Macros

masak

GPW 2012

2012-03-06

What are macros?

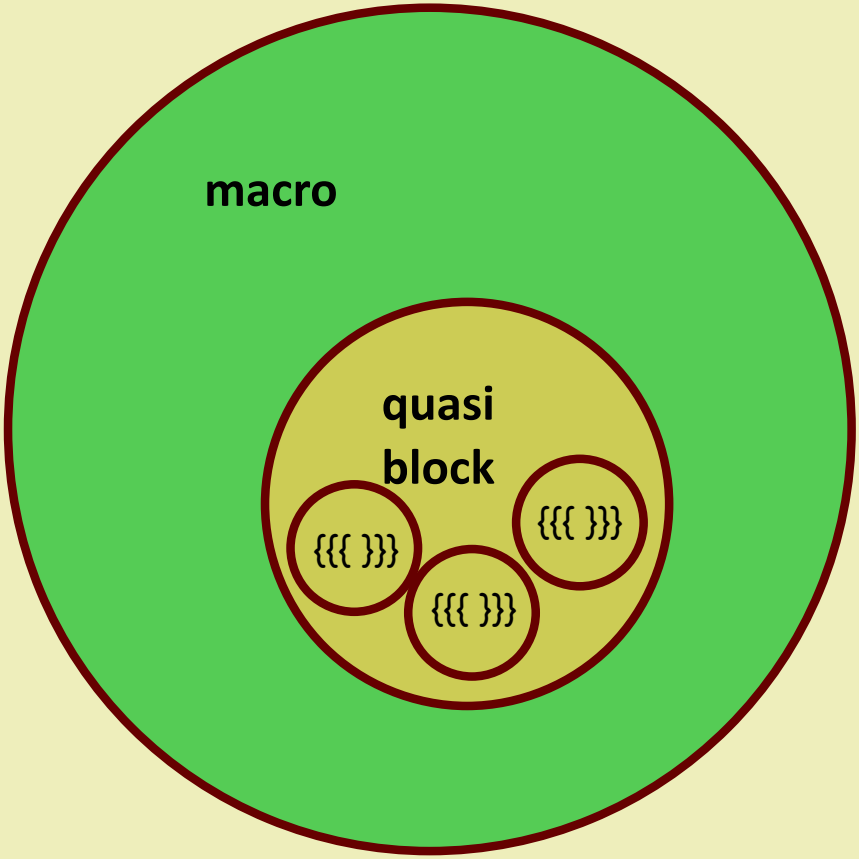
Dive-through example

```
constant LOGGING = False;

sub LOG($message) {
    if LOGGING {
        $*ERR.say: $message;
    }
}

LOG "The value is: {hard-to-compute()}";
```

```
constant LOGGING = False;

macro LOG($message) {
    if LOGGING {
        quasi {
            $*ERR.say: {{{$message}}};
        }
    }
}

LOG "The value is: {hard-to-compute()}";
```

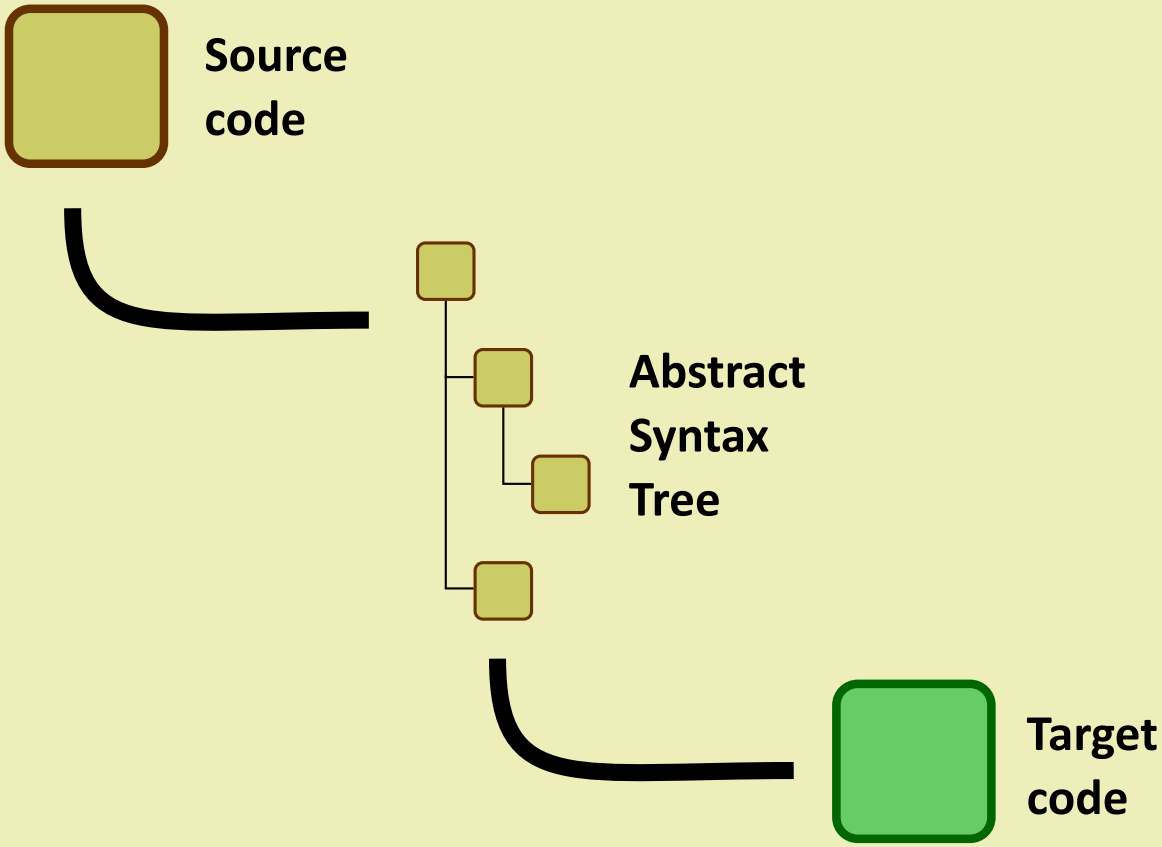
hi, I'm masak

What are macros?


