
            to right,
            #800080 0%,
            #800080 1%,
            #ebc7eb 1%,
            #ebc7eb 100%
        );
}

.notebrown {
  /*коричневый brown #805c38  #f0d698*/
  /*background-image: url(images/tip.png);*/
    background: linear-gradient(
            to right,
            #805c38 0%,
            #805c38 1%,
            #f0d698 1%,
            #f0d698 100%
        );
}

.noteblack {
  /*черный black #000000  d3d3d3*/
```

```
  /*background-image: url(images/tip.png);*/
    background: linear-gradient(
            to right,
            black 0%,
            black 1%,
            #d3d3d3 1%,
            #d3d3d3 100%
        );
}


/*серый grey #708090  #dcdcdc*/
/*синий blue #0000ff  #eeeeff*/
/*желтый yellow #ffff00 #ffffcc*/
/*зеленый green #008000  #ddffdd*/
/*красный red #ff0000* #ffdddd/
/*оранжевый orange #ffda99 #ffe2bd*/
/*фиолетовый purple #800080 #ebc7eb*/
/*коричневый brown #805c38  #e8b37d*/
/*черный black #000000  e6e8e6*/
```

Основной файл меню выбора заметок плагина, это action.php. Отредактированно:

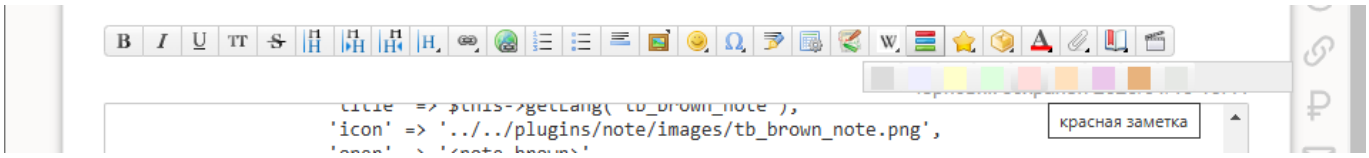
- Количество кнопок заметок увеличенно до 9 шт.
- Заменены иконки кнопок по цветовой схеме.

action.php

```
<?php
/**
 * @license    GPL 2 (http://www.gnu.org/licenses/gpl.html)
 */

class action_plugin_note extends DokuWiki_Action_Plugin
{

    /**
     * register the eventhandlers
     *
     * @author Andreas Gohr <andi@splitbrain.org>
     */
    function register(Doku_Event_Handler $controller)
    {
        $controller->register_hook('TOOLBAR_DEFINE', 'AFTER', $this,
'handle_toolbar', array());
    }

    function handle_toolbar(&$event, $param)
    {
        $event->data[] = array(
```

```
            'type' => 'picker',
            'title' => $this->getLang('note_picker'),
            'icon' => '../../plugins/note/images/note_picker.png',
            'list' => array(
                array(
                    'type' => 'format',
                    'title' => $this->getLang('tb_grey_note'),
                    'icon' =>
'../../plugins/note/images/tb_grey_note.png',
                    'open' => '<note>',
                    'close' => '</note>',
                ),
                array(
                    'type' => 'format',
                    'title' => $this->getLang('tb_blue_note'),
                    'icon' =>
'../../plugins/note/images/tb_blue_note.png',
                    'open' => '<note blue>',
                    'close' => '</note>',
                ),
                array(
                    'type' => 'format',
                    'title' => $this->getLang('tb_yellow_note'),
                    'icon' =>
'../../plugins/note/images/tb_yellow_note.png',
                    'open' => '<note yellow>',
                    'close' => '</note>',
                ),
                array(
                    'type' => 'format',
                    'title' => $this->getLang('tb_green_note'),
                    'icon' =>
'../../plugins/note/images/tb_green_note.png',
                    'open' => '<note green>',
                    'close' => '</note>',
                ),
                array(
                    'type' => 'format',
                    'title' => $this->getLang('tb_red_note'),
                    'icon' =>
'../../plugins/note/images/tb_red_note.png',
                    'open' => '<note red>',
                    'close' => '</note>',
                ),
                array(
                    'type' => 'format',
                    'title' => $this->getLang('tb_orange_note'),
                    'icon' =>
'../../plugins/note/images/tb_orange_note.png',
                    'open' => '<note orange>',
                    'close' => '</note>',
```

