Errors in PHP

**Introduction**

An error is a type of mistake. We can say an error is a condition of having incorrect or false knowledge or an error is defined as an unexpected, invalid program state from which it is impossible to recover.

Error can also be defined as "a deviation from accuracy or correctness". A "mistake" is an error caused by a fault: the fault being misjudgement, carelessness, or forgetfulness. An error message with filename, line number and a message describing the error is sent to the browser.

**Types of error**

Basically there are four types of errors in PHP, which are as follows:

* Parse Error (Syntax Error)
* Fatal Error
* Warning Error – “you are doing something wrong and it is likely to cause errors in the future so fix it”, will not halt execution
* Notice Error – You probably shouldn’t be doing what you are doing but I will let you away with it” will not halt execution

1. **Parse Errors (syntax errors)**

The parse error occurs if there is a syntax mistake in the script; the output is Parse errors. A parse error stops the execution of the script. There are many reasons for the occurrence of parse errors in PHP. The common reasons for parse errors are as follows:

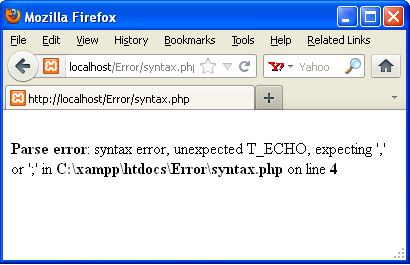
Common reason of syntax errors are:

* Unclosed quotes
* Missing or Extra parentheses
* Unclosed braces
* Missing semicolon

**Example**<?php  
**echo** "Cat";  
**echo** "Dog"  
**echo** "Lion";  
?>

**Output**

In the above code we missed the semicolon in the second line. When that happens there will be a parse or syntax error which stops execution of the script, as in the following image:



**2. Fatal Errors**

Fatal errors are caused when PHP understands what you've written, however what you're asking it to do can't be done. Fatal errors stop the execution of the script. If you are trying to access the undefined functions, then the output is a fatal error.

**Example**

<?php  
**function** fun1()  
{  
**echo** "Vineet Saini";  
}  
fun2();  
**echo** "Fatal Error !!";  
?>

**Output**

In the above code we defined a function fun1 but we call another function fun2 i.e. fun2 is not defined. So a fatal error will be produced that stops the execution of the script. Like as in the following image.



**3. Warning Errors**

Warning errors will not stop execution of the script. The main reason for warning errors are to include a missing file or using the incorrect number of parameters in a function.

**Example**<?php  
**echo** "Warning Error!!";  
**include** ("Welcome.php");  
?>

**Output**

In the above code we include a welcome.php file, however the welcome.php file does not exist in the directory. So there will be a warning error produced but that does not stop the execution of the script i.e. you will see a message Warning Error !!. Like in the following image:



**http://jason.pureconcepts.net/2013/05/fixing-php-errors/**