macro
quasi
block
{{{ }}}
{{{ }}}
{{{ }}}

Subroutines map values

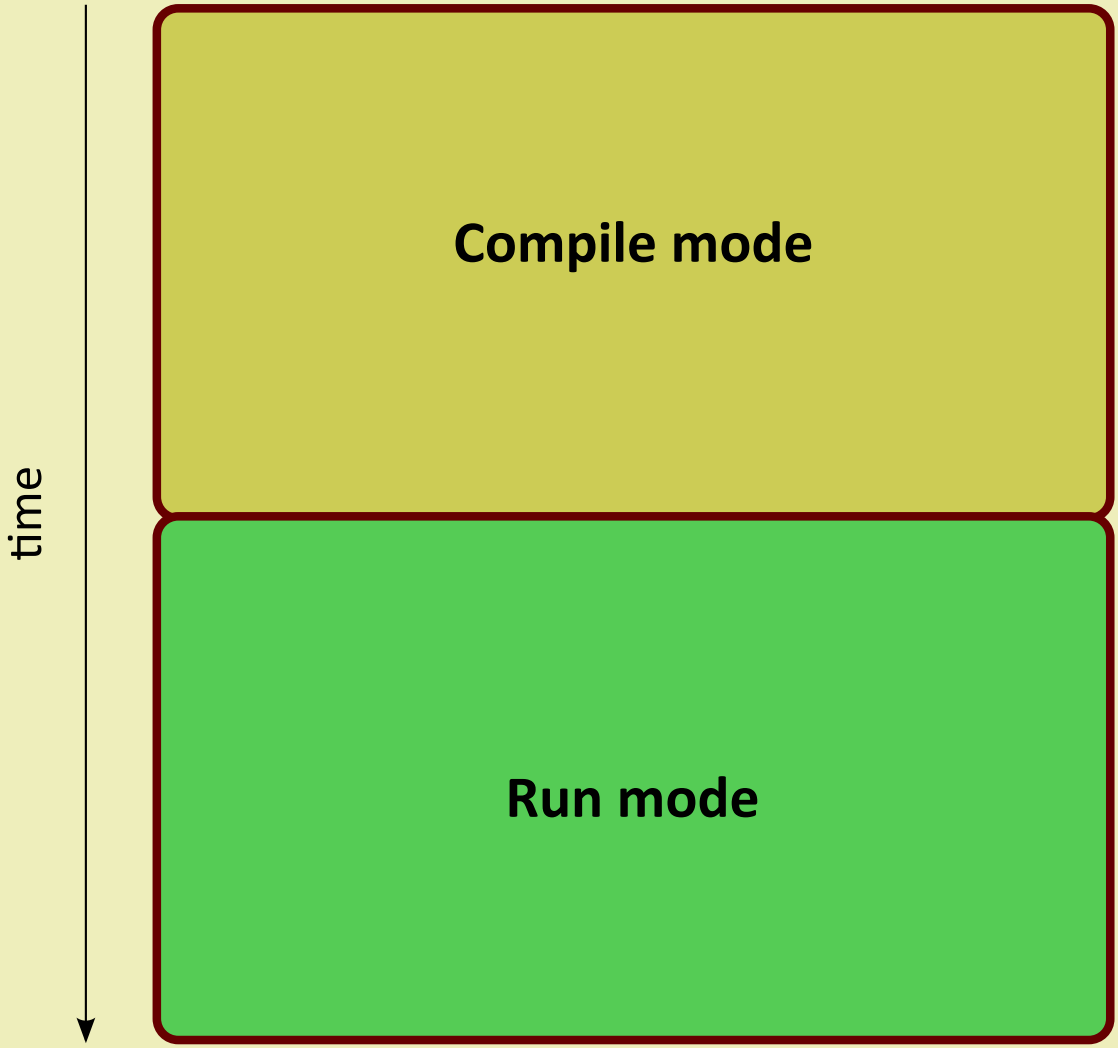
Macros map code

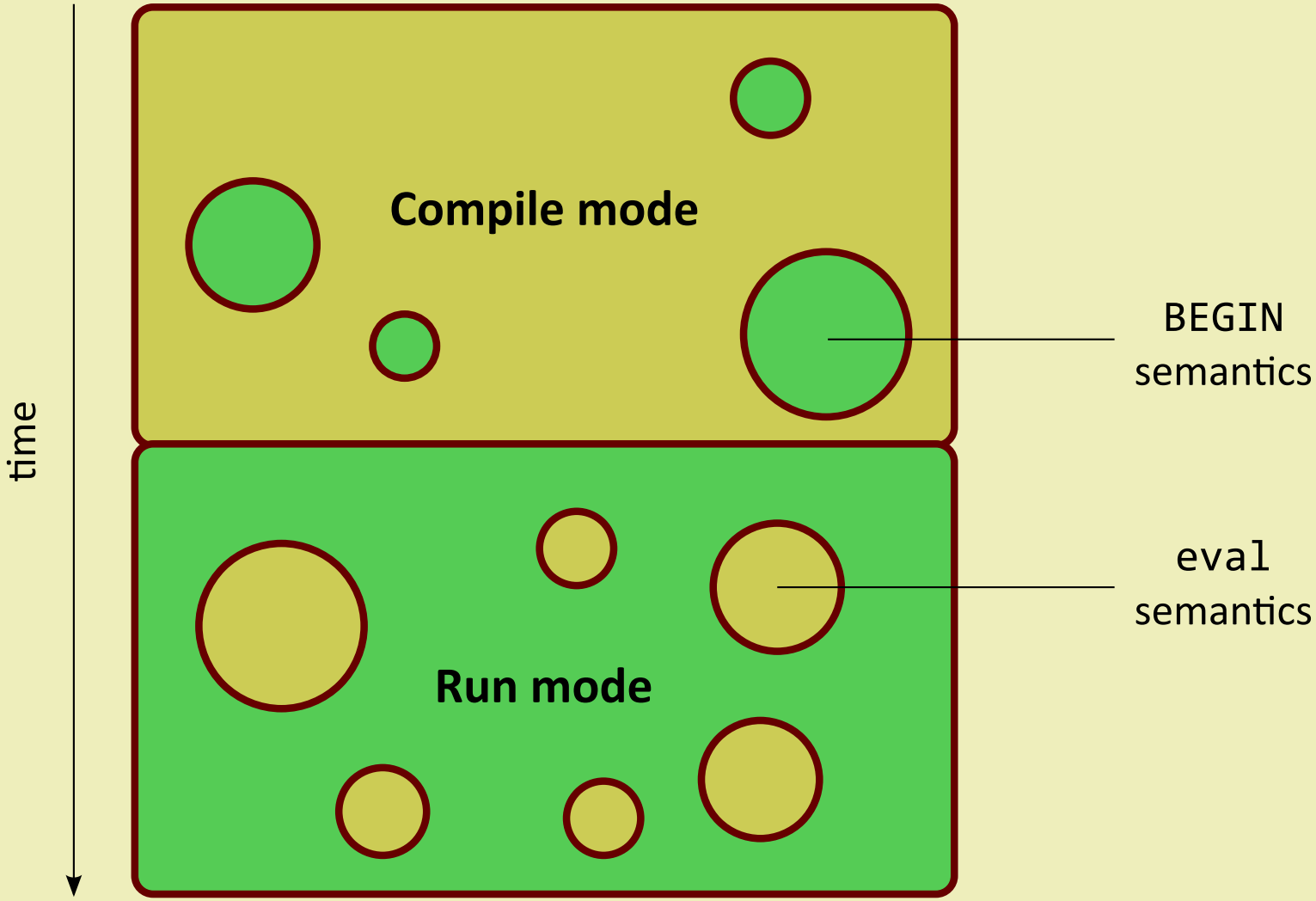

Source
code
Abstract
Syntax
Tree
Target
code

"Meta" code *to talk about **code***

Normal code *to talk about **values***

Perl 6: modable


time
Compile mode
Run mode


time
Compile mode
Run mode
BEGIN
