

Écrans de terminal

Ces modèles sont à utiliser lorsque vous souhaitez illustrer:

- Une commande entrée par l'administrateur système (root) avec le résultat obtenu
- Un extrait d'affichage sur un terminal

Pour le moment, les seuls modèles sont un écran à texte orange sur fond noir et leur largeur est exprimée en pourcentage par rapport à la largeur de page dans le fichier CSS du thème du wiki 😊

Terminal avec invite 'root' avec affichage du résultat

Le modèle [Modèle:TerminalRoot](#) sert à afficher des commandes comportant un résultat d'exécution. Le modèle supporte les paramètres suivants:

- **invite:** Le texte de l'invite (ce qui précède le caractère '#' de l'invite)
- **commande:** La commande entrée (est mise en évidence)
- **resultat:** Le résultat de la commande en texte brut (les espacements et retours à la ligne sont conservés). Noter que le texte est encadré par les balises ouvrante et fermante *nowiki*.

```
{{TerminalRoot|invite=linux-desktop01 ~ |commande=cat /proc/partitions | grep
sdc|resultat=</nowiki>
 8  32  7782400 sdc
 8   33   97656 sdc1
 8   34  1832631 sdc2</nowiki>}}
```

Produira:

```
<pre<noinclude></noinclude> class="terminal">linux-desktop01 ~# cat
/proc/partitions</pre<noinclude></noinclude>>
```

```
<pre<noinclude></noinclude> class="resultatterminal"> 8 32 7782400 sdc 8 33 97656 sdc1 8 34
1832631 sdc2</pre<noinclude></noinclude>>
```

Rappel: Notez l'usage des balises ouvrante et fermante *nowiki* au début et à la fin du contenu à citer !

Terminal avec invite 'root' sans affichage du résultat

L'usage est similaire à celui décrit au paragraphe précédent excepté que le paramètre commande n'est pas renseigné. Ainsi :

```
{{TerminalRoot|invite=linux-desktop01 ~ |commande=cat /dev/zero> /dev/null}}
```

Produira :

```
<pre<noinclude></noinclude> class="terminal">linux-desktop01 ~# cat /dev/zero >
```

```
</dev/null</pre></noinclude></noinclude>>
```

```
<pre></noinclude></noinclude> class="resultatterminal"> </pre></noinclude></noinclude>>
```

Rappel: Notez l'usage des balises ouvrante et fermante *nowiki* au début et à la fin du contenu à citer !

Extrait d'affichage de terminal

Le modèle [Modèle:ExtraitTerminal](#) d'afficher un extrait de terminal sans avoir à y faire figurer de commande comme dans les paragraphes précédents. Le modèle supporte les paramètres suivants:

- **resultat:** L'extrait à afficher en texte brut (les espacements et retours à la ligne sont conservés). Noter que le texte est encadré par les balises ouvrante et fermante *nowiki*.

Par exemple pour illustrer l'affichage :

```
[ 0.000000] PLATFORM: banner-name [Sun Fire(TM) T1000]
[ 0.000000] PLATFORM: name [SUNW,Sun-Fire-T1000]
[ 0.000000] Unable to handle kernel NULL pointer dereference
[ 0.000000] tsk->{mm,active_mm}->context = 0000000000000000
[ 0.000000] tsk->{mm,active_mm}->pgd = fffff800010d46dc
[ 0.000000]   %%\%%|/ ____ %%\%%|/
[ 0.000000]           "@'/ .. %%\%%`@"
[ 0.000000]           /_| %%\%%_/ |_%%\%%
[ 0.000000]           %%\%%__U_/
[ 0.000000] swapper(0): Oops [#1]
[ 0.000000] TSTATE: 0000009980e01604 TPC: 00000000008a9714 TNPC:
00000000008a9718 Y: 00000000 Not tainted
[ 0.000000] TPC: <sun4v_mdsc_init+0x1cc/0x2ac>
[ 0.000000] g0: 00000000008c8748 g1: 0000000000000045 g2: 0000000000000002
g3: 0000000000000074
[ 0.000000] g4: 0000000000848368 g5: 0000000000000000 g6: 000000000083c000
g7: 0000000000000000
```

Il faudra utiliser insérer le code suivant dans l'article du wiki:

```
{{ExtraitTerminal|resultat=<nowiki>
[ 0.000000] PLATFORM: banner-name [Sun Fire(TM) T1000]
[ 0.000000] PLATFORM: name [SUNW,Sun-Fire-T1000]
[ 0.000000] Unable to handle kernel NULL pointer dereference
[ 0.000000] tsk->{mm,active_mm}->context = 0000000000000000
[ 0.000000] tsk->{mm,active_mm}->pgd = fffff800010d46dc
[ 0.000000]   %%\%%|/ ____ %%\%%|/
[ 0.000000]           "@'/ .. %%\%%`@"
[ 0.000000]           /_| %%\%%_/ |_%%\%%
[ 0.000000]           %%\%%__U_/
[ 0.000000] swapper(0): Oops [#1]
```

```
[ 0.000000] TSTATE: 0000009980e01604 TPC: 00000000008a9714 TNPC:
00000000008a9718 Y: 00000000 Not tainted
[ 0.000000] TPC: <sun4v_mdsc_init+0x1cc/0x2ac>
[ 0.000000] g0: 00000000008c8748 g1: 0000000000000045 g2: 0000000000000002
g3: 0000000000000074
[ 0.000000] g4: 0000000000848368 g5: 0000000000000000 g6: 000000000083c000
g7: 0000000000000000</nowiki>}}
```

