[00:00.440 --> 00:15.080] Всем привет, добро пожаловать. Этим стулом GAMS начинает новую серию полукресел, а мы этим видео начинаем новую серию видео, она будет называться «Читаем Луца».

[00:15.080 --> 00:41.940] Когда человек, который начинает учить Python, спрашивает, что же мне почитать, наверное, процентов 90 разработчиков говорят «Читай Луца, там все есть, это такой огромный двухтомник, он жутко нудный, он очень подробный, он достаточно плохо переведен, иногда он очень косноязычен».

[00:41.940 --> 00:44.940] Но, к сожалению, пожалуй, лучше, чем…

[00:45.080 --> 00:50.140] Этот учебник просто по базовому Python пока ничего нет.

[00:51.260 --> 01:09.380] Мы будем читать это дело по главам, на каждую главу у нас будет, скорее всего, отдельный оратор, и он будет давать презентацию, а по ходу вот этой презентации я буду задавать вопросы, каверзные и не очень, ну или подкидывать тему для обсуждения.

[01:09.380 --> 01:15.060] Мы тут не одни зрители, которые безмолвно присутствуют у нас.

[01:15.080 --> 01:26.680] Сейчас в чате они тоже будут задавать вопросы, мы их будем или зачитывать в конце, или мы будем в процессе обсуждения с ними как-то подключаться.

[01:27.700 --> 01:35.300] Сегодня на первую главу у нас будет читать Алексей Троянов. Привет, Алексей.

[01:37.980 --> 01:38.960] Добрый день.

[01:39.540 --> 01:43.760] Ну что, я не буду утомлять тебя софтами.

[01:45.080 --> 01:46.600] Поехали, наверное, да?

[01:49.260 --> 01:50.920] Включаю демонстрацию экрана?

[01:51.140 --> 01:52.620] Да, совершенно так.

[01:55.220 --> 01:55.980] Отлично.

[01:58.820 --> 02:05.240] Добрый день, меня зовут Алексей, я студент факультета Backend Python, 27-я когорта.

[02:06.140 --> 02:12.080] Сегодня предлагаю обсудить первую главу замечательной книги Марка Лутца «Изучаем Python».

[02:12.080 --> 02:13.080] Что?

[02:15.080 --> 02:24.120] Обсудим сегодня достоинства и недостатки Python, технические превосходства по сравнению с другими языками программирования.

[02:27.120 --> 02:28.820] Так, погоди минуту.

[02:29.300 --> 02:32.360] Сразу некоторые вопросы у меня есть, да?

[02:33.020 --> 02:36.000] Я обратил внимание, что там пятое издание.

[02:37.460 --> 02:40.900] Пятое издание, оно вышло на английском.

[02:41.020 --> 02:45.000] Это оно на русском вышло в 19-м году, в 19-м, 20-м.

[02:45.080 --> 02:48.800] А на английском оно вышло в 2013-м.

[02:49.900 --> 02:53.040] Вот, это очень важно, это нужно понимать.

[02:53.660 --> 02:57.100] Издание рассчитано на версию языка Python.

[02:58.620 --> 03:02.700] Там даже 2.7 был, но мы его как-то, наверное, не будем, да?

[03:03.280 --> 03:06.120] Оно рассчитано на Python 3.3.

[03:06.720 --> 03:08.880] Что это значит, как по-твоему?

[03:10.520 --> 03:11.800] Какие недостатки?

[03:11.800 --> 03:15.060] То есть нам нужно понять границу применимости наших знаний.

[03:15.080 --> 03:18.020] Которые мы сейчас будем обсуждать.

[03:21.560 --> 03:24.800] Я не могу сходу сказать разницу.

[03:25.420 --> 03:32.040] Ага, ну смотри, то есть если говорить про сейчас вот уже 3.10, мы все-таки живем в 22-м.

[03:33.740 --> 03:37.340] Ну, очевидно, что какие-то фичи добавлены.

[03:37.580 --> 03:38.780] Это совершенно точно.

[03:38.780 --> 03:45.000] Я вот, например, навскидку могу сказать про TypeHint, про дата классы.

[03:45.080 --> 03:59.320] Про какие-нибудь там кучу сахара, тому, что у нас нужно объединять словари, например, новыми вот этими вещами.

[03:59.780 --> 04:04.200] Морж-оператор, иниты в пакетах и прочее.