semantics
eval
semantics

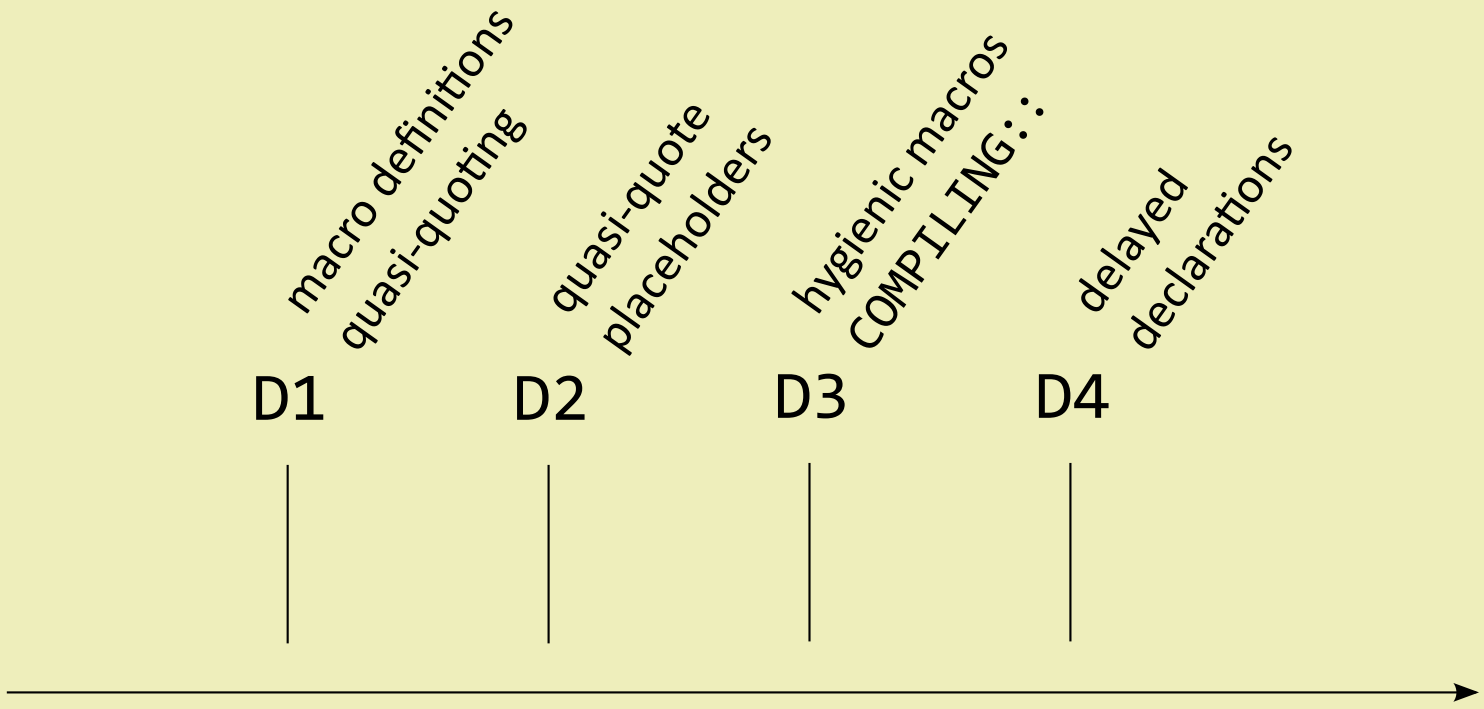
```
BEGIN { say "OH HAI" }
my $value = eval $code;

constant ANSWER = 42;
class Dugong {
    has $.nose = calculate-nose();
}

calculation1() && calculation2();
```

```
Compile time:
BEGIN -- CHECK

Run time:
INIT -- START -- END

Blocks:
ENTER -- LEAVE
PRE -- POST

Loops:
FIRST -- NEXT -- LAST
```