Ce qui donnera :

BSoD

Inutile de préciser à quoi le modèle [Modèle:BSoD](#) sert... 😊 Il s'utilise de la même manière que [Modèle:ExtraitTerminal](#) :

```
{{BSoD|resultat=
Un problème a été détecté et Linux a été arrêté afin de prévenir tout dommage
sur votre ordinateur.

KERNEL_PRIV_LESS_OR_EQUAL

Le module noyau <inconnu> a été placé dans l'état <inconnu> du fait de
l'erreur <non spécifiée>
}}
```

Donnera:

```
<pre<noinclude></noinclude> class="bsod"> Un problème a été détecté et Linux a été arrêté afin de
prévenir tout dommage sur votre ordinateur. KERNEL_PRIV_LESS_OR_EQUAL Le module noyau
<inconnu> a été placé dans l'état <inconnu> du fait de l'erreur <non spécifiée>
</pre<noinclude></noinclude>
```

Listage de code source

Avec numéro de ligne

Pour afficher un code source dont les lignes sont numérotées vous devez utiliser le [Modèle:FichierSourceAvecNumeros](#). Ce modèle utilise en arrière plan l'extention wikimedia [SyntaxHighlight_GeSHi](#) et supporte 5 paramètres

- **nom** : Nom du fichier figurant dans le pied de l'encadré ;
- **langage** : Langage source du contenu à afficher (voir ci-après) ;
- **nolignedebut** : Numéro affiché pour la première ligne du contenu (par défaut: 1)
- **nolignesurlignee** : Rangée la ligne à mettre en surbrillance (par défaut: ne rien mettre en surbrillance) ;
- **contenu** : Texte à formater. **Il n'est pas nécessaire de l'encadrer des balises**

La liste des langages supportés par la version sur la dompe de LinuQ est:

Code	Language
abap	ABAP
actionscript	ActionScript (Adobe Flash)
ada	Ada
apache	Fichier de configuration Apache
applescript	Code en langage AppleScript
asm	Code en langage assembleur
asp	Active Server Pages (ASP)
autoit	AutoIt
bash	Script Shell Bash
basic4gl	Code en langage Basic4GL
bf	Code en langage Brainfuck
blitzbasic	Code BASIC Blitz
bnf	Notation en format Backus-Naur
c	Code en langage Langage C
c_mac	C (Mac)
caddcl	AutoCAD Dialog Control Language (AutoCAD DCL)
cadlisp	AutoLISP
cfg	CFDG
cfm	ColdFusion Markup
cil	Code en Common Intermediate Language (Plateforme .Net)
cobol	Code en langage COBOL
cpp-qt	Bibliothèque C++ Qt
cpp	Code en langage C++
csharp	Code en langage C# (Plateforme .Net)
css	Feuille de style CSS (Cascading Style Sheets)
d	Code en langage D
delphi	Code Delphi
diff	Diff
div	DIV
dos	Fichier de commande en lot (batch) DOS
dot	Code DOT
eiffel	Code Eiffel
fortran	Code Fortran
freebasic	Code FreeBASIC
genero	Genero
gettext	Bibliothèque d'internationalisation GNU (i18n)
glsl	Code OpenGL Shading Language (GLSL)
gml	Code Game Maker Language (GML)
gnuplot	Code Gnuplot
groovy	Code Groovy

Code	Language
haskell	Code Haskell
hq9plus	HQ9+
html4strict	Code HTML
html5	Code HTML 5
idl	Universal Network Objects (Uno IDL)
ini	Fichier INI (Microsoft Windows)
inno	Inno Setup
intercal	INTERCAL
io	Code Io
java	Code Java
java5	Code Java (Java(TM) 2 Platform Standard Edition 5.0)
javascript	Code JavaScript
kixtart	KiXtart
klonec	Code Klone C
klonecpp	Code Klone C++
latex	Code LaTeX
lisp	Code Lisp
lolcode	LOLCODE
lotusscript	LotusScript (IBM Lotus Notes)
lua	Code Lua

Code	Language
m68k	Assembleur Motorola 68000
make	Makefile
matlab	Fichier MATLAB
mircc	Script mIRC
mxml	MXML
mpasm	Code Assembleur PIC (microcontrôleur)
mysql	Code MySQL
nsis	Script Nullsoft Scriptable Install System (NSIS)
objc	Code Objective-C
ocaml-brief	Code Objective Caml (OCaml)
ocaml	Code Objective Caml (OCaml)
oobas	Macro en BASIC StarOffice/OpenOffice.org/LibreOffice.org
oracle8	Script Oracle PL/SQL Oracle (Oracle 8)
oracle11	Script Oracle PL/SQL Oracle (Oracle 11)
pascal	Code Pascal
per	per
perl	Code Perl
php-brief	Code PHP
php	Code PHP
pixelbender	Adobe Pixel Bender
plsql	Script Oracle PL/SQL