[04:04.340 --> 04:09.480] В принципе, для первой главы не критично, но для первой главы критично может быть вот что.

[04:09.480 --> 04:13.120] Когда мы сравниваем это дело с какими-то другими языками,

[04:13.520 --> 04:14.480] вполне возможно, что ситуация...

[04:15.080 --> 04:22.040] И вот эти граничные условия нам нужно иметь в виду, когда мы читаем эту книжку.

[04:22.140 --> 04:29.440] Книжка хорошая, но у нее есть некоторые моменты просто из-за того, что лутц не может.

[04:30.160 --> 04:31.900] Ну, это так отскочим в сторону.

[04:32.460 --> 04:33.080] Давай дальше.

[04:35.900 --> 04:38.520] Итак, почему же люди используют Python?

[04:40.240 --> 04:44.300] В наши дни, естественно, имеется огромное количество языков программирования.

[04:45.920 --> 04:49.040] Во-первых, это качество программного обеспечения.

[04:49.900 --> 04:55.460] Пишем код мы один раз, а перечитываем его или используем многократно.

[04:56.020 --> 04:59.040] Поэтому по замыслу Python должен быть читабельным,

[04:59.100 --> 05:03.200] а значит, сопровождение его намного легче, чем традиционных языков.

[05:04.340 --> 05:12.860] Тем более при соблюдении PEP-8 он понятен, даже если написан не вами, кем-то сторонним.

[05:13.960 --> 05:14.200] Второе.

[05:15.080 --> 05:16.980] Продуктивность труда разработчика.

[05:18.020 --> 05:20.140] Код Python более краток и лаконичен.

[05:21.540 --> 05:30.160] Аналогичный код Python по сравнению с плюсами или Java примерно на четверть короче.

[05:32.500 --> 05:38.280] А это не только меньшие затраты при наборе кода, но и при отладке или сопровождении.

[05:40.220 --> 05:40.980] Третье.

[05:41.040 --> 05:42.980] Почему выбирают Python?

[05:43.120 --> 05:44.280] Это переносимость программ.

[05:44.360 --> 05:44.920] То есть кроссоверность.

[05:44.920 --> 05:45.880] Кроссплатформенность.

[05:47.200 --> 05:51.040] В большинстве случаев программы, написанные на одной платформе,

[05:51.100 --> 05:56.500] достаточно просто скопировать на другую операционную систему, где установлен Python.

[05:58.220 --> 05:59.680] Четвертая причина.

[06:00.220 --> 06:03.560] Это, соответственно, большое количество поддерживающих библиотек.

[06:04.580 --> 06:10.100] В стандартной библиотеке входит большое количество различных расширений для различных прикладных задач.

[06:10.860 --> 06:14.900] Кроме того, существуют библиотеки сторонних разрешений.

[06:14.920 --> 06:19.920] У работчиков также охватывающие все сферы прикладного программирования.

[06:20.980 --> 06:26.640] От веб-сайтов конструирования до низкоуровневого доступа к железу.

[06:28.380 --> 06:33.920] Еще одна из причин – это интеграция компонентов.

[06:35.600 --> 06:39.900] Python позволяет легко обращаться к библиотекам и взаимодействовать с другими частями программы,

[06:40.260 --> 06:41.960] которые написаны на других языках.

[06:41.960 --> 06:43.500] А вот здесь вопрос.

[06:43.680 --> 06:44.000] Погоди.

[06:44.920 --> 06:46.480] Отлично, что коснулся.

[06:46.560 --> 06:48.360] А каким образом вообще это можно сделать?

[06:48.680 --> 06:51.200] Как ты себе это представляешь на практике?

[06:53.580 --> 06:54.660] Библиотекам обращаться.

[06:55.040 --> 06:58.900] То есть мы импортируем библиотеку и, соответственно, ее используем у нас.

[06:59.040 --> 06:59.740] Да, правильно я понимаю?

[07:00.000 --> 07:00.200] Да.

[07:00.740 --> 07:03.460] Можно ведь, наверное, еще другие какие-то варианты?

[07:03.700 --> 07:04.420] Какие есть мысли?

[07:07.320 --> 07:11.300] Мы можем, например, код на плюсах взять и засунуть в питаньячий файл?

[07:11.860 --> 07:13.220] Или не можем?

