

Parrot Magic

Building Dynamic Languages With Parrot

Stephen Weeks

Penguicon, 2008



Outline

- 1 Parrot
- 2 Parts of a language
- 3 GIL
- 4 Summary



What is Parrot?

- Virtual machine for dynamic languages.
- Compiler tools to build compilers for dynamic languages.



Languages running on Parrot.

- abc
- APL
- befunge
- bf
- c99
- ruby (cardinal)
- scheme (eclectus)
- scheme (pheme)
- lisp
- ecmascript
- hq9+
- lolcode
- lua
- NQP
- Perl 6
- Perl 1
- PHP (plumhead)
- python (pynie)
- Tcl
- WMLScript



Parts of a language

- Parsing
- AST Construction
- Builtins



Parsing

```
token ws {  
    <!ww> [ '#' \N* \n? | \s+ ]*  
}  
rule statement {  
    'say' <value> [ ',' <value> ]* ';' ;  
    {*}  
}  
token value {  
    | <integer> {*}           #= integer  
    | <quote> {*}           #= quote  
}  
token integer { \d+ {*} }  
token quote {  
    [ \' <string_literal: '> \' | \" <string_literal: "> \"]  
    {*}  
}
```



Parsing

```
method statement($/) {
    my $past := PAST::Op.new( :name('say'), :pasttype('call'),
                              :node( $/ ) );

    for $<value> {
        $past.push( $( $_ ) );
    }
    make $past;
}
method value($/, $key) {
    make $( $/{ $key } );
}
method integer($/) {
    make PAST::Val.new( :value( ~$/ ), :returns('Integer'), :node($/) )
}
method quote($/) {
    make PAST::Val.new( :value( $( $<string_literal> ), :node($/) );
}
}
```



Make Titles Informative. Use Uppercase Letters. Subtitles are optional.

- Use `itemize` a lot.
- Use very short sentences or short phrases.

