NAME

perlcheat – Perl 5 Cheat Sheet

DESCRIPTION

This 'cheat sheet' is a handy reference, meant for beginning Perl programmers. Not everything is mentioned, but 195 features may already be overwhelming.

The sheet

```
CONTEXTS  SIGILS  ref          ARRAYS         HASHES
void      $scalar SCALAR       @array         %hash
scalar    @array  ARRAY        @array[0, 2]   @hash{'a', 'b'}
list      %hash   HASH         $array[0]      $hash{'a'}
          &sub    CODE
          *glob   GLOB         SCALAR VALUES
                  FORMAT       number, string, ref, glob, undef
REFERENCES
\         reference    $$foo[1]      aka $foo->[1]
$@%&*     dereference  $$foo{bar}    aka $foo->{bar}
[]        anon. arrayref ${$$foo[1]}[2] aka $foo->[1]->[2]
{}        anon. hashref  ${$$foo[1]}[2] aka $foo->[1][2]
\()       list of refs

                        SYNTAX
OPERATOR PRECEDENCE     foreach (LIST) { }     for (a;b;c) { }
->                      while   (e) { }        until (e)   { }
++ --                   if      (e) { } elsif (e) { } else { }
**                      unless  (e) { } elsif (e) { } else { }
! ~ \ u+ u-             given   (e) { when (e) {} default {} }
=~ !~
* / % x                 NUMBERS vs STRINGS  FALSE vs TRUE
+ - .                   =            =      undef, "", 0, "0"
<< >>                   +            .      anything else
named uops              == !=     eq ne
< > <= >= lt gt le ge   < > <= >=  lt gt le ge
== != <=> eq ne cmp ~~  <=>          cmp
&
| ^                     REGEX MODIFIERS     REGEX METACHARS
&&                      /i case insensitive ^     string begin
|| //                   /m line based ^$    $     str end (bfr \n)
.. ...                  /s . includes \n    +     one or more
?:                      /x ignore wh.space  *     zero or more
= += last goto          /p preserve         ?     zero or one
, =>                    /a ASCII    /aa safe {3,7} repeat in range
list ops                /l locale   /d  dual |     alternation
not                     /u Unicode          []    character class
and                     /e evaluate /ee rpts \b    word boundary
or xor                  /g global           \z    string end
                        /o compile pat once ()    capture
DEBUG                                       (?:p)  no capture
-MO=Deparse             REGEX CHARCLASSES   (?#t)  comment
-MO=Terse               .  [^\n]            (?=p)  ZW pos ahead
-D##                    \s  whitespace      (?!p)  ZW neg ahead
-d:Trace                \w  word chars      (?<=p) ZW pos behind \K
                        \d  digits          (?<!p) ZW neg behind
CONFIGURATION           \pP named property  (?>p)  no backtrack
perl -V:ivsize          \h  horiz.wh.space  (?|p|p)branch reset
                        \R  linebreak       (?<n>p)named capture
```

```
                          \S \W \D \H negate    \g{n}  ref to named cap
                                                \K     keep left part
    FUNCTION RETURN LISTS
    stat         localtime      caller        SPECIAL VARIABLES
      0 dev      0 second         0 package   $_    default variable
      1 ino      1 minute         1 filename  $0    program name
      2 mode     2 hour           2 line      $/    input separator
      3 nlink    3 day            3 subroutine $\   output separator
      4 uid      4 month-1        4 hasargs   $|    autoflush
      5 gid      5 year-1900      5 wantarray $!    sys/libcall error
      6 rdev     6 weekday        6 evaltext  $@    eval error
      7 size     7 yearday        7 is_require $$   process ID
      8 atime    8 is_dst         8 hints     $.    line number
      9 mtime                     9 bitmask   @ARGV command line args
     10 ctime                    10 hinthash  @INC  include paths
     11 blksz                     3..10 only  @_    subroutine args
     12 blcks                     with EXPR   %ENV  environment
```

ACKNOWLEDGEMENTS

The first version of this document appeared on Perl Monks, where several people had useful suggestions. Thank you, Perl Monks.

A special thanks to Damian Conway, who didn't only suggest important changes, but also took the time to count the number of listed features and make a Perl 6 version to show that Perl will stay Perl.

AUTHOR

Juerd Waalboer <#####@juerd.nl>, with the help of many Perl Monks.

SEE ALSO

- <http://perlmonks.org/?node_id=216602> – the original PM post

- <http://perlmonks.org/?node_id=238031> – Damian Conway's Perl 6 version

- <http://juerd.nl/site.plp/perlcheat> – home of the Perl Cheat Sheet