[07:14.920 --> 07:18.920] Скорее всего, можем, только я не пользовался такой хищникой.

[07:19.480 --> 07:20.520] Окей, ладно.

[07:20.640 --> 07:21.100] Ну смотри.

[07:21.620 --> 07:24.160] Дальше как бы у нас есть…

[07:24.160 --> 07:27.560] На самом деле я вот себе вижу три варианта, как мы все это можем сделать.

[07:28.160 --> 07:31.100] То есть мы, например, запускаем программу на плюсах.

[07:31.540 --> 07:38.260] Она что-то там делает у себя, да, потом вызывает, импортирует у себя скрипт в питон, да,

[07:38.420 --> 07:40.260] и что-то там ему отдает.

[07:40.260 --> 07:44.260] То есть фактически весь код питона мы можем запускать,

[07:44.920 --> 07:48.680] под контролем кода на C+, ну на плюсах на C++.

[07:49.540 --> 07:54.840] Второй вариант – это что мы можем действительно взять и вызвать какую-нибудь функцию

[07:54.840 --> 08:00.080] из модуля библиотеки, вот когда мы исполняем питон,

[08:00.540 --> 08:02.880] и он плюс работает внутри питонного.

[08:03.640 --> 08:08.420] Ну а третий вариант, мне кажется, что есть, например, две независимые программы,

[08:08.560 --> 08:10.580] одна на плюсах, другая на питон,

[08:10.580 --> 08:14.860] ну и они каким-то образом обмениваются информацией.

[08:14.980 --> 08:21.600] То есть там общая память, может быть, каналы, сигналы, азбука Морзе, REST, все что угодно,

[08:21.700 --> 08:26.280] но они работают независимо и, соответственно, между собой обмениваются.

[08:26.900 --> 08:31.940] Как по-твоему, какие вообще варианты из этих возможны?

[08:33.120 --> 08:37.080] Ну, мне кажется, первый и второй – это точно библиотеки и вызов кода.

[08:37.080 --> 08:44.240] Ну, в принципе, обмен через какой-то стандартный интерфейс, там REST это или что-то другое,

[08:44.260 --> 08:44.840] вполне возможно.

[08:44.920 --> 08:45.800] Возможно, почему нет.

[08:46.000 --> 08:48.880] То есть, получается, три варианта, как мы можем интегрировать.

[08:49.020 --> 08:52.480] Окей, давай про наслаждение программированием поговорим.

[08:52.600 --> 08:56.160] Что для тебя лично наслаждение программированием?

[08:56.880 --> 08:57.760] В чем кайф?

[08:58.680 --> 08:59.200] Кайф?

[08:59.940 --> 09:05.480] Ну, во-первых, если говорить простым языком, это заставить железяку сделать то, что тебе нужно.

[09:06.200 --> 09:12.240] А с другой стороны, это должно быть увлекательно и приятно, и легко,

[09:13.260 --> 09:14.760] а не погружаясь там.

[09:14.920 --> 09:24.880] То есть, у тебя ломает там обрабатывать ошибки, управлять памятью и все, вот такую вещь.

[09:26.080 --> 09:32.280] Если это как бы интуитивно понятно и легко обрабатывается, почему бы нет?

[09:32.280 --> 09:37.100] А если для этого нужно там в ассемблер лезть, то это, наверное, сложновато.

[09:37.440 --> 09:40.100] Ну, отлично. Давай поедем дальше.

[09:42.820 --> 09:43.680] Поедем дальше.

[09:44.920 --> 09:47.220] Так, о...

[09:47.220 --> 09:52.400] Автор задается вопросом, является ли питон языком написания сценариев.

[09:52.960 --> 09:56.320] Ох ты. А что такое вообще язык написания сценариев?

[09:57.160 --> 10:06.660] Ну вот, как раз в том тексте, который я сейчас написал для этого слайда и озвучиваю, как раз это и будет рассматриваться.

[10:06.660 --> 10:09.080] То есть, ну давай по-простому.

[10:09.420 --> 10:11.660] То есть, как ты себе это понимаешь?

[10:11.820 --> 10:14.460] То есть, что такое язык написания сценариев?

[10:14.920 --> 10:21.120] Не человеком, не луцевским языком, от которого мы как раз хотим отойти.

[10:21.720 --> 10:25.900] А вот как, на твой взгляд, вот это все...

