

Slide 3.11: PHP looping

Slide 3.13: PHP looping (cont.)

[Home](#)

## PHP Looping (Cont.)

### The do...while Statement

The do...while statement will execute a block of code at least once—it then will repeat the loop as long as a condition is true.

```
do {
    code to be executed;
} while ( condition );
```

This program breaks a string into an array of words, in which two words are separated by one and only one space, by using the function

[explode\(separator, string\):](#)

- separator specifies where to break the string.
- string—the string to split

```
<html><body>
<?php
    $sep = " ";
    $str = trim( "This is only a test." );
    $word = explode( $sep, $str );
    $i = 0;
    do
        if ( $i == 0 ) echo $word[$i];
        while ( strlen( $word[++$i] ) != 0 );
    ?>
</body></html>
```

Output =

Check

Reset

### The for Statement

```
for ( initialization; condition; increment ) {
    code to be executed; }
```

The for statement is used when you know how many times you want to execute a statement or a list of statements.

- The [sleep\(seconds\)](#) function delays execution of the current script for a

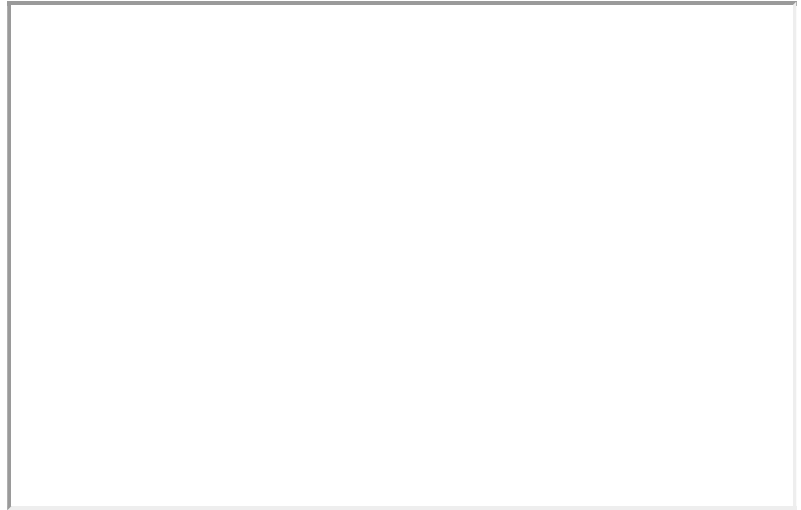
```
<html><body>
<?php
    $number = 10;
    for ( $i=1; $i<=$number; $i++ ) {
        sleep( $i );
        echo date( 'Y-m-d H:i:s' )."<br />";
    }
```

specified number of seconds.

- Use the function [date\(format,timestamp\)](#) for the format, e.g.,

February 08th, 2013, 11:26:05

```
}  
?>  
</body></html>
```



[Slide 3.11: PHP looping](#)

[Slide 3.13: PHP looping \(cont.\)](#)

[Home](#)

