

# Exception Handling



```
if(condition) {
```

**Block will execute if the  
condition is true**

```
} else {
```

**Block will execute if the  
condition is false**

```
}
```

```
try {
```

**Code goes here**

```
} catch( ) {
```

**Executes if any error occurs  
in try block**

```
}
```

# Exception Handling

Exception handling in PHP allows developers to manage errors and unexpected situations that occur during the execution of a PHP script. PHP provides the **try**, **catch**, **finally**, and **throw** keywords for implementing exception handling.

Exception handling in PHP helps in writing robust and error-tolerant code by separating error-handling code from the main program logic.

# Exception Handling

- **With PHP 5 came a new object-oriented way of dealing with errors.**
- **Exception handling is used to change the normal flow of the code execution if a specified error (exceptional) condition occurs. This condition is called an exception.**
- **Exceptions should only be used with error conditions, and should not be used to jump to another place in the code at a specified point.**

# This is what normally happens when an exception is triggered:

- The current code state is saved
- The code execution will switch to a predefined (custom) exception handler function
- Depending on the situation, the handler may then resume the execution from the saved code state, terminate the script execution or continue the script from a different location in the code

# Different error handling methods

- **Basic use of Exceptions**
- **Creating a custom exception handler**
- **Multiple exceptions**
- **Re-throwing an exception**
- **Setting a top level exception handler**