[10:25.900 --> 10:33.600] Для меня ближе язык написания сценариев, это вот первый вариант, который на слайде есть, это инструменты командной оболочки.

[10:33.600 --> 10:40.740] То есть, сценарии для управления системой, какие-то скрипты, утилитки и подобное.

[10:41.060 --> 10:44.600] Слушай, ну логично, потому что в Unix-системе он, например, прямо встроен.

[10:44.920 --> 10:53.120] И, в общем-то, написать на нативном языке что-то, что работает с самой системой, ну, наверное, окей.

[10:53.580 --> 10:58.120] Насколько я помню, еще Raspberry Pi очень здорово это дело любит.

[10:58.860 --> 11:07.220] Там, в общем, Python тоже нативный язык, и это круто, потому что мы, получается, под себя гребем целый кусок интернета вещей.

[11:08.040 --> 11:08.580] Логично?

[11:09.040 --> 11:09.700] Да, абсолютно.

[11:09.700 --> 11:09.960] Вот.

[11:10.360 --> 11:14.700] Окей, я смотрю, у тебя есть процедурно-объектно-ориентированная функциональность.

[11:14.920 --> 11:18.620] Ориентированное программирование ты выделил, а в чем между ними разница вообще?

[11:21.320 --> 11:25.300] Ну, соответственно, это три парадигмы программирования.

[11:26.000 --> 11:37.780] Процедурное программирование – это стандартное программирование, которое было основополагающим, соответственно, последовательным выполнением каких-то там простых действий.

[11:37.780 --> 11:44.780] Потом пришло объектно-ориентированное программирование, где стали рассматриваться конструкции.

[11:44.920 --> 11:53.020] Конкретные объекты и подходы работы с ними, такие как инкапсуляция, наследование, полиморфизм.

[11:54.400 --> 12:02.240] Соответственно, по прошествии какого-то времени подошли к функциональному программированию.

[12:02.240 --> 12:08.260] В нем отсутствуют операторы условия, циклы.

[12:08.900 --> 12:14.900] Все используется в основном функции, которые могут вкладываться друг в друга, обертываться.

[12:14.920 --> 12:15.600] И так далее.

[12:17.840 --> 12:20.280] Окей, ладно, хорошо.

[12:21.140 --> 12:28.340] Но не вполне себе уверен, что это правильная формулировка, если честно.

[12:28.820 --> 12:37.820] Я надеюсь, что ребята, которые нас будут смотреть, они имеют что сказать по этому вопросу.

[12:37.820 --> 12:44.380] То есть, в общем-то, три парадигмы и разница между ними – это интересно, это познавательно.

[12:44.920 --> 12:46.380] Это очень классно.

[12:48.000 --> 12:55.400] То есть, вопрос возникает номер раз по точности формулировки, и вопрос номер два – по времени возникновения.

[12:55.620 --> 13:01.760] То есть, возникло ли функциональное программирование до или после ООП.

[13:03.380 --> 13:11.880] Предлагаю пока на этот вопрос не отвечать, посмотреть, что напишут нам наши замечательные зрители.

[13:12.440 --> 13:14.700] Я почти уверен, что у них есть что сказать.

[13:14.920 --> 13:16.180] Давай дальше посмотрим.

[13:20.800 --> 13:22.220] Переходим к недостаткам по этому.

[13:23.320 --> 13:23.680] Ага.

[13:25.420 --> 13:26.720] Да, давай по-простому.

[13:26.720 --> 13:30.380] Да, то есть, как бы знаю, что ты подготовился, прочитал текст.

[13:31.260 --> 13:41.420] То есть, кайф заключается в том, что у нас, у Питона есть, на нем писать быстро, но бесплатного ничего не бывает.

[13:42.120 --> 13:42.200] Да.

[13:42.780 --> 13:43.960] То есть, в чем минус?

[13:43.960 --> 13:44.720] За удобство.

[13:44.920 --> 13:46.060] За удобство надо оплатить скорость.

[13:46.060 --> 13:47.020] Конечно, конечно.

[13:47.940 --> 13:48.040] Да.

[13:48.600 --> 13:51.620] Почему частота внесения изменений – это минус по-твоему?

[13:51.820 --> 13:57.160] Наоборот, классно, мне кажется, язык развивается, придумываются новые фичи.

[13:57.780 --> 14:05.120] Да, но возникает вопрос с поддержкой собственных ранее написанных программ.