Macros grant

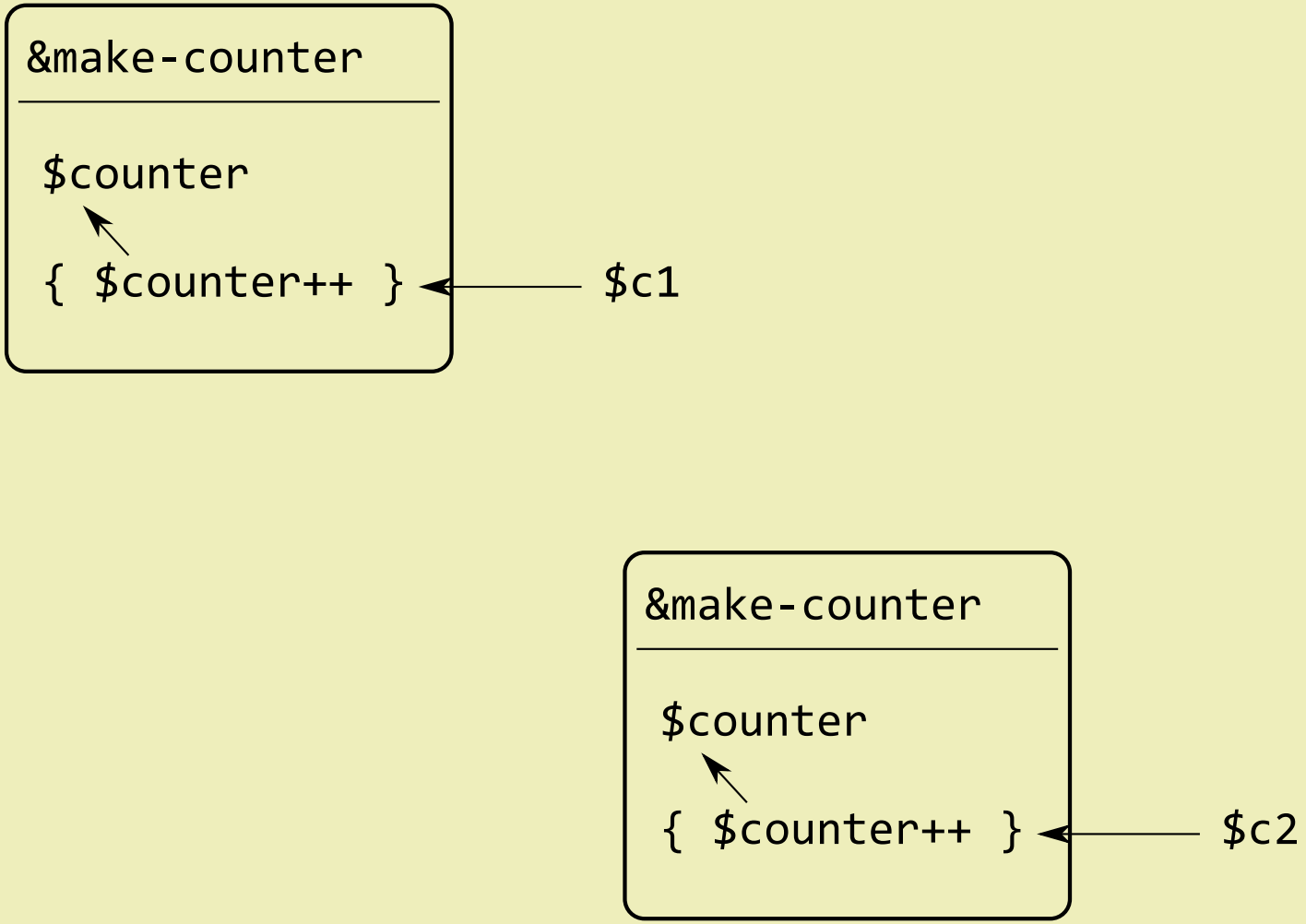
I remember there

being lots of beer


macro definitions
quasi-quoting
quasi-quote
placeholders
hygienic macros
COMPILING::
delayed
declarations
D1
D2
D3
D4

Closures are wonderful

```
sub make-counter($init) {
    my $count = $init;
    return { $count++ };
}

my $c1 = make-counter(5);
my $c2 = make-counter(42);
```


&make-counter
$counter
{ $counter++ }
$c1
&make-counter
$counter
{ $counter++ }
$c2

