PHP

Matt Might

University of Utah

matt.might.net

Extra credit

- root `winterfell`: +15% to final grade
- One bonus, globally, per attack vector

Today

Rasmus Lerdorf

“Designer” of PHP

"I actually hate programming."

Rasmus Lerdorf

"PHP is about as exciting as your toothbrush."

Rasmus Lerdorf

"I was really, really bad at writing parsers.
I still am really bad at writing parsers."

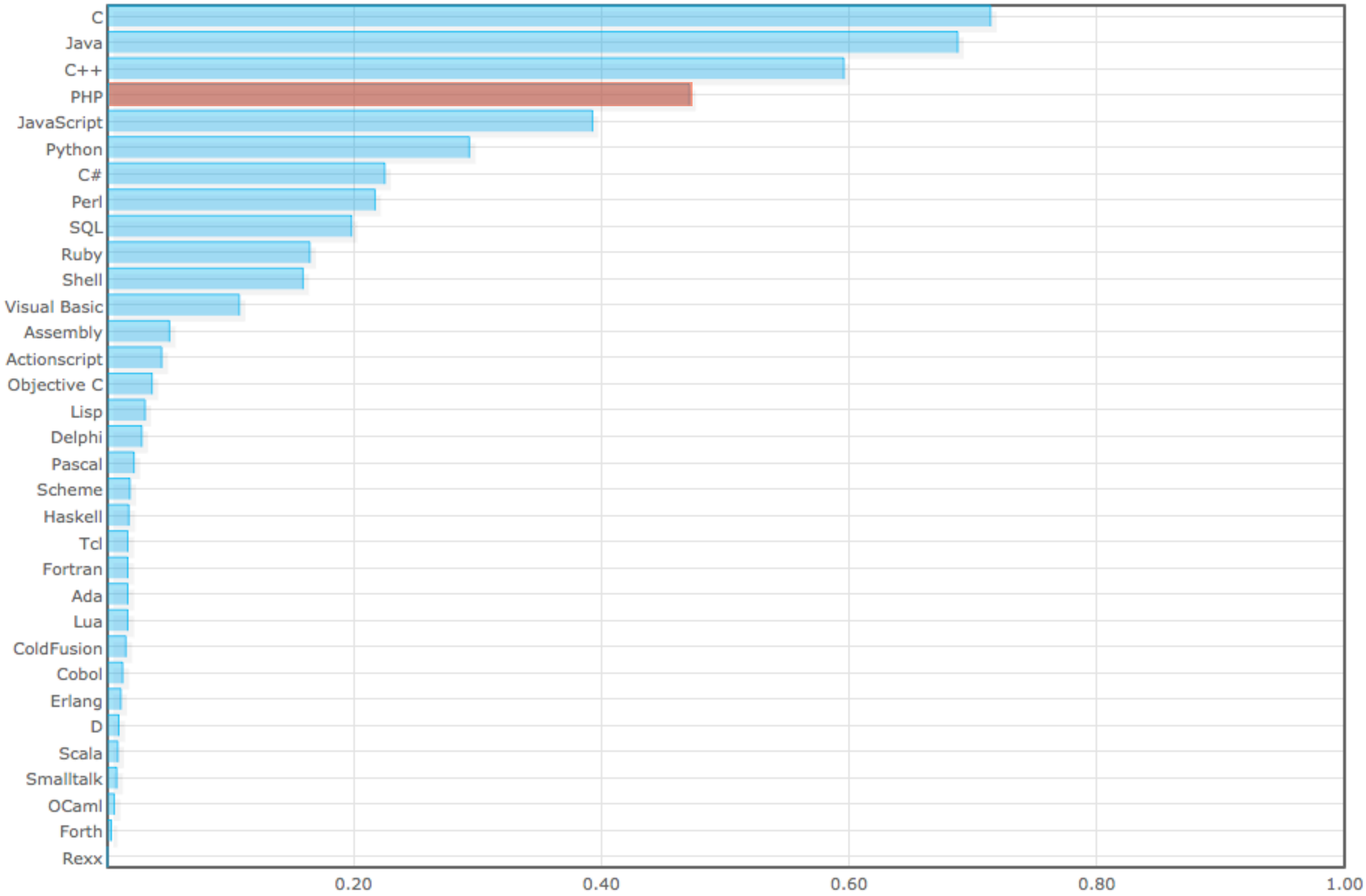
Rasmus Lerdorf

"I don't know how to stop it, there was never any intent to write a programming language [...] I have absolutely no idea how to write a programming language, I just kept adding the next logical step on the way."