[14:05.600 --> 14:08.740] Соответственно, нужно отслеживать обратную совместимость своего кода.

[14:09.800 --> 14:10.960] Ну, в чем проблема?

[14:10.960 --> 14:13.960] В чем проблема, как бы, если ты пишешь…

[14:13.960 --> 14:19.280] Ну, просто более новые программы, они используют старые фичи.

[14:19.400 --> 14:21.020] То есть, старые фичи тоже можно использовать.

[14:21.540 --> 14:28.600] То есть, ну, например, та же самая строка, которая, не помню, в 3.6, по-моему, или в 3.7, в 3.6, по-моему, Питоне появилась.

[14:29.020 --> 14:29.960] Ну, до этого был формат.

[14:30.580 --> 14:34.240] Но в 10, например, формат тоже используется без проблем.

[14:34.860 --> 14:37.620] То есть, можно форматировать строчку другими методами.

[14:37.840 --> 14:39.160] То есть, обратная совместимость.

[14:39.160 --> 14:43.920] Мне кажется, все-таки частота внесения изменений – тут в чем проблема больше.

[14:43.960 --> 14:55.520] В том, что если ты, например, читаешь курс по Питону, ты такой, ааа, и тебе из зала такие, а это уже делается по-другому, чувак, ты чего?

[14:55.720 --> 14:57.320] Для преподавателей в основном, да.

[14:57.340 --> 15:04.960] Да, мне кажется, это вот у Луца, он это в минус сделал, это он просто про себя думает, что ж такое-то каждый год книжку писать.

[15:05.480 --> 15:07.100] Это я улучшусь, кто же ее купит.

[15:07.440 --> 15:13.100] Он не говорит про отрицательные стороны, он говорит именно про компромиссы, которые возникают.

[15:13.960 --> 15:17.780] Возникают из-за тех или иных событий.

[15:18.160 --> 15:22.640] Ну окей, но видишь, у тебя слайд недостатки Питона, поэтому я подумал, что…

[15:22.640 --> 15:25.960] А, у тебя недостатки – это скорость выполнения, а это компромиссы.

[15:26.280 --> 15:26.840] Окей.

[15:27.320 --> 15:29.420] Слушай, а батарейка в комплекте, что это значит?

[15:30.580 --> 15:39.440] Ну, с точки зрения Луца, батарейка в комплекте – это когда все включено, что с Питоном поставляется большое количество библиотек, инструментов.

[15:39.440 --> 15:43.840] И иногда привязывание к тому или иному инструменту.

[15:43.960 --> 15:45.980] И со временем может как бы…

[15:45.980 --> 15:49.260] Ну, ты его можешь перерасти, а поддерживать…

[15:49.260 --> 15:54.340] Бывает, что, как он говорит, батарейка изменяется или повреждается, или устаревает.

[15:54.500 --> 15:57.100] Соответственно, можно поддерживать этот код свой.

[15:57.140 --> 16:02.120] Слушай, ну, блин, проблема Legacy, она, в общем-то, характерна для программирования.

[16:02.740 --> 16:05.840] У меня есть ребята, которые работают в Яндексе.

[16:06.360 --> 16:08.480] Правда, они с этого проекта уже ушли.

[16:09.040 --> 16:13.840] У которых там в одном месте вообще джанго, не соврать, 1.3.

[16:13.960 --> 16:15.160] 1.3, что ли, 1.4.

[16:15.820 --> 16:24.960] Просто потому, что к ней приплели стороннюю библиотеку, которая как-то очень слабо обновляется, и ничего не получается.

[16:25.700 --> 16:31.160] В общем, с одной стороны, оно работает, с другой стороны, как-то развивать это сложно.

[16:31.380 --> 16:33.920] С третьей стороны, ну, блин, работать не трогай.

[16:34.460 --> 16:36.960] И деньги на рефакторинг как-то…

[16:36.960 --> 16:39.200] Ну, не нашлось и ресурсов, ни денег.

[16:39.580 --> 16:40.200] Возможно, сильно…

[16:40.200 --> 16:41.560] Версия будет дороже себе.

[16:42.180 --> 16:42.660] Согласен.

[16:42.660 --> 16:43.940] То есть, как бы здесь…

[16:43.960 --> 16:49.160] Ну, мне кажется, эта проблема, в общем, даже не питона, а проблема самого программирования.

