

# Print and echo

- The most common PHP operation is output

```
print "A lot like printing in C\n";
```

```
print "Backslash for \" character escapes\"\\n";
```

```
print "Line breaks  
don't matter";
```

```
echo "Echo is the same as print";
```

```
echo 'Single "quotes" can be used as well';
```

# Variables

- PHP variables are *weakly typed*
  - You don't declare what type they are, as in C++ or Java
  - There's no compiler step to make use of that information
- Variables are declared on assignment
  - Type is implied by the data type
  - All variable names always start with \$

```
$user_name = "wuvcatslolol132";
```

```
$days = 35;
```

```
$weeks = $days / 7;
```

```
$best_var_ever = TRUE;
```

# Data types

- int, float, boolean, string, array, object, NULL
  - Can test types with `is_type` functions
    - `is_string`, `is_float`, etc.
- Type conversions are mostly automatically done by operators
  - `"5" + "7"` (converts both to int first)
  - `"9" / "2"` (converts answer 4.5 to float)
- Can also do explicit type casting
  - `$number = (float) "56.7";`

# Operators

- Same as C++

+ - \* / % ++ --

= for assignment

== != <= >= for comparison

- Same shortcuts too

+= -= \*= /= %=

- String concatenation operator

```
$text = "I am " . $age . " years old";
```

```
$list .= " and another thing";
```

# Comments

- Can use C++ comments

```
// this is a comment
```

```
/*
```

```
    Multi-line comment
```

```
*/
```

- Also can use unix script-style comments
  - Probably more common

```
# this is a comment
```

# Conditional execution

- Same as C++/Java

```
if( condition )
{
    statements;
} elseif {           # can also be else if
    statements;
} else {
    statements;
}
```

# Iterative execution

- Same as C++/Java

```
for( $i=0; i<5; i++ )  
{  
    print "$i\n";  
}
```

```
while( condition )  
{  
    statements;  
}
```

```
do(  
{  
    statements;  
} while( condition );
```