Types of errors in PHP

- Basically, an error is a mistake in a program that may be caused by writing incorrect syntax or incorrect code.
- An error message is displayed on your browser containing the filename along with location, a message describing the error, and the line number in which error has occurred.

There are usually different types of error. In PHP, mainly four types of errors are considered:

1. Syntax Error or Parse Error
2. Fatal Error
3. Warning Error
4. Notice Error

We will discuss all these errors in detail with examples:

1) Syntax Error or Parse Error

A syntax error is a mistake in the syntax of source code, which can be done by programmers due to their lack of concern or knowledge. It is also known as **Parse error**. Compiler is used to catch the syntax error at compile time.

These errors can occur due to these common reasons like unclosed quotes, missing semicolon, extra or missing parentheses, or unclosed brackets and many more. While compiling the program, syntax error can be caught by the compiler. It gives a parse error or syntax error message.

Example 1: Missing semicolon

```
<?php
   /*------------------syntax error-------------------*/
   echo "Alex: Hie! I'm Alex. </br>";
   echo "Bob: I'm Bob. How are you?"
   echo "Alex: I'm good! and you?";
   echo "Bob: I'm also good";
?>
```

Output

```
Parse error: syntax error, unexpected 'echo' (T_ECHO), expecting ',' or ';' in
C:\xampp\htdocs\program\fatalerror.php on line 5
```

Explanation: In this above example, a semicolon (;) was missing in **line 5**. So, it generated a parse error and displayed an error message on browser as given in the output.

Example 2: Missing dollar symbol

```
<?php
   /*------------------syntax error-------------------*/
   $telecom = "Airtel";
   automobile = "Jaguar";
   echo $telecom;
   echo $automobile;
?>
```

Output

```
Parse error: syntax error, unexpected '=' in
C:\xampp\htdocs\program\fatalerror.php on line 5
```

Explanation: In this above example, dollar ($) symbol was missing in **line 5**. So, it generated a parse error and displayed an error message on browser as given in the output.

2) Fatal Error

A fatal error is another type of error, which is occurred due to the use of undefined function. The PHP compiler understands the PHP code but also recognizes the undefined function. This means that when a function is called without providing its definition, the PHP compiler generates a fatal error.

A fatal error is generated when a function is called without its definition. See the below example containing the fatal error -

Example: Calling undefined function

```
<?php
   /*------------------fatal error-------------------*/
   function add($f1, $f2) {
        $sum = $f1 + $f2;
        echo "Addition:" . $sum;
    }

    $f1 = 23;
    $f2 = 56;

    //call the function that is not defined
    //Generate fatal error
    catch_fatal_error();
    //echo "Fatal Error";
?>
```

- In the above code we have defined the add() function but called other function, which is **catch_fatal_error()**. Therefore, it generates a fatal error and print an error message on the browser as given below:

Output

```
Fatal error: Uncaught Error: Call to undefined function
catch_fatal_error() in C:\xa
```

3) Warning Error

A warning is generated when the programmer tries to include a missing file. The PHP function calls that missing file which does not exist. The warning error does not stop/prevent the execution of the program.

The main reason behind generating a warning error is to pass an incorrect number of parameters to a function or to include a missing file.

Example: Include missing file

```
<?php
    /*-------------------warning error------------------*/
    $cmpny = 'javatpoint';
    echo "Warning Error: ";

    //include a file in the code
    include ('jtp.php');
```

```
?>
```

Output

```
Warning Error:
Warning: include(jtp.php): failed to open stream: No such file or directory in C:\xampp\htdocs\program\fatalerror.php on line 7

Warning: include(): Failed opening 'jtp.php' for inclusion (include_path='C:\xampp\php\PEAR') in C:\xampp\htdocs\program\fatalerror.php on line 7
```

Explanation: In this example, we tried to include a file in our program, which does not exist. So, it generated a warning and displayed an error message.

Notice Error

Notice error is same as warning error. When program contains something wrong, the notice error occurs. But it allows/continue the execution of the program with a notice error. Notice error does not prevent the execution of the code. **For example** - access to undefined variable.