Code	Language
povray	Code Persistence of Vision Raytracer (POV-Ray)
powershell	Script Microsoft Windows PowerShell
progress	Code OpenEdge Advanced Business Language
prolog	Code Prolog
providex	ProvideX
python	Code Python
qbasic	Code Microsoft QuickBASIC/QBasic
rails	Code Ruby on Rails
reg	Fichier de registre Windows
robots	Fichier d'exclusion d'engin d'indexation Web automatique (robots.txt)
ruby	Code Ruby
sas	Requête SAS
scala	Code Scala
scheme	Code Scheme
scilab	Scilab
sdlbasic	Code SdlBasic
smalltalk	Code Smalltalk
smarty	Smarty
sql	Requête SQL
tcl	Code TCL
teraterm	TeraTerm
text	Texte brut
thinbasic	Code thinBasic
tsql	Code Transact-SQL (Microsoft SQL Server)
typoscript	Code TYPO3
vb	Code Visual Basic
vbnet	Code Visual Basic .NET
verilog	Code Verilog
vhdl	Code VHSIC Hardware Description Language (VHDL)
vim	Script Vim
visualfoxpro	Code Visual FoxPro
visualprolog	Code Visual Prolog
whitespace	Code en langage Whitespace
winbatch	Winbatch
xml	XML
xorg_conf	Fichier de configuration Xorg
xpp	Microsoft Dynamics AX (X++)
z80	Code en assembleur Zilog Z80

Cas d'utilisation: formater le code C du fichier toto.c suivant en commençant la numérotation à 55 et en mettant en surbrillance la 5ème ligne

- **nolignesurlignee** : Rangée la ligne à mettre en surbrillance (par défaut: ne rien mettre en surbrillance) ;
- **contenu** : Texte à formater. **Il n'est pas nécessaire de l'encadrer des balises**

Cas d'utilisation: formater le code C du fichier toto.c sans afficher de numéros et en mettant en surbrillance la 5ème ligne:

```
#include <stdio.h>

int main(int argc, char %*%*%*%*%*%argv)
{
    printf("Hello world%%\%%n");    <---- Surlicer cette ligne
    return 0;
}
```

Pour ce faire insérer directement le code suivant dans l'article:

```
{{FichierSourceSansNumeros|nom=toto.c|langage=c|nolignesurlignee=5|contenu=
#include <stdio.h>

int main(int argc, char %*%*%*%*%*%argv)
{
    printf("Hello world%%\%%n");
    return 0;
} }}
```

Ce qui donnera :

```
#include <stdio.h>

intmain(intargc,char**argv)
{
printf("Hello world\n" );
return0;
}
```

Fichier toto.c

Boîtes de messages

Message informatif

Cette boîte de message s'utilise pour glisser une note au lecteur à caractère **informatif** au lecteur. Le texte qui y figure fournit une information additionnelle secondaire qu'il est utile d'évoquer sans pour autant la noyer dans le texte principale.

<label_boite_information> Information </label_boite_information>

Une petite note **avec** un peu *de mise en forme*. Il est possible:

- de glisser un petit point de forme
 - un second
- et un dernier !

Message d'avertissement (« Attention! »)

Cette boîte de message s'utilise pour glisser une note au lecteur à caractère de recommandation facultative ou d'avertissement de la possibilité d'un dysfonctionnement mineur. au lecteur. Le texte qui y figure fournit une information additionnelle importante mais qui n'a pas de conséquences graves si le lecteur choisit de ne pas y prêter attention et de passer outre. **Les opérations pouvant occasionner des pertes de données ou des dommages matériels doivent impérativement être signalées avec la boîte de message « Critique »** (voir paragraphe suivant).

<label_boite_attention> Attention </label_boite_attention>

Une petite note **avec** un peu *de mise en forme*. Il est possible:

- de glisser un petit point de forme
 - un second
- et un dernier !

Message critique (conséquences majeures)

Cette boîte de message s'utilise pour glisser une note au lecteur à **caractère de recommandation devant être obligatoirement suivie ou une information indispensable à connaître sous peine de dysfonctionnement majeur**. Au point extrême, les conséquences pour le lecteur peuvent être même catastrophiques : bris matériel, corruption de données (perte!), dommage difficilement réparable voire totalement irréparable.

<label_boite_critique> Critique </label_boite_critique>

Une petite note **avec** un peu *de mise en forme*. Il est possible:

- de glisser un petit point de forme
 - un second
- et un dernier !

From:
<https://linuq.org/> - LinuQ: Logiciels libres à Québec

Permanent link:
https://linuq.org/wiki/utilisation_des_gabarits

Last update: **2018/09/28 17:37**



