|  |  |  |
| --- | --- | --- |
| **Название этапа** | **Название папки/листинга** | **Комментарии автора** |
| Немного об истории багов | **нет** | Рассказываем про известные баги, и что их можно избежать с помощью exception |
| Исключения | Исключение\_1 | Для управления ошибками, которые возникают в ходе выполнения программ, в Python используются специальные объекты, которые называются исключениями. Если при выполнении кода Python не знает, что ему нужно делать дальше, создается объект исключения. Если в программу был включен код обработки исключения, то выполнение программы продолжится, а если нет – выполнение скрипта остановится. Посмотрим на распространённые исключения |
| Zero-division | zero\_division\_1.py | Рассказываем про исключение zero division |
| Zero-division except | zero\_division\_2.py | Теперь подкладываем «подложку» в виде except |
| division\_3 | division\_3.py | Использование нашего except в реальной ситуации с реальным калькулятором |
| Zero\_division\_except else | division\_4.py | Рассказываем про конструкцию try except  else |
| Обработка Исключений “файл не найден” | except\_file\_5.py | Здесь показываем неуспешную попытку открытия файла, если файла с его именем на самом деле нет |
| Обработка исключения «File not Found» | except\_file\_more\_6.py | Обработка исключений File not found |
| Чтение множества файлов | amount\_of\_files\_2 | Смотрим как можно читать множество файлов и обсуждаем pass |
| pass | Pass/test\_function | Просто пробуем написать пустую функцию без pass и с ней |
| … | Pass/test\_function\_alternate | Рассказываем про альтернативный синтаксис трех точек |
| Пользовательские типы исключений | custom\_exception\_4/custom\_exception | Создаем собственные исключения, чтобы программа могла отклонять к примеру неподходящего |
| assert | Assert-6/index\_assert | Пишем программу, которая запрещает делать отрицательную скидку |
| Работа в OS | | |
| Перебор всех файлов в папке | file\_ary\_6/just\_spisok1.py | Просто перебираем все файлы, которые есть в папке |
| Поиск файла по окончанию | File\_ary\_6/search\_2.py | Здесь мы смотрим на те файлы, которые у нас есть в папке, делаем небольшую фильтрацию по их окончанию |
| Используем Unix Style | File\_ary\_6/ unix\_style\_3 | Используем звездочку и вопросик, как это было принято в Unix |
| Удаление конкретного файла | Delete/delete.py | Здесь мы смотрим на удаление одного конкретного файла, если последний присутствует в папке |
| Удаление целых директорий | del\_dir/delete\_folder.py | Создаем простейшую утилиту,которая удаляет папки |
| Копирование файлов | Backup\_10/backup.py | Создание программы простого бекапа, которая может переносить из одной папки файлы в другую |
| Создание простого блокнота, в котором можно заносить только одно дело | Bloknot\_dlya\_bednih/bloknot.py | Используем модуль pickle, который делает все наши данные максимально неизменяемыми |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |