Aide:Syntaxe:Blocs de code

Vous pouvez inclure des blocs non analysés dans vos documents soit en les indentant avec au moins deux espaces (comme on peut le voir dans les exemples précédents) soit en utilisant les balises <code> ou <file>.

```
Ce texte est indenté avec deux espaces.
```

```
Ceci est du code préformaté, tous les espaces sont préservés :
comme           <- ceci
```

```
C'est pratiquement la même chose, mais vous pouvez l'utiliser pour montrer
que vous avez cité un fichier.
```

```
  Ce texte est indenté avec deux espaces.

<code>
Ceci est du code préformaté, tous les espaces sont préservés :
comme           <- ceci
</code>

<file>
C'est pratiquement la même chose, mais vous pouvez l'utiliser pour montrer
que vous avez cité un fichier.
</file>
```

Coloration syntaxique

DokuWiki peut mettre en forme et en couleur du code source, ce qui facilite sa lecture. Il utilise le Generic Syntax Highlighter GeSHi – donc n'importe quel langage connu de GeSHi est accepté. La syntaxe est la même que dans le bloc de code de la section précédente, mais cette fois le nom du langage utilisé est inséré dans la balise. Par exemple : <code java>.

```java
/**
 * La classe HelloWorldApp implémente une application qui
 * affiche simplement "Hello World!" dans la sortie standard.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); //Affiche la chaîne.
    }
}
```

DokuWiki reconnait les identifiants de langage suivants : *4cs*, *6502acme*, *6502kickass*, *6502tasm*, *68000devpac*, *abap*, *actionscript*, *actionscript3*, *ada*, *aimms*, *algol68*, *apache*, *applescript*, *apt_sources*, *arm*, *asm*, *asp*, *asymptote*, *autoconf*, *autohotkey*, *autoit*, *avisynth*, *awk*, *bascomavr*, *bash*, *basic4gl*, *batch*, *bf*, *biblatex*, *bibtex*, *blitzbasic*, *bnf*, *boo*, *c*, *caddcl*, *cadlisp*, *ceylon*, *cfdg*, *cfm*,

chaiscript, *chapel*, *cil*, *c_loadrunner*, *clojure*, *c_mac*, *cmake*, *cobol*, *coffeescript*, *cpp*, *cpp-qt*, *cpp-winapi*, *csharp*, *css*, *cuesheet*, *c_winapi*, *d*, *dart*, *dcl*, *dcpu16*, *dcs*, *delphi*, *diff*, *div*, *dos*, *dot*, *e*, *ecmascript*, *eiffel*, *email*, *epc*, *erlang*, *euphoria*, *ezt*, *f1*, *falcon*, *fo*, *fortran*, *freebasic*, *freeswitch*, *fsharp*, *gambas*, *gdb*, *genero*, *genie*, *gettext*, *glsl*, *gml*, *gnuplot*, *go*, *groovy*, *gwbasic*, *haskell*, *haxe*, *hicest*, *hq9plus*, *html*, *html4strict*, *html5*, *icon*, *idl*, *ini*, *inno*, *intercal*, *io*, *ispfpanel*, *j*, *java*, *java5*, *javascript*, *jcl*, *jquery*, *julia*, *kixtart*, *klonec*, *klonecpp*, *kotlin*, *latex*, *lb*, *ldif*, *lisp*, *llvm*, *locobasic*, *logtalk*, *lolcode*, *lotusformulas*, *lotusscript*, *lscript*, *lsl2*, *lua*, *m68k*, *magiksf*, *make*, *mapbasic*, *mathematica*, *matlab*, *mercury*, *metapost*, *mirc*, *mk-61*, *mmix*, *modula2*, *modula3*, *mpasm*, *mxml*, *mysql*, *nagios*, *netrexx*, *newlisp*, *nginx*, *nimrod*, *nsis*, *oberon2*, *objc*, *objeck*, *ocaml*, *ocaml-brief*, *octave*, *oobas*, *oorexx*, *oracle11*, *oracle8*, *oxygene*, *oz*, *parasail*, *parigp*, *pascal*, *pcre*, *per*, *perl*, *perl6*, *pf*, *phix*, *php*, *php-brief*, *pic16*, *pike*, *pixelbender*, *pli*, *plsql*, *postgresql*, *postscript*, *povray*, *powerbuilder*, *powershell*, *proftpd*, *progress*, *prolog*, *properties*, *providex*, *purebasic*, *pycon*, *pys60*, *python*, *q*, *qbasic*, *qml*, *racket*, *rails*, *rbs*, *rebol*, *reg*, *rexx*, *robots*, *rpmspec*, *rsplus*, *ruby*, *rust*, *sas*, *sass*, *scala*, *scheme*, *scilab*, *scl*, *sdlbasic*, *smalltalk*, *smarty*, *spark*, *sparql*, *sql*, *standardml*, *stonescript*, *swift*, *systemverilog*, *tcl*, *tclegg*, *teraterm*, *texgraph*, *text*, *thinbasic*, *tsql*, *twig*, *typoscript*, *unicon*, *upc*, *urbi*, *uscript*, *vala*, *vb*, *vbnet*, *vbscript*, *vedit*, *verilog*, *vhdl*, *vim*, *visualfoxpro*, *visualprolog*, *whitespace*, *whois*, *winbatch*, *xbasic*, *xml*, *xojo*, *xorg_conf*, *xpp*, *yaml*, *z80*, *zxbasic*.

Numérotation des lignes

Le code suivant active la numérotation des lignes:

```
<code JavaScript [enable_line_numbers="true"]>
var de = function() {
    return (typeof(window.de) == 'object') ? window.de : {};
}();
</code>
```

Voici le résultat:

```
 1. var de = function() {
 2.     return (typeof(window.de) == 'object') ? window.de : {};
 3. }();
```

La numérotation peut commencer à la valeur donnée:

```
<code C [enable_line_numbers="true", start_line_numbers_at="42"]>
void main () {
    printf ("Hello World!");
    exit 0;
}
</code>
```

Voici le résultat:

```
42. void main () {
43.     printf ("Hello World!");
```

```
44.     exit 0;
45. }
```

Mettre des lignes en évidence

L'exemple suivant met en plus en évidence une ligne:

```
<code JavaScript [enable_line_numbers="true",highlight_lines_extra="2"]>
var de = function() {
    return (typeof(window.de) == 'object') ? window.de : {};
}();
</code>
```

Voici le résultat:

```
1. var de = function() {
2.     return (typeof(window.de) == 'object') ? window.de : {};
3. }();
```

Vous pouvez aussi mettre en évidence plusieurs lignes:

```
<code JavaScript [enable_line_numbers="true",highlight_lines_extra="2,3"]>
var de = function() {
    return (typeof(window.de) == 'object') ? window.de : {};
}();
</code>
```

Voici le résultat:

```
1. var de = function() {
2.     return (typeof(window.de) == 'object') ? window.de : {};
3. }();
```

Bloc de code téléchargeables

Quand vous utilisez les balises *<code>* ou *<file>*, vous pouvez rendre disponible en téléchargement le code affiché. Il faut alors préciser un nom de fichier juste après le code du langage.

monexemple.php

```
<?php echo "hello world!"; ?>
```

```
<file php monexemple.php>
<?php echo "hello world!"; ?>
</file>
```

Si vous ne voulez pas de coloration syntaxique, il suffit de fournir un tiret (-) en guise de code de langage :
`<code - monfichier.toto>`.

Extension

L'extension CLI (CLI plugin) affiche du texte comme dans une interface en lignes de commande. Il s'utilise avec les balises `<cli>...</cli>`.