

Perl parameter passing

Written by Administrator
Friday, 15 July 2011 13:26 -

Here's an interesting thing I should have learned a long time ago: when you pass an array to a subroutine in Perl, it passes the *elements* of the array, not a *pointer* to the array.

One might assume that this function call would pass the array in as an array... but it actually causes all subsequent function parameters to become part of the array once it is passed in:

```
my @array = (1,2,3);
ArrayFunction(@array, 1, "hello");
sub ArrayFunction() {
    my (@array, $num, $str) = @_;
    print @array;
}
```

This would actually print out

```
1,2,3,1,"hello"
```

... and the variables \$num and \$str would be empty. Not what you're expecting to see! Unless you're expecting it

To properly pass an array into a function and use it as an array, you would need to pass the parameters by reference and call the function like this

```
ArrayFunction(@array, 1, "hello");
```

... and declare the function like this:

```
sub ArrayFunction() {
    my ($array, $num, $str) = @_;
    my @array = @{$array};
    print @array;
} This will print out 1,2,3 the way you'd expect    If you're last function parameter is always an
```

Perl parameter passing

Written by Administrator
Friday, 15 July 2011 13:26 -

array, or your only parameter is an array, you may never notice this behavior.