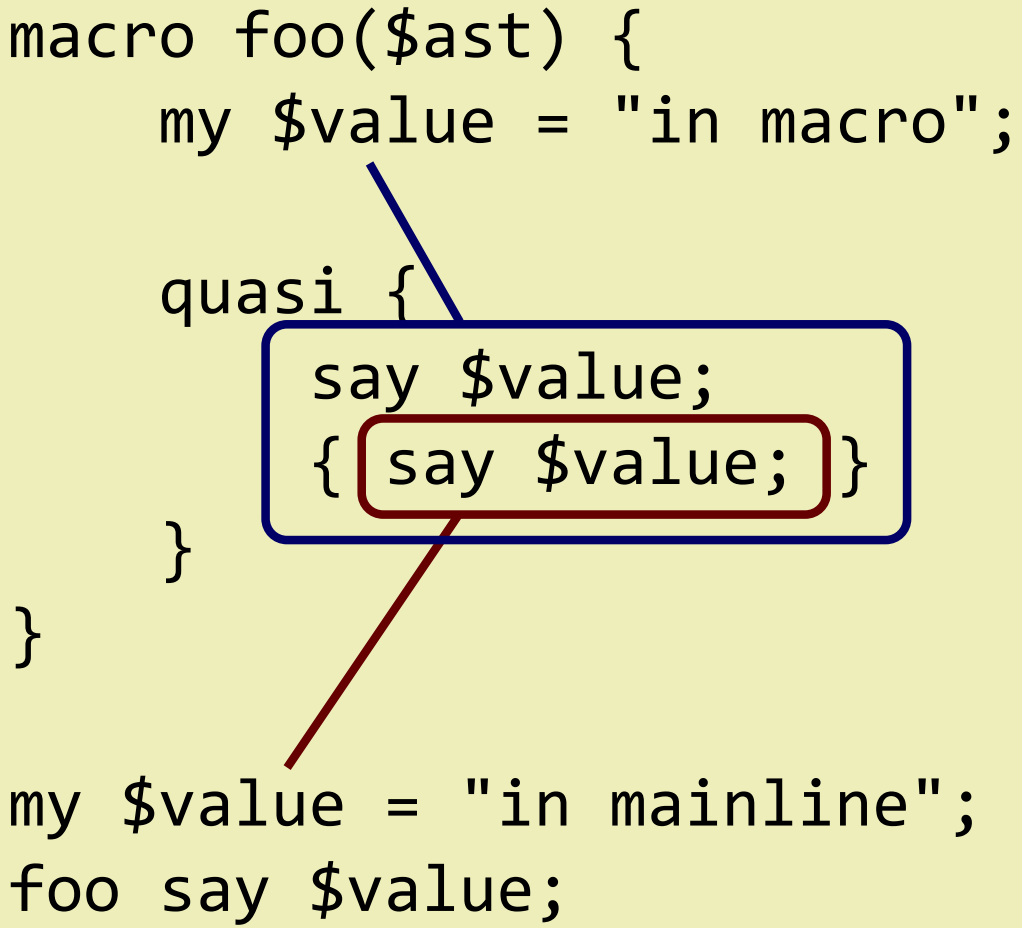
```
class Rectangle {
    has $.width;
    has $.height;

    method area { $.width * $.height }

    method circumference {
        $.width + $.height
        + $.width + $.height
    }
}
```

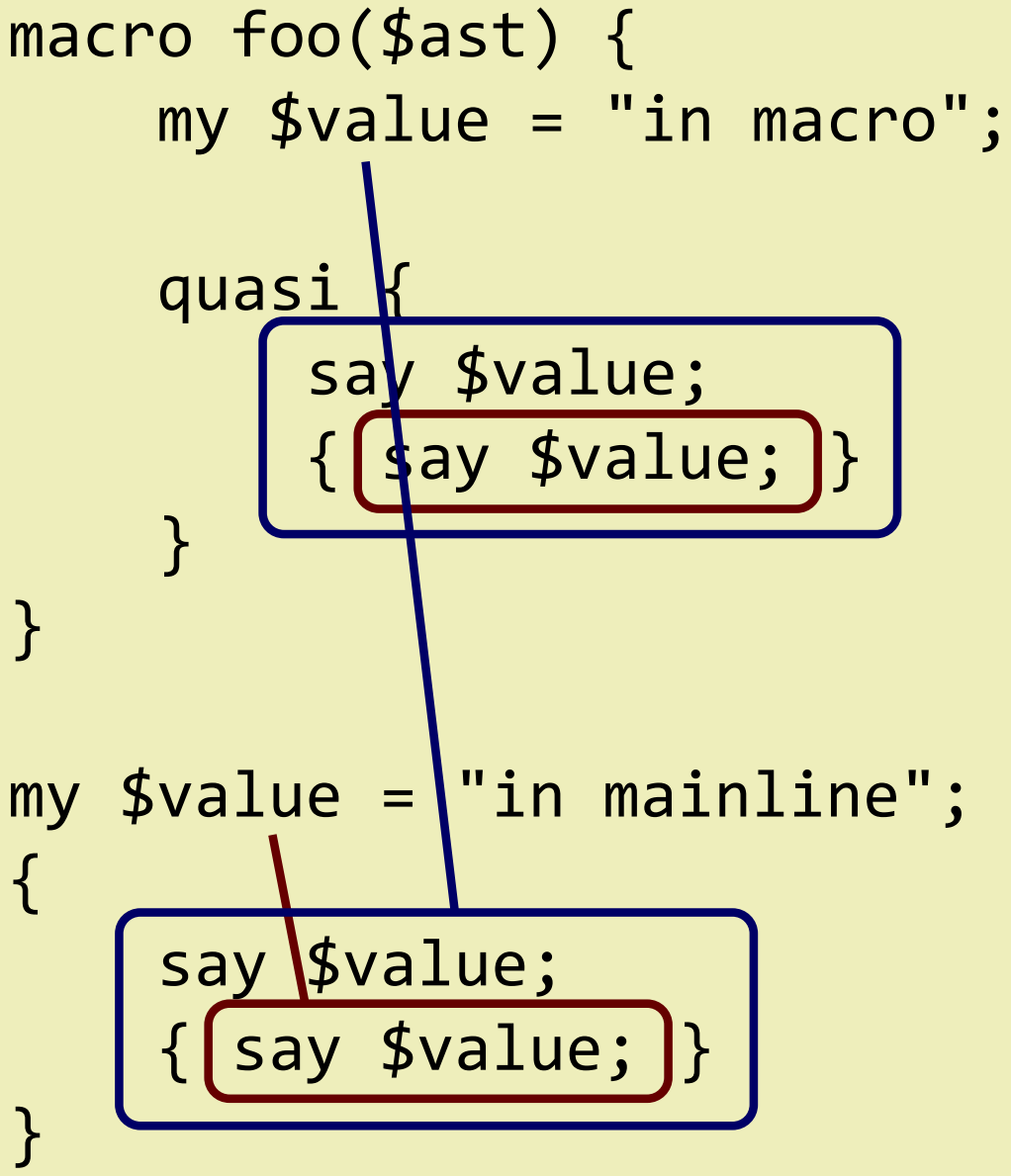
```
sub make-rectangle($width, $height) {
    return sub($_) {
        when 'area' { $width * $height }
        when 'circumference' {
            $width + $height
            + $width + $height
        }
    }
}
```

