

Errors are program events that are handled by the system that cause the program to shut down. In some cases, the system can shut down the program and display an error message. Some errors immediately cause the program to crash (such as the server itself crashing). Errors are usually events beyond the control of the program and not directly caused by code (or lack of code) in the program. For example, insufficient memory will cause application errors.

Exceptions are events that are not part of the normal flow of the program logic. All exceptions should be handled by the program. Exceptions can be “raised” when the application anticipates a problem (a missing file) or when the user does something out of the ordinary (tries to enter invalid information). The program should “catch” all the exceptions. It can then examine the exception and determine if it can be corrected, ignored, or if the application must shut down. If a program does not catch exceptions, the system will display the exception message and then shut down the application.

PHP produces a mixture of errors and exceptions depending on the circumstances. Before PHP 5, exception handling did not exist. Thus, some older PHP commands produced errors (which shut down the program) instead of exceptions. In PHP 7 exception handling is the “rule”. PHP 7 Errors can be handled with exception handling techniques. If exceptions are not handled with program code, the program will halt as if it were a fatal error.

Any time an application is dependent on something external, it is probable that at some point that action will not take place. For example, in the Dog application, the user is expected to enter the proper information. The application must anticipate that not all users will enter correct information. The application is also dependent on several files existing on server (`dog_interface`, `dog_container`, `dog_applications`, and `get_breeds`). If any of these files are missing, the application cannot continue to function properly.

Most object-oriented programming languages use a standard format for handling exceptions. The current version of PHP also uses this approach. As you explore PHP examples on the Internet, you will discover existing PHP code that does not use this standard format. While this code will still execute in the current version of PHP, it is recommend that the standard techniques be used. The standard approach uses the `try-catch` **block**.

```
try {
// code that might cause an exception
}
catch(Exception $e) {
// code that executes if there is an exception
}
catch(Error $e) {
// PHP 7+ capture and handle errors
}
```

Any code that could cause an exception should be included in the `try` block. In addition, you may also want to consider placing other conditions (such as math calculations) in the `try` block.

```
try {
$result = $firstNumber / $secondNumber;
}
catch(Exception $e) {
// code that executes if there is an exception
}
catch(Error $e) {
// PHP 7+ capture and handle errors
}
```