```
                ),
                array(
                    'type' => 'format',
                    'title' => $this->getLang('tb_purple_note'),
                    'icon' =>
'../../plugins/note/images/tb_purple_note.png',
                    'open' => '<note purple>',
                    'close' => '</note>',
                ),
                array(
                    'type' => 'format',
                    'title' => $this->getLang('tb_brown_note'),
                    'icon' =>
'../../plugins/note/images/tb_brown_note.png',
                    'open' => '<note brown>',
                    'close' => '</note>',
                ),
                array(
                    'type' => 'format',
                    'title' => $this->getLang('tb_black_note'),
                    'icon' =>
'../../plugins/note/images/tb_black_note.png',
                    'open' => '<note black>',
                    'close' => '</note>',
                ),
            )
        );
    }
}
```



Основной файл исполняемого кода, это syntax.php. Отредактированно (стр.40-50):

- Количество заметок увеличенно до 9 шт.
- Заменены названия заметок по цветовой схеме.

syntax.php

```
<?php
/**
 * Add Note capability to dokuwiki
 *
 * <note>This is note</note>
 * <note classic>This is note</note>
 * <note important>This is an important note</note>
 * <note warning>This is a big warning</note>
```

```
 * <note tip>This is a tip</note>
 *
 * by Olivier Cortès <olive@deep-ocean.net>
 * under the terms of the GNU GPL v2.
 *
 * Originaly derived from the work of :
 * Stephane Chamberland <stephane.chamberland@ec.gc.ca> (Side Notes
PlugIn)
 * Carl-Christian Salvesen <calle@ioslo.net> (Graphviz plugin)
 *
 * Contributions by Eric Hameleers <alien [at] slackware [dot] com> :
 *   use <div> instead of <table>,
 *   contain the images and stylesheet inside the plugin,
 *   permit nesting of notes,
 *
 * Contributed by Christopher Smith <chris [at] jalakai [dot] co [dot]
uk>
 *   fix some parsing problems and a security hole.
 *   make note types case independent
 *   simplify code reading
 *   modernise the plugin for changes/fixes/improvements to the
underlying Dokuwiki plugin class,
 *   improve efficiency.
 *
 * Contributed by Aurélien Bompard <aurelien [at] bompard [dot] org>
 *   support for the ODT output format.
 *
 * @license    GNU_GPL_v2
 * @author     Olivier Cortes <olive@deep-ocean.net>
 */

class syntax_plugin_note extends DokuWiki_Syntax_Plugin
{

    protected $notes = array(
        'noteblue' => array('blue', 'bleue'),
        'noteyellow' => array('yellow', 'jaune'),
        'notegreen' => array('green', 'verte'),
        'notered' => array('red', 'rouge'),
        'noteorange' => array('orange', 'orange'),
        'notepurple' => array('purple', 'violette'),
        'notebrown' => array('brown', 'marron'),
        'noteblack' => array('black', 'noire'),
        'noteclassic' => array('', 'classic', 'classique')
    );

    protected $default = 'plugin_note noteclassic';

    function getType()
    {
        return 'container';
```

```
    }

    function getPType()
    {
        return 'block';
    }

    function getAllowedTypes()
    {
        return array('container', 'substition', 'protected',
'disabled', 'formatting', 'paragraphs');
    }

    function getSort()
    {
        return 195;
    }

    // override default accepts() method to allow nesting
    // - ie, to get the plugin accepts its own entry syntax
    function accepts($mode)
    {
        if ($mode == substr(get_class($this), 7)) {
            return true;
        }
        return parent::accepts($mode);
    }

    function connectTo($mode)
    {
        $this->Lexer->addEntryPattern('<note.*?>(?=.*?</note>)', $mode,
'plugin_note');
    }

    function postConnect()
    {
        $this->Lexer->addExitPattern('</note>', 'plugin_note');
    }

    function handle($match, $state, $pos, Doku_Handler $handler)
    {
        switch ($state) {
            case DOKU_LEXER_ENTER:
                $note = strtolower(trim(substr($match, 5, -1)));

                foreach ($this->notes as $class => $names) {
                    if (in_array($note, $names)) {
                        return array($state, $class);
                    }
                }
                return array($state, $this->default);
```

```
            case DOKU_LEXER_UNMATCHED:
                return array($state, $match);

            default:
                return array($state, null);
        }
    }

    function render($format, Doku_Renderer $renderer, $indata)
    {
        if ($format == 'xhtml') {
            list($state, $data) = $indata;

            switch ($state) {
                case DOKU_LEXER_ENTER :
                    $renderer->doc .= '<div class="plugin_note ' .
htmlspecialchars($data) . '">';
                    break;

                case DOKU_LEXER_UNMATCHED :
                    $renderer->doc .= $renderer->_xmlEntities($data);
                    break;

                case DOKU_LEXER_EXIT :
                    $renderer->doc .= "\n</div>";
                    break;
            }
            return true;
        } elseif ($format == 'odt') {
            list($state, $data) = $indata;

            $this->render_odt($renderer, $state, $data);
            return true;
        }

        // unsupported $mode
        return false;
    }

    protected function render_odt($renderer, $state, $data)
    {
        static $first = true;
        static $new;

        if ($first == true) {
            $new = method_exists($renderer,
'getODTPropertiesFromElement');
            $first = false;
        }
```

```
        if (!$new) {
            // Render with older ODT plugin version.
            $this->render_odt_old($renderer, $state, $data);
        } else {
            // Render with newer ODT plugin version.
            $this->render_odt_new($renderer, $state, $data);
        }
    }

    protected function render_odt_old($renderer, $state, $data)
    {
        switch ($state) {
            case DOKU_LEXER_ENTER:
                $type = substr($data, 4);
                if ($type == 'classic') {
                    // The icon for classic notes is named note.png
                    $type = 'note';
                }
                $colors = array('note' => '#eeeeff', 'warning' =>
'#ffdddd', 'important' => '#ffffcc', 'tip' => '#ddffdd');

                // Content
                $properties = array();
                $properties ['width'] = '100%';
                $properties ['align'] = 'center';
                $properties ['shadow'] = '#808080 0.18cm 0.18cm';
                $renderer->_odtTableOpenUseProperties($properties);

                $properties = array();
                $properties ['width'] = '1.5cm';
$renderer->_odtTableAddColumnUseProperties($properties);

                $properties = array();
                $properties ['width'] = '13.5cm';
$renderer->_odtTableAddColumnUseProperties($properties);

                $renderer->tablerow_open();

                $properties = array();
                $properties ['vertical-align'] = 'middle';
                $properties ['text-align'] = 'center';
                $properties ['padding'] = '0.1cm';
                $properties ['border'] = '0.002cm solid #000000';
                $properties ['background-color'] = $colors[$type];
                $renderer->_odtTableCellOpenUseProperties($properties);

                $src = DOKU_PLUGIN . 'note/images/' . $type . '.png';
                $renderer->_odtAddImage($src);

                $renderer->tablecell_close();
```

```
                $properties = array();
                $properties ['vertical-align'] = 'middle';
                $properties ['padding'] = '0.3cm';
                $properties ['border'] = '0.002cm solid #000000';
                $properties ['background-color'] = $colors[$type];
                $renderer->_odtTableCellOpenUseProperties($properties);
                break;

            case DOKU_LEXER_UNMATCHED :
                $renderer->cdata($data);
                break;

            case DOKU_LEXER_EXIT :
                $renderer->tablecell_close();
                $renderer->tablerow_close();
                $renderer->_odtTableClose();
                $renderer->p_open();
                break;
        }
    }