Macros, bullet time

```
macro foo($ast) {
    my $value = "in macro";

    quasi {
        say $value;
        {{{$ast}}};
    }
}

my $value = "in mainline";
foo say $value;
```

```
macro foo($ast) {
    my $value = "in macro";

    quasi {
        say $value;
        {{{$ast}}};
    }
}

my $value = "in mainline";
foo say $value;
```


macro foo($ast) {
my $value = "in macro";
quasi {
say $value;
{ say $value; }
}
}
my $value = "in mainline";
foo say $value;


macro foo($ast) {
my $value = "in macro";
quasi {
say $value;
{ say $value; }
}
}
my $value = "in mainline";
{
say $value;
{ say $value; }
}

There is no spoon

```
my $x;

role R {
method foo { $x }
}

class C does R {
}
```

“ We need a solution that makes us need less vodka.”

-- jnthn

Conclusion

Macros rock

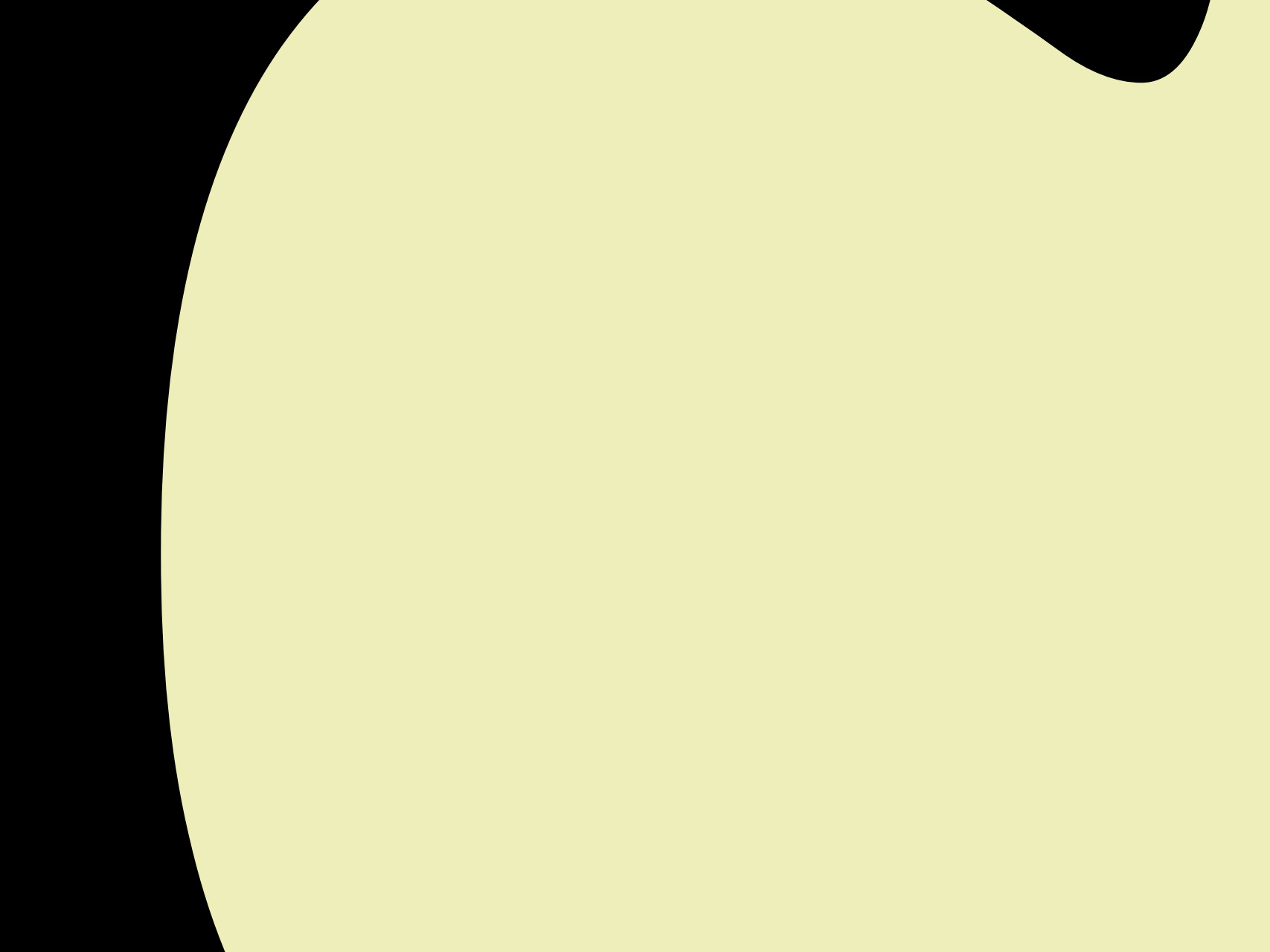
Implementing macros

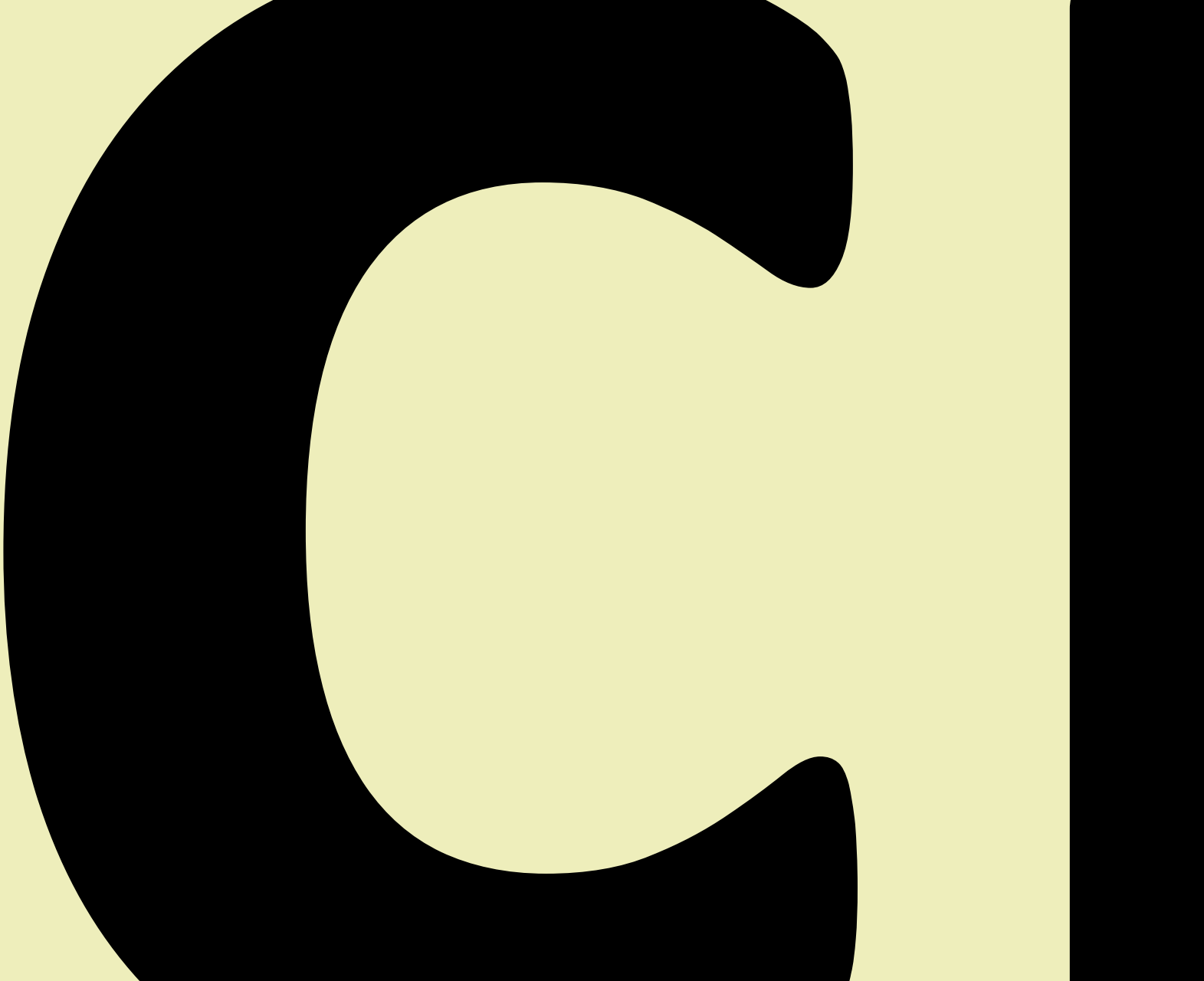
is interesting

Stand by

for macros

in Rakudo





acr

Macro

Macros

masa

GPW 201

2012-03-0

Conclusion

There is no spoon

What are macros?

hi, I'm

cros, bullet time

Macros

What are macro

masak
GPW 2012
2012-03-06

Closures are wonderful

Perl 6: modable

Macros grant

```
my $x;

role R {
method foo ( $x )
}

class C does R {
}
```

is interesting

Macros rock

```
constant LOGGING = False;

sub LOG($message) {
    if LOGGING {
        $*ERR.say: $message;
    }
}

LOG "The value is: {hard-to-compute()}";
```

Dive-through example

Conclusion

There is no spoon

What are macros?

