# Human Readable Numbers

By Cal Henderson

Ever wanted to display numbers in a human readable format? Then just use this simple routine. It uses simple recursion and a 'stepping' process to gradually work it's way through the supplied number.

```
function readable_number($a){

$bits_a = array("thousand", "million", "billion", "trillion",
 "zillion");
$bits_b = array("ten", "twenty", "thirty", "fourty", "fifty",
 "sixty", "seventy", "eighty", "ninety");
$bits_c = array("one", "two", "three", "four", "five", "six",
 "seven", "eight", "nine", "ten", "eleven", "twelve",
 "thirteen", "fourteen", "fifteen", "sixteen", "seventeen",
 "eighteen", "nineteen");

if ($a==0){return 'zero';}

$out = ($a<0)?'minus ':'';

$a = abs($a);
for($i=count($bits_a); $i>0; $i--){
 $p = pow(1000, $i);
 if ($a > $p){
  $b = floor($a/$p);
  $a -= $p * $b;
  $out .= readable_number($b).' '.$bits_a[$i-1];
  $out .= (($a)?', ':'');
 }
}
if ($a > 100){
 $b = floor($a/100);
 $a -= 100 * $b;
 $out .= readable_number($b).' hundred'.(($a)?' and ':' ');
}
if ($a >= 20){
 $b = floor($a/10);
 $a -= 10 * $b;
 $out .= $bits_b[$b-1].' ';
}
if ($a){
 $out .= $bits_c[$a-1];
}
return $out;
}
```

Here's some sample output:

```
 8 = eight
12 = twelve
-27 = minus twenty seven
186 = one hundred and eighty six
```

```
1,234,567 = one million, two hundred and thirty four thousand, five hundred
 and sixty seven
```

**Note:** American English speakers may find the use of 'and' confusing, as it's used in America as the decimal point delimiter, where as here we're using 'point'. Changing the code to be compatible with Americans is left as a (simple) exercise for the reader.

*(end of article)*