

**NAME**

json\_pp – JSON::PP command utility

**SYNOPSIS**

```
json_pp [-v] [-f from_format] [-t to_format] [-json_opt options_to_json1[,option2]]
```

**DESCRIPTION**

json\_pp converts between some input and output formats (one of them is JSON). This program was copied from json\_xs and modified.

The default input format is json and the default output format is json with pretty option.

**OPTIONS****-f**

-f from\_format

Reads a data in the given format from STDIN.

Format types:

json

as JSON

eval

as Perl code

**-t**

Writes a data in the given format to STDOUT.

null

no action.

json

as JSON

dumper

as Data::Dumper

**-json\_opt**

options to JSON::PP

Acceptable options are:

```
ascii latin1 utf8 pretty indent space_before space_after relaxed canonical allow_
allow_singlequote allow_barekey allow_bignum loose escape_slash
```

Multiple options must be separated by commas:

Right: -json\_opt pretty,canonical

Wrong: -json\_opt pretty -json\_opt canonical

**-v**

Verbose option, but currently no action in fact.

**-V**

Prints version and exits.

**EXAMPLES**

```
$ perl -e'print q|{"foo": "", "bar": 1234567890000000000000000000}|' | \
  json_pp -f json -t dumper -json_opt pretty,utf8,allow_bignum
```

```
$VAR1 = {
    'bar' => bless( {
        'value' => [
            '0000000',
```

```

                                '0000000',
                                '5678900',
                                '1234'
                                ],
                                'sign' => '+'
                                }, 'Math::BigInt' ),
    'foo' => "\x{3042}\x{3044}"
};

$ perl -e'print q|{"foo":"","bar":12345678900000000000000000}|' |\
  json_pp -f json -t dumper -json_opt pretty

$VAR1 = {
    'bar' => '1234567890000000000000000000',
    'foo' => "\x{e3}\x{81}\x{82}\x{e3}\x{81}\x{84}"
};

```

**SEE ALSO**

JSON::PP, json\_xs

**AUTHOR**

Makamaka Hannyaharamitu, <makamaka[at]cpan.org>

**COPYRIGHT AND LICENSE**

Copyright 2010 by Makamaka Hannyaharamitu

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself.