Testing
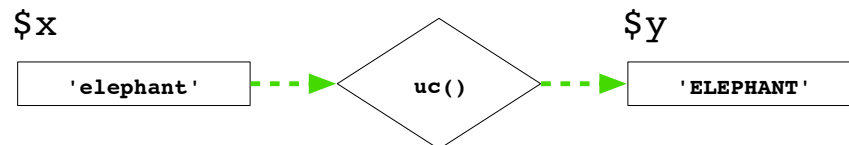
From the Perl documentation

```
uc EXPR
uc      Returns an uppercased version of EXPR.  This is the internal
        function implementing the "\U" escape in double-quoted strings.
        Respects current LC_CTYPE locale if "use locale" in force.  See
        perllocale and perlunicode for more details about locale and
        Unicode support.  It does not attempt to do titlecase mapping
        on initial letters.  See "ucfirst" for that.

        If EXPR is omitted, uses $_.
```
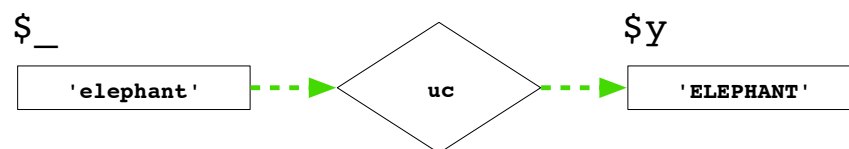
What does it mean?

```
uc EXPR
uc       Returns an uppercased version of EXPR.  [...]
         If EXPR is omitted, uses $_.
```

$x
| 'elephant' | - - - ▶ | uc() | - - - ▶ | 'ELEPHANT' |

$y

$x = 'elephant'; $y = uc($x);

What does it mean?

```
uc EXPR
uc       Returns an uppercased version of EXPR.  [...]
         If EXPR is omitted, uses $_.
```

$_
| 'elephant' | - - - ▶ | uc | - - - ▶ | 'ELEPHANT' |

$y

$_ = 'elephant'; $y = uc;

How can we be sure?

```perl
while (defined(my $line = <STDIN>)) {
    my $LINE = uc($line);
    print "$line --> $LINE -- OK? ";
    my $ok = <STDIN>;
    last unless defined($ok);
    if ($ok =~ /^y(es)?$/i) {
        print "ok - $line --> $LINE\n";
    }
    else {
        print "not ok - $line --> $LINE\n";
    }
}
```

Automated testing

```perl
my %test = (
    'elephant' => 'ELEPHANT',
    'baby'     => 'BABY',
    'Yolanda'  => 'YOLANDA',
    '123'      => '123',
    ''         => '',
);
while (my ($lower, $upper) = each %test) {
    if ( uc($lower) eq $upper ) {
        print "ok - $lower --> $upper\n";
    }
    else {
        print "not ok - $lower --> $upper\n";
    }
}
```

Simpler automated testing

```perl
use Test::More;
plan 'tests' => 5;
my %test = (
    'elephant' => 'ELEPHANT',
    'baby'     => 'BABY',
    'Yolanda'  => 'YOLANDA',
    '123'      => '123',
    ''         => '',
);
while (my ($lower, $upper) = each %test) {
    is( $lower, $upper, "$lower --> $upper" );
}
```

CPAN

Comprehensive Perl Archive Network