Rasmus Lerdorf

"I'm not a real programmer. I throw together things until it works then I move on. The real programmers will say 'Yeah it works but you're leaking memory everywhere. Perhaps we should fix that.' I'll just restart Apache every 10 requests."

Rasmus Lerdorf


C
Java
C++
PHP
JavaScript
Python
C#
Perl
SQL
Ruby
Shell
Visual Basic
Assembly
Actionscript
Objective C
Lisp
Delphi
Pascal
Scheme
Haskell
Tcl
Fortran
Ada
Lua
ColdFusion
Cobol
Erlang
D
Scala
Smalltalk
OCaml
Forth
Rexx
0.20
0.40
0.60
0.80
1.00

Advantages

- Enormous library
- Fast at the frequent
- Low startup cost

<confession>

</confession>

Flaws

- Insecurity
- Inconsistency
- Inscrutability
- Unscalability

Syntax

html

<?php *code* ?>

html

```
<!DOCTYPE html>
<html>
 <body>
  <?php echo “Hello, world!” ?>
 </body>
</html>
```

```
html
<? code ?>
html
```

```
html
<?= expr ?>
html
```

```
html
<% code %>
html
```

```
html
<%= expr %>
html
```

html

<script language="**php**">*code*<script>

html

Comments

```
# comment

// comment

/* comment */
```

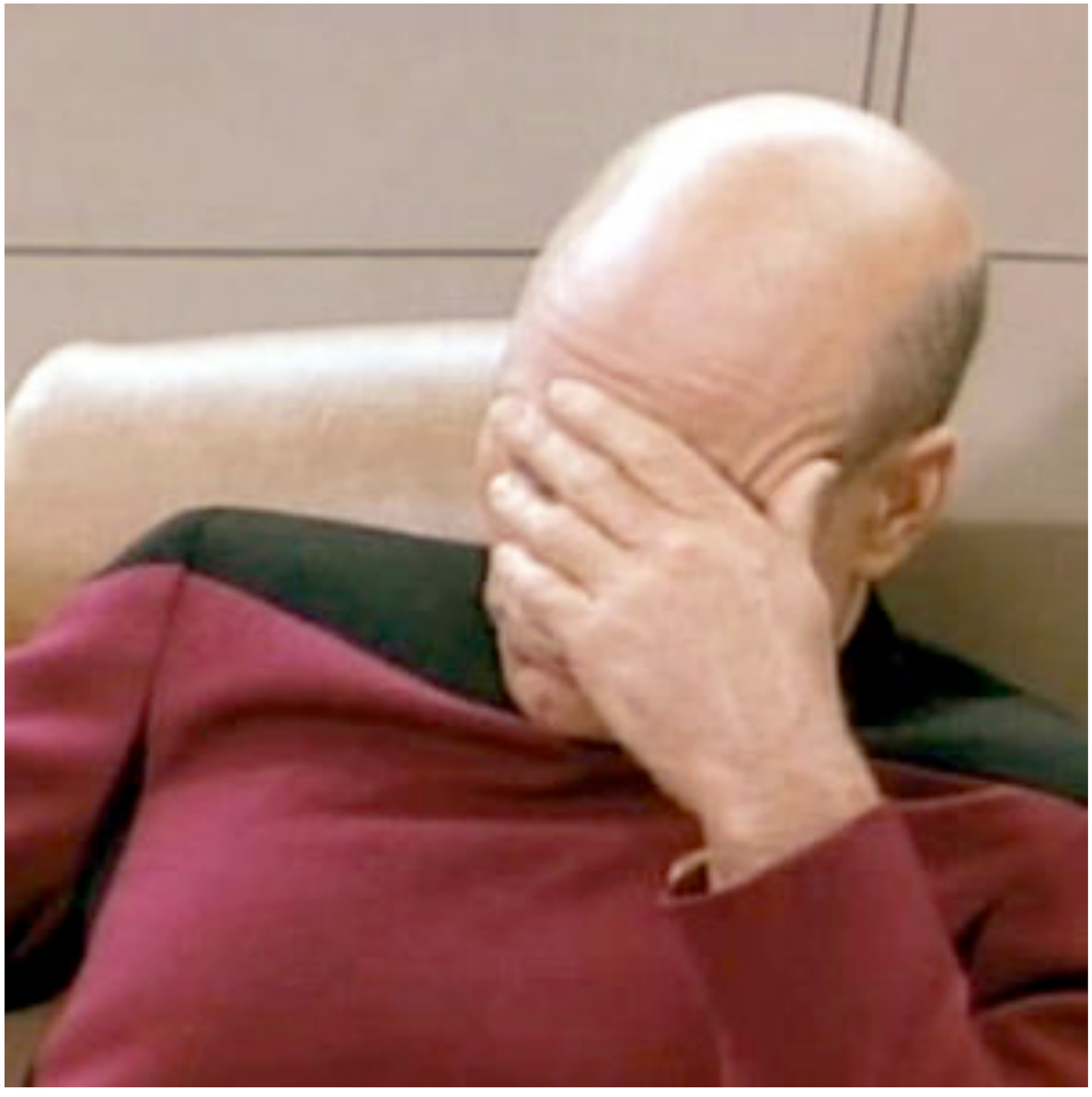
Formal grammar for PHP?

Variables

$name

$$\$name = exp$$

Scope



```
$x = 10 ;

function foo() {
  echo $x ;
}
```

```
function name() {
  global $gvar ;
  ... $gvar ...
}
```

```
function name() {
  ... $GLOBALS['gvar'] ...
}
```

```
global $GLOBALS ; // ????
```

Superglobals

$_GET, $_POST

“Constants”

foo

"foo"

```
define("foo",42)
```

foo

42

References

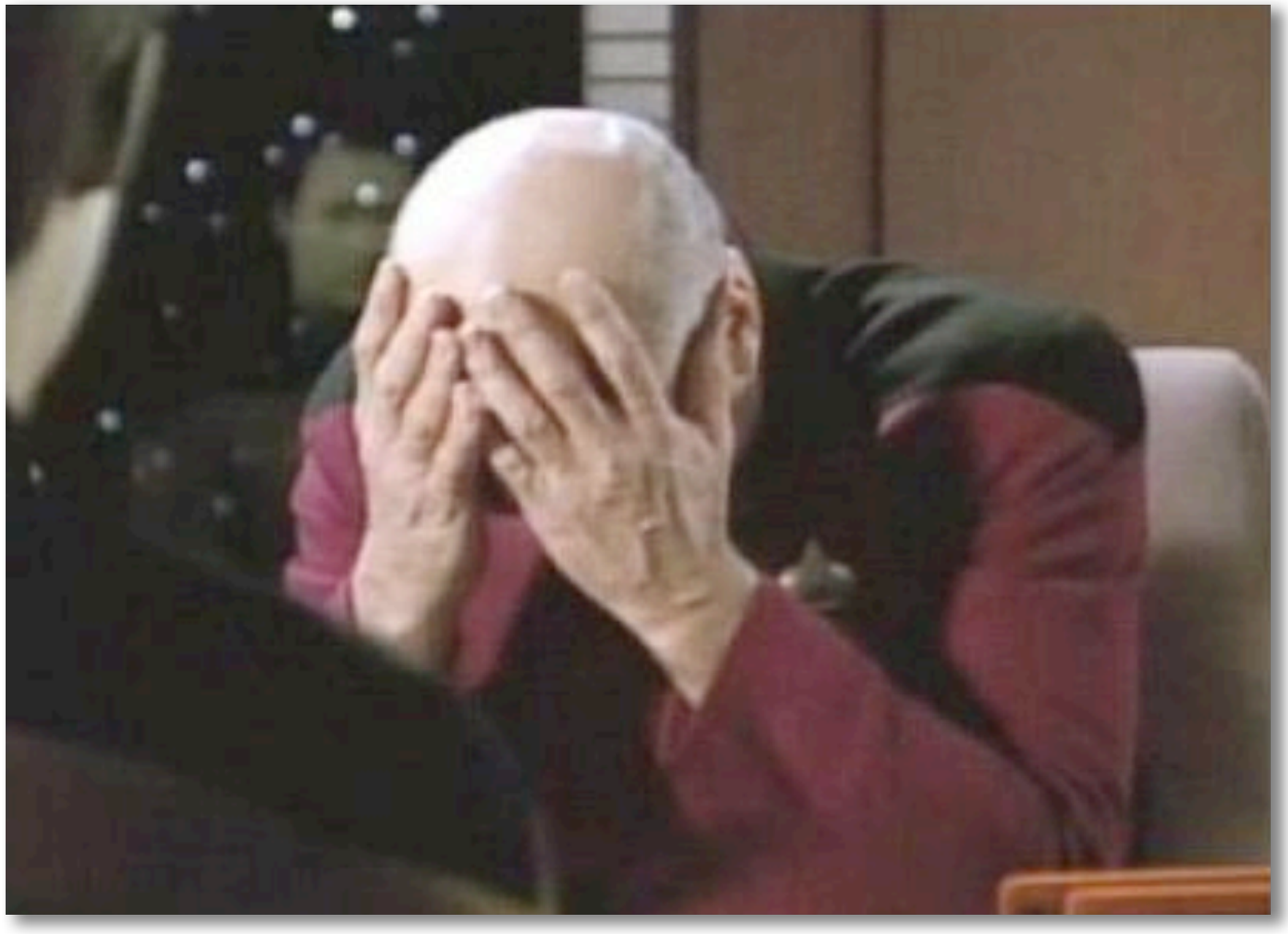
```
$name =& $other
```

```
$foo =  3 ;

$bar =& $foo ;

$bar = 20 ;

echo $foo ;
```

```
$foo = 3 ;

function f($baz) {
  $baz = 30 ;
}

f($foo) ;

echo $foo ;
```

```
$foo = 3 ;

function f(&$baz) {
  $baz = 30 ;
}

f($foo) ;

echo $foo ;
```

```
$a = array('foo') ;

function g($b) {
  $b[0] = 'notfoo' ;
}

g($a) ;

echo $a[0] ;
```



