

## NAME

CORE - Pseudo-namespace for Perl's core routines

## SYNOPSIS

```
BEGIN {
    *CORE::GLOBAL::hex = sub { 1; };
}

print hex("0x50"), "\n";    # prints 1
print CORE::hex("0x50"), "\n"; # prints 80
```

## DESCRIPTION

The `CORE` namespace gives access to the original built-in functions of Perl. There is no `CORE` package, and therefore you do not need to use or require an hypothetical "CORE" module prior to accessing routines in this namespace.

A list of the built-in functions in Perl can be found in *perlfunc*.

## OVERRIDING CORE FUNCTIONS

To override a Perl built-in routine with your own version, you need to import it at compile-time. This can be conveniently achieved with the `subs` pragma. This will affect only the package in which you've imported the said subroutine:

```
use subs 'chdir';
sub chdir { ... }
chdir $somewhere;
```

To override a built-in globally (that is, in all namespaces), you need to import your function into the `CORE::GLOBAL` pseudo-namespace at compile time:

```
BEGIN {
    *CORE::GLOBAL::hex = sub {
        # ... your code here
    };
}
```

The new routine will be called whenever a built-in function is called without a qualifying package:

```
print hex("0x50"), "\n";    # prints 1
```

In both cases, if you want access to the original, unaltered routine, use the `CORE::` prefix:

```
print CORE::hex("0x50"), "\n"; # prints 80
```

## AUTHOR

This documentation provided by Tels <nospam-abuse@bloodgate.com> 2007.

## SEE ALSO

*perlsub*, *perlfunc*.