

CIRAD
Département BIOS, Unité DAP, Équipe ID

Valentin GUIGNON, 02/03/2009
Version 1.0.0

CONVENTIONS DE PROGRAMMATION EN PERL
RÉCAPITULATIF

Sommaire

1. Récapitulatif.....	3
-----------------------	---

1. Récapitulatif

- utiliser les squelettes « Template.pl », « Template.cgi » et « Template.pm »;
- la rédaction du code se fait intégralement en anglais international dans un souci d'ouverture du code à la communauté internationale.

☺ Bien	☹ Pas bien
nommage fichiers:	
<code>my_program.pl # minuscules et underscores</code>	
<code>SomeModule.pm # même nom que le package</code>	
<code>hello world.cgi # minuscules et underscores</code>	
nommage constantes:	
-nom long	
-majuscules et underscores	
-documenter l'usage	<code>my \$MIN; # trop générique et court</code>
-si path, documenter si "/" final ou non	<code>my \$CircleCoordinatesX; # minuscules!</code>
<code>my \$MIN_CELL_X = 25;</code>	
nommage variables:	
-nom long	<code>my \$cx; # trop court</code>
-minuscules et underscores	<code>my \$Circle_Coordinates_X; # majuscules!</code>
-ne commence jamais par un verbe	
<code>my \$circle_label_coordinate_x;</code>	
nommage des packages:	
-mots collés avec première lettre en majuscule	
<code>package PackageName;</code>	
nommage de fonctions:	
-commence par un verbe sans majuscule	
-1 majuscule par mot et sans underscores	
<code>sub doTheFunction</code>	
nommage d'étiquettes:	
<code>LABEL_1:</code>	
nommage de blocs:	
-numéro de ligne initial	
-nombre de lignes (facultatif)	
-nom en majuscules et underscores	
-3 underscores avant et après	
<code>___1234_3_WELCOME_MESSAGE___</code>	
une ligne, une expression	
<code>confess, cluck</code>	<code>die, warn</code>
@_ en une ligne plutôt que shift:	
<code>my (\$self, \$arg1, \$arg2, \$arg3) = @_;</code>	<code>my \$self = shift;</code> <code>my \$arg1 = shift;</code>
acolades style BSD	
<code>sub doTest</code>	<code>else/elsif seulement en début de ligne:</code>
<code>{</code>	<code>} else {</code>
<code> if (...)</code>	
<code> {</code>	
<code> ...</code>	
<code> }</code>	
<code> else</code>	
<code> {</code>	
<code> ...</code>	
<code> }</code>	
<code>}</code>	
pas d'accolade fermante sur une ligne d'instruction:	
	<code>... }</code>
ou (exclusif) accolades style K&R	
<code>sub doTest {</code>	
<code> if (...) {</code>	
<code> ...</code>	
<code> }</code>	
<code> else {</code>	
<code> ...</code>	
<code> }</code>	
<code>}</code>	
espaces et indentation:	
indentation 4 espaces	ne pas utiliser les tabulations
<code>while (0 == \$variable)</code>	<code>while(0 == \$variable)</code>

CIRAD
Département BIOS, Unité DAP, Équipe ID

<pre>if (\$self->doTest()) print "toto" . "tutu" . "tata"; \$self->doMyFunc(\$argument_number_one, \$argument_number_two, \$last_argument); if ((\$self->doMyTest1WhichHasALongName()) (\$self->doTest2() and \$self->doTest3() and \$self->doTest4())) print "tototututatititi" . "popopupupapapipi" . "lololululalalili";</pre>	<pre>if (\$self->doTest ()) print "toto"."tutu"."tata"; \$self->doMyFunc(\$argument_number_one, \$argument_number_two, \$last_argument); if ((\$self->doMyTest1WhichHasALongName()) (\$self->doTest2() and \$self->doTest3() and \$self->doTest4())) print "tototututatititi" . "popopupupapapipi" . "lololululalalili";</pre>
indentation spéciale pour last et next:	
<pre>LINE: for (;;) { ... # some code --\$toto; last LINE if (0 == \$toto); next LINE if /^#/; \$tutu = \$self->doSomething(); ... # some other code }</pre>	<pre>if ((0 != \$toto) (0 >= \$tutu)) if ((\$toto != 0) (\$tutu <= 0)) if (\$toto (0 >= \$tutu)) if (0 != \$toto 0 >= \$tutu)</pre>
incrémenteur préfixe:	
<pre>while (\$MAX_VALUE > \$i++) # postfix souvent ok for (my \$i = 0; \$MAX_VALUE > \$i; ++\$i) ++j;</pre>	<pre>for (my \$i = 0; \$MAX_VALUE > \$i; \$i++) j++;</pre>
passage de paramètres optionnels:	
<pre>doSomeStuff({'highlight' => 1, 'text' => 'Bla'}); doSomeStuff({'text' => 'Bla', 'highlight' => 1}); doSomeStuff({'text' => 'Bla'});</pre>	<pre>non recommandé: doSomeStuff(1, 'Bla'); doSomeStuff(undef, 'Bla'); doSomeStuff('Bla');</pre>
ouverture de fichiers:	
<pre>if (open(FH, "<myfile.ext"))</pre>	<pre>open(FH, "myfile.ext");</pre>