foo

```
$a = array(1, 2, 3) ;

$b = $a ;

$b[0] = 20 ;

echo $a[0] ;
```

1

```
$a = array('foo') ;

function g(&$b) {
  $b[0] = 'notfoo' ;
}

g($a) ;

echo $a[0] ;
```

notfoo

```
$foo = 'foo' ;

$a = array(&$foo) ;

function g($b) {
  $b[0] = 'notfoo' ;
}

g($a) ;

echo $a[0] ;
```

notfoo

```
class C {
  public $foo ;
}

$a = new C;
$a->foo = 10 ;
```

```
function o($b) {
  $b->foo = 20 ;
}

o($a) ;

echo $a->foo ;
```

Arrays

```
array(1, 2, 3)
```

```
[1, 2, 3]
```


```
[1, 2, 3]
```

```
$aa = array( "foo" => 1, "bar" => 10 ) ;
```

```
echo $aa["foo"] ;
```

1

```
$aa = array() ;

$aa["0"] = 42 ;

echo $aa[0] ;
```

```
$aa = array() ;

$aa["00"] = 42 ;

echo $aa[0] ;
```

```
function a () {
 return array(1,2,3);
}

echo a()[0] ;
```

```
$foo = array( "foo" => "bar",
              "baz" => "qux" ) ;
foreach ($foo as $key => $value) {
  echo "$key: $value\n" ;
}
```

```
$foo = array( "foo" => "bar",
              "baz" => "qux" ) ;

$foo[] = 20 ;
```

```
foreach ($foo as $key => $value) {
  echo "$key: $value\n" ;
}
```

Functions

```
function name (formals) {
   body
}
```

```
function (formals) {
    body
}
```

```
function f() {
   $a = 30 ;
   $g = function () {
     return $a ;
   } ;
   echo $g() ;
}

f() ;
```

```
function f() {
   $a = 30 ;
   $g = function () use ($a) {
     return $a ;
   } ;
   echo $g() ;
}

f() ;
```

Equality

Equaliciousness

problem, boole?

Loose comparisons with ==

	TRUE	FALSE	*1*	*0*	*-1*	*"1"*	*"0"*	*"-1"*	NULL	*array()*	*"php"*	*""*
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE
FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE
1	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
0	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
-1	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
"1"	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
"0"	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
"-1"	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
NULL	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
array()	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
"php"	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
""	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE

Classes

```
class name {
  modifier $name ;
  modifier $name = value ;
  modifier function name () {
     body
  }
}
```

```
class name extends super {
  # parent::method
}
```

new *ClassName*

```
public function __construct()
```

```
new ClassName(args)
```

Namespaces

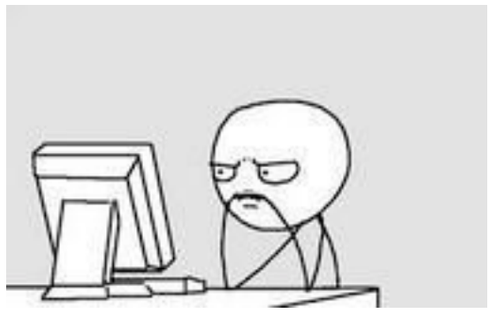
mysql_real_escape_string

$$parent\backslash child\backslash id$$

Web scripting

Model

Client **Server**



GET /foo.php

foo.php

response

<form>

/path?id=value&...

env.smash

stdin.smash

print_GET.php

print_GET.php

print_POST.php

</form>

Questions?