

IT350 Web and Internet Programming

Fall 2005

SlideSet #16: Perl Functions and More

(see online references)

Perl Function Calls

```
use CGI qw( :standard );
print( header() );

# Prints "hello", takes no arguments
sub hello {
    print "\n<br/> Hello.";
}

# Takes two arguments, return their product
sub multiply {
    my($valA, $valB) = @_;
    return $valA * $valB;
}

$x = 2;
hello;
print "\n<br/> $x * 7 = " . multiply($x,7);
hello();
hello(72145);

print(end_html());
```

Function Calls and Arrays

```
# Takes an array as argument, returns minimum value
sub findMin {
    my(@array) = @_;
    my $min = $array[0];
    my $ii;
    my $len = @array;
    for ($ii=0; $ii < $len; $ii++) {
        if ($array[$ii] < $min) {
            $min = $array[$ii];
        }
    }
    return $min;
}

# Defines new global array, @array1
# And returns a new array with 4 elements.
sub makeArray() {
    @array1 = (89, 23, 90);
    my @array2 = (34, 5.4, 123, 2.01);
    return @array2;
}

@test1 = makeArray();
@test2 = (89, 23, 40, -17);
print "\nMin1 is: " . findMin(@test1);
print "\nMin2 is: " . findMin(@test2);
print "\nMin3 is: " . findMin(@array1);
print "\nMin4 is: " . findMin(@array2);
```

Functions and Arrays Gotchas

```
my(@array) = @_;  
    not the same as  
my(@array) = $_;  
  
my ($valA, $valB) = @_;  
    not the same as  
my $valA, $valB = @_;  
  
References:  
@array = (1, 2, 3);  
$ref_array = \@array;  
@array2 = @$ref_array;  
  
print "\nfrom ref: " . $$ref_array[1];  
print "\nfrom array2: " . $array2[1];
```

Multiple Perl Files:

```
require "question_struct.pl";
```

Be sure not to use same names (e.g., function names) in different files!