

1: variables

variables are pretty simple, the only difference between php here and most other languages is that php is not typestrong, meaning a variable can contain any type of data, whereas in languages like C or Java, variables are created for a specific type of data, and can/should only contain such.

Here is an example of setting a variable called 'test' to 'hest':

```
$test = 'hest';
```

2: arrays

Arrays are just a lot of variables together in one piece, and they are typeless in PHP just like normal variables.

You can address/use arrays in different ways in php, here are some examples:

example 1: create an array and add some text lines to it:

```
$arr = array();  
// the [] means that it adds to the next available position.  
$arr[] = 'test';  
$arr[] = 'hest';
```

example 2: set elements by key explicitly:

```
$arr[0] = 'test';  
$arr[1] = 'hest';
```

the above types of arrays are not what's generally called an associative array, however, php does support associative arrays, which are used like this:

```
$arr['test'] = 'hest';
```

3: conditions/logic

conditions in php works pretty much the same way as most other languages, its C style. I will just list some examples:

```
// runs code if $somevar is 1  
if( $somevar == 1 ) { code here; }
```

```
//runs code if $somevar is not 1  
if( $somevar != 1 ) { code here; }
```

```
//runs code if $somevar is 1, and $othervar is not 1  
if( $somevar == 1 && $othervar != 1 ) { code here; }
```

```
// runs code if $somevar is 1, or $othervar is 1  
if( $somevar == 1 || $othervar == 1 ) { code here; }
```

All these things can also be combined further, with standard order of resolution.

4: loops

loops also work in much the same way as other languages, notably C/C++.

while is pretty simple, heres an example:

```
// Should be pretty self explanatory, we start by setting the variable to 10, and then we run the loop
that runs while var is bigger than 0, and then in the loop we decrement var by one.
```

```
$var = 10;
while( $var > 0 ) {
    some code;
    $var--;
}
```

for loops are a bit more complicated, but also pretty easy, consider the same loop as a for:

```
for( $var = 10; $var > 0; $var-- ) {
    code here;
}
```

also pretty self explanatory, and saves a little space.

Foreach can be used to loop across arrays easily, so if you have an array:

```
$arr[] = 'abc'; $arr[] = 'def';
```

you can loop over like this:

```
foreach($arr as $value) { $value would be 'abc' first then 'def' }
```

5: functions

Functions are pretty simple in php, you create them like this:

```
function NameOfFunction() {
    code here;
}
```

then you can just call it like this:

```
NameOfFunction();
```

if you want the function to have parameters, thats easy aswell, just create it like this:

```
function NameOfFunction( $parameter1, $param2 ) { code here; }
```

and call it like:

```
NameOfFunction( 'abc', 'def' );
```

Functions can also return things, like this example:

```
function DescribePerson( $name ) {
    return $name . ' Comes from Denmark';
}
```

```
// $var will contain: 'Kasper Comes from Denmark'
```

```
$var = DescribePerson( 'Kasper' );
```

6: classes

classes are mostly working the same way too lol, its pretty simple really, I will only show simple class usage here, as its the most commonly needed, and very quick to get started with

Heres an example of a simple class that just stores some information and has a function:

```
class Test {
    // we will store the name internally in the class,
    private $name;

    // this is the constructor, gets run when the instance is
    created
    function Test( $thename ) {
        $this->name = $thename;
    }

    function DoSomething() {
        code here;
    }
}
```

then you can create it like this:

```
$t = new Test('Kasper');
```

7: printing

Some simple and basic functions for printing out stuff is echo, for instance to print a line of text:

```
echo 'testhest';
```

another good printing function that is built into php is for arrays, its called print_r, and you can use it like this:

```
print_r($somearray);
```

8: various minor things

These are just small things which are there in basically any language.

You do string concatenation with the dot character, like this:

```
$var = 'abc'.'def';
```

which produces in \$var: 'abcdef'

you can increment/decrement variables(with numbers) like this:

```
$var++; $var--;
```

9: Idea for a thing to write to get the hang of it.

I would suggest you create a class called Car, and have a variable in it called Name and Color. Then you have a function in that class which returns an associative array with that information.

Then you create three cars, and put them into normal arrays, non associative, then you loop over them, print out the name in the output from the class, and then `print_r` the entire array that came as output of the class.