The advantages of printf become more obvious as the number and complexity of substitutions increase.

```
<?php
foreach ($orderList as $order) {
  printf("Customer %s ordered %d items, with a total cost of $%.2f including $%.2f tax.",
        $order['first_name'].' '.$order['surname'],
        $order['count'],
        $order['cost'] * ($order['tax_rate'] + 1),
        $order['cost'] * $order['tax_rate']
  )
}
?>
```

The format for each conversion specification is: `%[-|+][padding character][-][width][.precision]type`. Only the leading % and the type specification is mandatory, with the output types as:

b	Argument is treated as an integer, and presented as a binary number.
c	Argument is treated as an integer, and presented as the character with that ASCII value.
d	Argument is treated as an integer, and presented as a (signed) decimal number.
e	Argument is treated as scientific notation (e.g. 1.2e+2). The precision specifier stands for the number of digits after the decimal point since PHP 5.2.1. In earlier versions, it was taken as number of significant digits (one less)
E	Like %e but uses uppercase letter (e.g. 1.2E+2).
u	Argument is treated as an integer, and presented as an unsigned decimal number.
f	Argument is treated as a float, and presented as a floating-point number (locale aware).
F	Argument is treated as a float, and presented as a floating-point number (non-locale aware).
g	Shorter of %e and %f.
G	Shorter of %E and %f.
o	Argument is treated as an integer, and presented as an octal number.
s	Argument is treated as and presented as a string.
x	Argument is treated as an integer and presented as a hexadecimal number (with lowercase letters).

X	Argument is treated as an integer and presented as a hexadecimal number (with uppercase letters).

Above modified from PHP manual list of specifiers

The optional elements are:

1. sign specifier (- or +)
2. padding specifier to the right size string. The default is space.
3. alignment specifier. - will make it left justified, with a default of right justified.
4. width specifier. Minimum number of characters.
5. precision specifier. Period (.) followed by a number that specifies how many digits to be displayed for floating point numbers, or a maximum character limit when applied to a string.

To display an actual % sign, use %% in the format string.

Argument Ordering and reuse

Placeholder substitution need not be done in parameter order, and supplied parameters can be substituted more than once, as shown below:

```
<?php
  printf('%2$s played against %1$s and the winner was %1$s', 'Spain', 'Netherlands');
?>
```

```
<?php
  printf('%1$s as a string, %1$.2f as a floating point, %1$d as a decimal number, %1$o as octal
?>
```