hi, I'm masak

```
macro foo($ast) {
    my $value = "in macro";

    quasi {
        say $value;
        {{{$ast}}};
    }
}

my $value = "in mainline";
foo say $value;
```

```
) {
= "in macro";

alue;
st});

in mainline";
);
```

Macros, bullet time

Macros

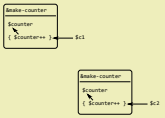
What are macros?

masak
GPW 2012
2012-03-06

Closures are wonderful

Perl 6: modable

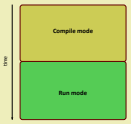
Macros grant

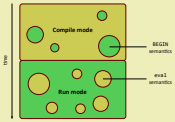



```
sub make-counter($init) {
    my $count = $init;
    return { $count++ };
}

my $c1 = make-counter(5);
my $c2 = make-counter(42);
```

I remember there

being lots of beer





Macros

masak
GPW 2012
2012-03-06

"We need a solution
that makes us
need less vodka."

– jnthn

Stand by

for macros

Implementing macros

in Rakudo

is interesting

Macros rock

```
my $x;

role R {
method foo { $x }
}

class C does R {
}
```

Dive-through example

```
constant LOGGING = False;

sub LOG($message) {
    if LOGGING {
        $*ERR.say: $message;
    }
}
LOG "The value is: {hard-to-compute()}";
```

```
constant LOGGING = False;

macro LOG($message) {
    if LOGGING {
        quasi {
            $*ERR.say: {{{$message}}};
        }
    }
}
LOG "The value is: {hard-to-compute()}";
```

Conclusion

There is no spoon

What are macros?

hi, I'm masak

```
macro foo($ast) {
    my $value = "in macro";

    quasi {
        say $value;
        {{{$ast}}};
    }
}

my $value = "in mainline";
foo say $value;
```

Macros, bullet time

```
macro foo($ast) {
    my $value = "in macro";

    quasi {
        say $value;
        {{{$ast}}};
    }
}

my $value = "in mainline";
foo say $value;
```

```
macro foo($ast) {
    my $value = "in macro";

    quasi {
        say $value;
        {{{say $value;}}}
    }
}

my $value = "in mainline";
foo say $value;
```

```
macro foo($ast) {
    my $value = "in macro";

    quasi {
        say $value;
        {{{say $value;}}}
    }
}

my $value = "in mainline";
{
    say $value;
    {{{say $value;}}}
}
```

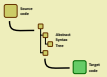
Closures are wonderful

```
sub make-counter($init) {
    my $count = $init;
    return { $count++; };
}

my $c1 = make-counter(5);
my $c2 = make-counter(42);
```

What are macros?

Subroutines map values

Macros map code

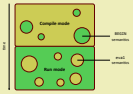


"Meta" code *to talk about **code***

Normal code *to talk about **values***

Perl 6: modable

Macros grant




```
BEGIN { say "OH HAI" }
my $value = eval $code;

constant ANSWER = 42;
class Dogong {
    has $.nose = calculate-nose();
}

calculation() && calculation2();
```

```
Compile time:
BEGIN -- CHECK

Run time:
INIT -- START -- END

Blocks:
ENTER -- LEAVE
PRE -- POST

Loops:
FIRST -- NEXT -- LAST
```

I remember there

being lots of beer

Macros

masak
GPW 2012
2012-03-06

"We need a solution
that makes us
need less vodka."

– jnthn

Stand by

for macros

Implementing macros

in Rakudo

is interesting

Macros rock

```
my $x;

role R {
method foo { $x }
}

class C does R {
}
```

Dive-through example

```
constant LOGGING = False;

sub LOG($message) {
    if LOGGING {
        $*ERR.say: $message;
    }
}
LOG "The value is: {hard-to-compute()}";
```

```
constant LOGGING = False;

macro LOG($message) {
    if LOGGING {
        quasi {
            $*ERR.say: {{{$message}}};
        }
    }
}
LOG "The value is: {hard-to-compute()}";
```

Conclusion

There is no spoon

What are macros?

hi, I'm masak

```
macro foo($ast) {
    my $value = "in macro";

    quasi {
        say $value;
        {{{$ast}}};
    }
}

my $value = "in mainline";
foo say $value;
```

Macros, bullet time

```
macro foo($ast) {
    my $value = "in macro";

    quasi {
        say $value;
        {{{$ast}}};
    }
}

my $value = "in mainline";
foo say $value;
```

```
macro foo($ast) {
    my $value = "in macro";

    quasi {
        say $value;
        {{{say $value;}}}
    }
}

my $value = "in mainline";
foo say $value;
```

```
macro foo($ast) {
    my $value = "in macro";

    quasi {
        say $value;
        {{{say $value;}}}
    }
}

my $value = "in mainline";
{
    say $value;
    {{{say $value;}}}
}
```

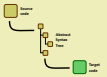
Closures are wonderful

```
sub make-counter($init) {
    my $count = $init;
    return { $count++; };
}

my $c1 = make-counter(5);
my $c2 = make-counter(42);
```

What are macros?

Subroutines map values

Macros map code

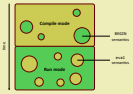


"Meta" code *to talk about **code***

Normal code *to talk about **values***

Perl 6: modable

Macros grant




```
BEGIN { say "OH HAI" }
my $value = eval $code;

constant ANSWER = 42;
class Dogong {
    has $.nose = calculate-nose();
}

calculation() && calculation2();
```

```
Compile time:
BEGIN -- CHECK

Run time:
INIT -- START -- END

Blocks:
ENTER -- LEAVE
PRE -- POST

Loops:
FIRST -- NEXT -- LAST
```

I remember there

being lots of beer