

Помилки в програмі

- 1. Синтаксичні помилки*
- 2. Логічні помилки*
- 3. Помилки часу виконання*
- 4. Обробка помилок*
- 5. Інструкція or die ()*

1. Синтаксичні помилки

Це помилки в імені оператора або функції, відсутність закриває або відкриваючою дужок і т. д., тобто помилки в синтаксисі мови.

Наприклад, якщо замість `echo $ i. "
;` написати `есго $ і. "
;` то Web-браузер відобразить щось подібне `Parse error: syntax error, unexpected T_VARIABLE in C: \ Apache2 \ htdocs \ index.php on line 22`

Часто зустрічаються синтаксичні помилки:

- відсутня крапка з комою в кінці виразу;
- опечатка в імені оператора або функції;
- буква набрана в кирилиці клавіатури замість латинської;
- відсутність відкриває або закриває дужки (або навпаки лишня дужки);
- в циклі `for` вказані параметри через кому, а не через крапку з комою.

2. Логічні помилки

Це помилки в логіці роботи програми, які можна виявити тільки по результатами роботи скрипта.

Наприклад, в логічному виразі замість оператора == (дорівнює) вказано оператор присвоювання =. З точки зору синтаксису тут помилки немає.

3. Помилки часу виконання

Це помилки, які виникають під час роботи скрипта. Причиною є події, не передбачені програмістом. Класичним прикладом служить поділ на нуль.

За допомогою оператора @ можна придушити виведення повідомлень про будь помилку у виразі, якому він передує.

Наприклад, можна придушити висновок про помилку ділення на нуль:

```
$ Val = @ (2/0);
```

```
або @ $ val = 2/0;
```

Однак після цього значення \$ val не матиме сенсу (в даному випадку

\$ Val отримає значення false і логічний тип даних), тобто сама помилка усунена не буде.

4. *Обробка помилок*

Задати ступінь обробки і протоколювання помилок дозволяє директива `error_reporting` у файлі `php.ini`:

```
error_reporting = E_ALL & ~ E_NOTICE
```

Значення директиви:

- E_ALL (всі помилки),
- E_ERROR (фатальні помилки),
- E_RECOVERABLE_ERROR, E_WARNING (попередження часу виконання),
- E_PARSE (синтаксичні помилки),
- E_NOTICE (зауваження, наприклад, про те, що змінна не ініціалізований) та ін

5. Інструкція *or die* ()

Для обробки критичних для всієї програми помилок можна використовувати інструкцію *or die* (). В круглих дужках може бути вказано повідомлення про помилку або функція, яка буде викликана в разі виникнення помилки. Після виведення повідомлення або виклику функції виконання скрипта припиниться:

```
@ $ File = fopen ("file.txt", "r") or die  
  ("Помилка");
```

Або

```
@ $ File = fopen ("file.txt", "r") or die (err_msg  
  ());
```

```
function err_msg () {  
  echo "Помилка";  
}
```