


```
sub logger_maker {  
    my $level = shift;  
    my $cb     = shift // sub { say STDERR "[" . shift . "]" ", shift };  
  
    return sub {  
        $cb->($level, $_) for @_;  
    };  
}  
  
my $fatal = logger_maker('fatal', sub { die "[" . shift . "]" ", shift . "\n" });  
$fatal->("Watch this!");  
[fatal] Watch this!  
  
say "Not reached.";
```

sub{die["",shift["","\n"]}]}

shifft

```
$fatal->("Watch this!");
```

```
[fatal] Watch this!
```

Function Composition

```
sub logger_maker {  
    my $level = shift;  
    my $cb     = shift // sub { say STDERR "[" . shift . "]" ", shift };  
  
    return sub {  
        $cb->($level, $_) for @_;  
    };  
}  
  
my $fatal = logger_maker('fatal', sub { die "[" . shift . "]" ", shift . "\n" });  
$fatal->("Watch this!");  
[fatal] Watch this!  
  
say "Not reached.";
```

HTML.pm

```
sub to_html {  
    my %args = @_;  
  
    html_start(%args);  
  
    $args{sth}->execute;  
  
    say "<table>";  
    while (my $row = $args{sth}->fetchrow_hashref) {  
        $args{row_builder}->($row);  
    }  
    say "</table>";  
  
    html_end();  
}
```