STEP IT ACADEMY

Занятие №13

Обработка исключений. Собственные исключения

Исключения (exceptions) - ещё один тип данных в python.

Самый простейший пример исключения - деление на ноль:

```
>>>
>>> 100 / 0
Traceback (most recent call last):
  File "", line 1, in
    100 / 0
ZeroDivisionError: division by zero
```

При возникновении исключения работа программы прерывается, и чтобы избежать подобного поведения и обрабатывать исключения в Python есть конструкция **try..except**.

```
try:
    инструкции
except [Тип_исключения]:
    инструкции
```

Протестируйте код программы

```
try:
    number = int(input("Введите число: "))
    print("Введенное число:", number)
except:
    print("Преобразование прошло неудачно")
print("Завершение программы")
```

Сверьте результат

Если введем строку

Введите число: hello
Преобразование прошло неудачно
Завершение программы

Если введем число

Введите число: 22
Введенное число: 22
Завершение программы

Блок finally

При обработке исключений также можно использовать необязательный блок **finally**.

Отличительной особенностью этого блока является то, что он выполняется вне зависимости, было ли сгенерировано исключение:

```
try:
    number = int(input("Введите число: "))
    print("Введенное число:", number)
except:
    print("Преобразование прошло неудачно")
finally:
    print("Блок try завершил выполнение")
print("Завершение программы")
```

Протестируйте код программы

```
try:
    n = input('Введите целое число: ')
    n = int(n)
except ValueError:
    print("Неверный ввод")
    3 / 0
except ZeroDivisionError:
    print("Деление на ноль")
else:
    print("Все нормально. Вы ввели число", n)
finally:
    print("Конец программы")
```

raise

```
raise ZeroDivisionError


Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    raise ZeroDivisionError
ZeroDivisionError
```

Разберемся на примере операции деления:

```
a,b=int(input()),int(input())  # вводим 1 затем 0
if b==0:
    raise ZeroDivisionError
Traceback (most recent call last):
  File "<pyshell#2>", line 3, in <module>
    raise ZeroDivisionError
ZeroDivisionError
```

Пояснение

Здесь ввод пользователя в переменные a и b конвертируется в целые числа. Затем проверяется, равна ли b нулю. Если да, то вызывается ZeroDivisionError.

Что выведет это программа?

```
a,b=int(input()),int(input())
try:
    if b==0:
        raise ZeroDivisionError
except:
   print("Деление на 0")
print("Будет ли это напечатано?"
```

Что такое исключения?

Исключения

Исключения — один из двух основных типов ошибок в программировании. В отличие от синтаксических ошибок, которые возникают во время написания, исключения могут появиться во время выполнения программы.

Пример исключения

Примером такого различия может служить автомобиль: он может быть неисправен, что сделает путешествие на нем невозможным (подобно синтаксическим ошибкам), кроме того, водитель может не справиться с управлением, что приведет к ДТП уже во время поездки (подобно исключениям). В программировании же исключения приводят не к ДТП, а к полному прекращению или к неверному выполнению программ. Для избегания прекращения работы или получения дополнительной информации об ошибке используют конструкции обработки исключений.

Типы исключений

У каждого исключения есть собственный тип, который определяется тем, какая ошибка его вызвала, и дает возможность по-разному реагировать на различные виды ошибок.

Типы исключений

BaseException — базовый тип, из которого происходят все остальные, в том числе системные:

Exception — базовый тип для «стандартных» и пользовательских исключений:

ArithmeticError — арифметическая ошибка:

OverflowError — возникает, когда результат арифметической операции слишком велик для представления;

Ошибки при исключениях

- ZeroDivisionError — деление на ноль.
- ImportError — импортировать модуль или его атрибут не удалось;
- LookupError — некорректный индекс или ключ:
- IndexError — индекс не входит в диапазон элементов;
- KeyError — несуществующий ключ (например, в словаре).

Ошибки при исключениях

- NameError — не найдено переменной с указанным именем;
- RuntimeError — возникает, когда исключение не попадает ни под одну из других категорий;
- SyntaxError — синтаксическая ошибка:
- IndentationError — неправильные отступы:
- TabError — смешивание в отступах табуляции и пробелов.
- TypeError — операция применена к объекту несоответствующего типа;
- ValueError — функция получает аргумент правильного типа, но некорректного значения.

Перехват исключений

Для обработки исключений в Python используют конструкцию «try … except». В общем случае для построения этой конструкции необходимо:

- открыть блок «try», введя соответствующую инструкцию и двоеточие;
- указать набор инструкций, в результате работы которых может возникнуть исключение;
- выделить набор инструкций отступом;
- открыть блок «except», введя соответствующую инструкцию и двоеточие;
- указать набор инструкций, которые нужно выполнить в случае возникновения исключения;
- выделить набор инструкций отступом.


STEP
IT ACADEMY