    /**
     * ODT rendering for new versions of the ODT plugin.
     *
     * @param $renderer the renderer to use
     * @param $state    the current state
     * @param $data     data from handle()
     * @author LarsDW223
     */
    protected function render_odt_new($renderer, $state, $data)
    {
        switch ($state) {
            case DOKU_LEXER_ENTER:
                $css_properties = array();

                // Get CSS properties for ODT export.
                $renderer->getODTPropertiesNew($css_properties, 'div',
'class="' . $data . '"', null, true);

                // Create Content
                // (We only use the CSS parameters that are meaningful
for creating the ODT table)
                $properties = array();
                $properties ['width'] = '100%';
                $properties ['align'] = 'center';
                $properties ['shadow'] = '#808080 0.18cm 0.18cm';
                $renderer->_odtTableOpenUseProperties($properties);

                $properties = array();
                $properties ['width'] = '1.5cm';
$renderer->_odtTableAddColumnUseProperties($properties);
```

```
                $properties = array();
                $properties ['width'] = '13.5cm';
$renderer->_odtTableAddColumnUseProperties($properties);

                $renderer->tablerow_open();

                $properties = array();
                $properties ['vertical-align'] = $css_properties
['vertical-align'];
                $properties ['text-align'] = 'center';
                $properties ['padding'] = '0.1cm';
                $properties ['border'] = '0.002cm solid #000000';
                $properties ['background-color'] = $css_properties
['background-color'];
                $renderer->_odtTableCellOpenUseProperties($properties);

                if ($css_properties ['background-image']) {
                    $renderer->_odtAddImage($css_properties
['background-image']);
                }

                $renderer->tablecell_close();

                $properties = array();
                $properties ['vertical-align'] = $css_properties
['vertical-align'];
                $properties ['text-align'] = $css_properties ['text-
align'];
                $properties ['padding'] = '0.3cm';
                $properties ['border'] = '0.002cm solid #000000';
                $properties ['background-color'] = $css_properties
['background-color'];
                $renderer->_odtTableCellOpenUseProperties($properties);
                break;

            case DOKU_LEXER_UNMATCHED :
                $renderer->cdata($data);
                break;

            case DOKU_LEXER_EXIT :
                $renderer->tablecell_close();
                $renderer->tablerow_close();
                $renderer->_odtTableClose();
                $renderer->p_open();
                break;
        }
    }
}
```

Основной файл перевода, это lang.php. Отредактированно :

- Файл «en»
- Файл «fr»
- Файл «ru»
- Количество заметок увеличенно до 9 шт.
- Заменены названия заметок по цветовой схеме.

lang.php

```
<?php
/**
 * English language file
 */

$lang['note_picker']  = 'Notes';

$lang['tb_grey_note']      = 'grey note';
$lang['tb_blue_note']       = 'blue_note';
$lang['tb_yellow_note'] = 'yellow note';
$lang['tb_green_note']   = 'green note';
$lang['tb_red_note']      = 'red note';
$lang['tb_orange_note']       = 'orange note';
$lang['tb_purple_note'] = 'purple note';
$lang['tb_brown_note']   = 'brown note';
$lang['tb_black_note']   = 'black note';
```

lang.php

```
<?php
/**
 * Russian language file
 */

$lang['note_picker'] = 'Заметки';