Generally, notice error occurs when we try to use or access a variable which is undefined. See the below example to understand it-

Example 2: Access undefined variable

```
<?php
   /*------------------notice error-------------------*/
   $telecom = "Airtel";
   echo $telecom;
   echo $automobile;
?>
```

Output

```
Airtel
Notice: Undefined variable: automobile in C:\xampp\htdocs\program\fatalerror.php on line 6
```

Explanation: In this above example, we were trying to use a variable **$automobile**, which was not defined. Therefore, it generated a notice "**Undefined variable**" and continued the execution of the program.

How to display all errors in PHP?

A PHP application generates several levels of errors and warnings during the runtime of the script. PHP provides four different ways to display these errors and warnings, which are listed below:

1. **error_reporting:** It displays all level errors except E-NOTICE, E-STRICT, and E_DEPRECATED level errors.
2. **display_errors:** By default, the value of display_errors is **off**. Set it to **on** to display all the errors, including parse (syntax) error.
3. **log_errors:** The default value of log_errors is **ON**, which indicates that the error logging should be done or not.
4. **error_log string:** The error_log string sets that file name where scripts error should be logged.

There are few lines of code given below, add this to your PHP file to display errors. It is a fastest way to display all the PHP errors and warnings.

```
ini_set ('display_errors', 1);
ini_set ('display_startup_errors', 1);
error_reporting (E_ALL);
```

The working of above function and directives is as follow:

ini_set()

This function tries to override the configuration that is found in **php.ini** file.

display_errors

The display_errors is a **directive** that determines whether the error will display to the user or remain hidden. It does not handle the errors that occur during PHP's startup sequence.

display_startup_errors

The display_startup_errors is also a **directive**, which is used to find the error during the startup sequence of PHP.

error_reporting()

The error_reporting is a **native function** of PHP. It is used to display errors.

Program

```
<?php
    ini_set('display_errors', 1);
    ini_set('display_startup_errors', 1);
    error_reporting(E_ALL);
    //include a php file which does not exist
    include("jtp.php");
?>
```

Output

The output will be shown a warning to the browser.

```
Warning: include(jtp.php): failed to open stream: No such file or directory in C:\xampp\htdocs\program\phperror.php on line 6

Warning: include(): Failed opening 'jtp.php' for inclusion (include_path='C:\xampp\php\PEAR') in C:\xampp\htdocs\program\phperror.php on line 6
```

Note: Both these directives display_errors and display_startup_errors would not be able to display the parse errors. Therefore, the PHP.ini configuration must be modified.

Configure PHP.ini to display all errors and warnings

The following changes must be done in **php.ini** file to **display all errors,** including parse error, and restart the **apache server** in **xampp**.

```
display_errors = on
```

Set the **display_errors** directive to "**on"** in PHP.ini file. It will display all the errors, which cannot be displayed by just calling ini_set() function, such as - syntax and parse errors

Program

PHP program when **display_errors** is **disabled** or set to **off** in php.ini file.

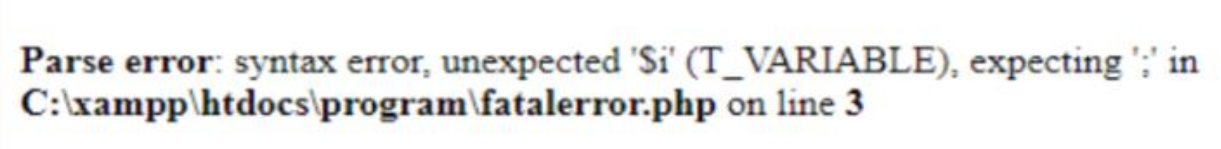
```
<?php
    //semicolon is missing, which generates parse error in php code
    for ($i = 5; $i >= 0 $i--) {
        echo "It will generate parse error";
    }
?>
```

Output

The output will be shown to the browser like the below screenshot when the **display_errors** directive is **disabled**.

Output:

The output for the above program when **display_errors** is **enabled** or set to **on** in php.ini file, and the server is restarted.



```
Parse error: syntax error, unexpected '$i' (T_VARIABLE), expecting ';' in
C:\xampp\htdocs\program\fatalerror.php on line 3
```