**Ошибки и исключения.**

**Обработка исключений.python**

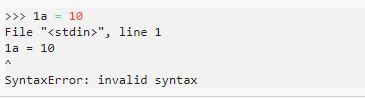
Артур Арламов и Даниил Хелилайд,SPTVR19

**Ошибки и исключения.**

До этого момента сообщения об ошибках лишь упоминались, но если вы пробовали примеры на практике — возможно, вы уже видели некоторые. Существует (как минимум) два различимых вида ошибок: синтаксические ошибки (syntax errors) и исключения (exceptions).

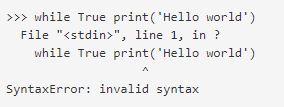
В любой, особенно большой, программе могут возникать ошибки, приводящие к ее неработоспособности или к тому, что программа делает не то, что должна. Причин возникновения ошибок много.

Программист может сделать ошибку в употреблении самого языка программирования. Другими словами, выразиться так, как выражаться не положено. Например, начать имя переменной с цифры или забыть поставить двоеточие в заголовке сложной инструкции. Подобные ошибки называют **синтаксическими**, они нарушают синтаксис и пунктуацию языка. Интерпретатор Питона, встретив ошибочное выражение, не знает как его интерпретировать. Поэтому останавливает выполнение программы и выводит соответствующее сообщение, указав на место возникновения ошибки:

****

**Синтаксические ошибки.**

Синтаксические ошибки, также известные как ошибки разбора кода (парсинга, parsing) — вероятно, наиболее привычный вид жалоб компилятора, попадающихся вам при изучении Python:

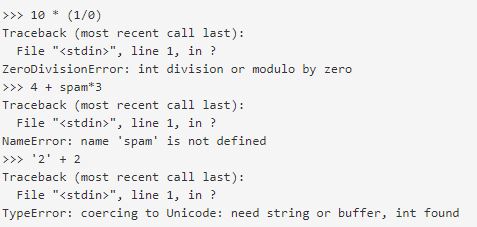


**Исключения и их обработка**

**Исключения.**

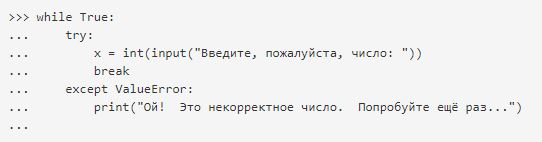
Даже если выражение или оператор синтаксически верны, они могут вызвать ошибку при попытке их исполнения. Ошибки, обнаруженные при исполнении, называются исключениями (exceptions). Они не фатальны: позже надо научиться перехватывать их в программах на Python. Большинство исключений, правда, как правило, не обрабатываются программами и приводят к сообщени об ошибке.

Последняя строка сообщения об ошибке описывает произошедшее. Исключения представлены различными типами и тип исключения выводится в качестве части сообщения: в примере это типы ZeroDivisionError, NameError и TypeError



**Обработка исключений.**

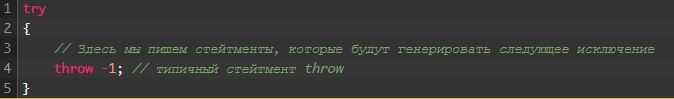
Существует возможность написать код, который будет перехватывать избранные исключения. Посмотрите на представленный пример, в котором пользователю предлагают вводить число до тех пор, пока оно не окажется корректным целым. Тем не менее, пользователь может прервать программу (используя сочетание клавиш Control-C или какое-либо другое, поддерживаемое операционной системой); заметьте — о вызванном пользователем прерывании сигнализирует исключение KeyboardInterrupt.

****

**Операторы исключения.**

**1.Оператор try**

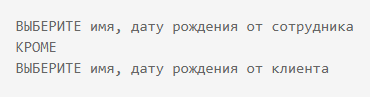
Оператор try (**ключевое слово try)** используеться для определения блока стейтментов (так называемого «**блока try**»). Блок try действует как наблюдатель, в поисках исключений, которые были выброшены каким-либо из операторов в этом же блоке try. Например:



**2.Оператор except**

Оператор EXCEPT используется для исключения одинаковых строк, найденных в одном запросе, но не в другом. Он возвращает строки, уникальные для одного результата. Чтобы использовать оператор EXCEPT, оба запроса должны возвращать одинаковое количество столбцов, и эти столбцы должны иметь совместимые типы данных.

Пример:



Или