$lang['tb_grey_note'] = 'серая заметка';
$lang['tb_blue_note'] = 'синяя заметка';
$lang['tb_yellow_note'] = 'желтая заметка';
$lang['tb_green_note'] = 'зеленая заметка';
$lang['tb_red_note'] = 'красная заметка';
$lang['tb_orange_note'] = 'оранжевая заметка';
$lang['tb_purple_note'] = 'фиолетовая заметка';
$lang['tb_brown_note'] = 'коричневая заметка';
$lang['tb_black_note']   = 'черная заметка';
```

Пример № 1 отображения ссылок отредактированного плагина в русской локализации

Свойства CSS marginиспользуются для создания пространства вокруг элементов за пределами определенных границ. С CSS у вас есть полный контроль над полями. Существуют свойства для установки полей для каждой стороны элемента (сверху, справа, снизу и слева).

Свойства CSS marginиспользуются для создания пространства вокруг элементов за пределами определенных границ. С CSS у вас есть полный контроль над полями. Существуют свойства для установки полей для каждой стороны элемента (сверху, справа, снизу и слева).

Свойства CSS marginиспользуются для создания пространства вокруг элементов за пределами определенных границ. С CSS у вас есть полный контроль над полями. Существуют свойства для установки полей для каждой стороны элемента (сверху, справа, снизу и слева).

Свойства CSS marginиспользуются для создания пространства вокруг элементов за пределами определенных границ. С CSS у вас есть полный контроль над полями. Существуют свойства для установки полей для каждой стороны элемента (сверху, справа, снизу и слева).

Свойства CSS marginиспользуются для создания пространства вокруг элементов за пределами определенных границ. С CSS у вас есть полный контроль над полями. Существуют свойства для установки полей для каждой стороны элемента (сверху, справа, снизу и слева).

Свойства CSS marginиспользуются для создания пространства вокруг элементов за пределами определенных границ. С CSS у вас есть полный контроль над полями. Существуют свойства для установки полей для каждой стороны элемента (сверху, справа, снизу и слева).

Свойства CSS marginиспользуются для создания пространства вокруг элементов за пределами определенных границ. С CSS у вас есть полный контроль над полями. Существуют свойства для установки полей для каждой стороны элемента (сверху, справа, снизу и слева).

Свойства CSS marginиспользуются для создания пространства вокруг элементов за пределами определенных границ. С CSS у вас есть полный контроль над полями. Существуют свойства для установки полей для каждой стороны элемента (сверху, справа, снизу и слева).

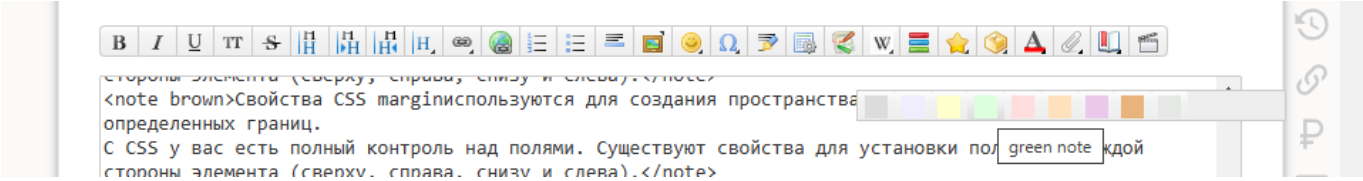
Пример № 2 отображения ссылок отредактированного плагина в русской локализации

зеленая заметка

оранжевая заметка

Свойства CSS margin используются для создания пространства вокруг элементов за пределами определенных границ. С CSS у вас есть полный контроль над полями. Существуют свойства для установки полей для каждой стороны элемента (сверху, справа, снизу и слева)

Пример № 3 отображения ссылок отредактированного плагина в английской локализации



yellow note

orange note

black note

Ссылка на

загрузку

модефицированного плагина и в конце статьи в разделе «Дополнения и Файлы» с пометкой «mod»

Обновление

Для обновления удалите исходную папку lib/plugins/displaywikipage и установите новую версию, как указано выше.

Файл для загрузки

- plugin note
- plugin note mode

Лицензия

- **Author:** Anael Mobilia
- **License:** GNU GPL v2

From:
https://wwoss.ru/ - **worldwide open-source software**

Permanent link:
https://wwoss.ru/doku.php?id=wiki:plugin:note&rev=1738952621

Last update: **2025/02/07 21:23**