[16:49.580 --> 16:53.200] Слушай, а недостатки для проектов с открытым кодом – это ты про что?

[16:53.420 --> 16:54.480] Что ты имеешь в виду?

[16:55.640 --> 17:04.900] Ну, вот ПО с открытым кодом бывает больше иногда похоже на анархию, как пишет Луц.

[17:04.900 --> 17:12.900] И ссылается на то, что, несмотря на выстроенный процесс контроля ошибок, иногда в релизы попадают…

[17:13.960 --> 17:17.460] Некоторые там ошибки нерабочие.

[17:17.960 --> 17:23.240] Например, в версии Python 3.2.0 поступила с неработающей консольной функцией input.

[17:23.560 --> 17:28.220] Окей, обрати внимание, что отличный свежий пример 3.2.0.

[17:29.940 --> 17:30.980] Ну, понятно, что…

[17:30.980 --> 17:33.440] Актуально было, наверное, году…

[17:33.440 --> 17:34.280] В 30-м году.

[17:35.280 --> 17:36.600] В 10-м, наверное.

[17:36.800 --> 17:37.520] В 10-м, ну…

[17:37.520 --> 17:38.140] Дай бог.

[17:38.660 --> 17:39.540] Вот, окей.

[17:39.540 --> 17:40.220] Ну, смотри.

[17:40.720 --> 17:42.540] На самом деле, вот по скорости выполнения…

[17:43.960 --> 17:44.840] Чатики пишут.

[17:45.000 --> 17:48.900] В общем-то, ну, такое на самом деле себе.

[17:49.560 --> 17:59.820] То есть, потому что какие-то вот bottlenecks, места, где есть проблемы, есть критичные вещи по поводу того, что нужно быстро.

[17:59.820 --> 18:06.040] Но мы можем кусок этот на плюсах переписать и взять эту библиотечку использовать для вычислений.

[18:06.460 --> 18:10.960] То же самое NumPy, да, то же самое Луц, он, в общем-то, на него ссылается.

[18:11.540 --> 18:13.880] Он очень хорошо умеет и математику, и быстро.

[18:13.960 --> 18:16.620] Отличный вариант, то есть, ну, такое.

[18:17.120 --> 18:28.700] Да, я хочу сказать, я извиняюсь, что слайд по недостаткам питона, поэтому нужно было что-то написать, нельзя написать, что у него нет недостатков.

[18:28.860 --> 18:30.060] Блин, ну…

[18:30.060 --> 18:41.040] Понятно, что дальше как раз на следующих слайдах будет говориться о том, что все узкие места можно переписать на C, C++, и будет вам счастье.

[18:41.040 --> 18:41.700] Ну, да.

[18:42.240 --> 18:43.040] Ну, например, вот народ пишет, что…

[18:43.960 --> 18:46.720] В 3.9 утечка памяти обнаружилась.

[18:46.940 --> 18:58.940] Вполне возможно, что наши любимые зрители могут еще какие-нибудь баги, косяки, с которыми они столкнулись в работе с питоном, в общем-то, написать в комментариях.

[18:59.300 --> 19:00.980] С большим удовольствием почитаю.

[19:01.300 --> 19:05.280] Это, в общем-то, хейт питона, это наше все, это интересно.

[19:05.800 --> 19:11.400] Слушай, ну, а вот если говорить, просто я не знаю твоих следующих слайдов, там будет про пеп?

[19:12.440 --> 19:13.540] Нет, про пеп нет.

[19:13.960 --> 19:15.940] Отлично, самое время поговорить про пеп.

[19:15.940 --> 19:24.500] Ну, вот смотри, то есть, по сути, вот эти вот проблемы, они, по большому счету, лечатся тем, что мы используем пеп.

[19:24.860 --> 19:37.320] То есть, все изменения сначала тщательно обкатываются, должны, ну, вот в идеале, и потом только обсуждаться комьюнити, да, и потом только приниматься или не приниматься.

[19:37.320 --> 19:40.320] Ну, кажется, что…

[19:43.960 --> 19:47.660] Лучшего варианта вообще нельзя придумать.

[19:48.160 --> 19:50.420] То есть, открытый код, закрытый код, какая разница?

[19:51.600 --> 20:11.600] Ну, одним еще из недостатков про открытый код, он говорит о том, что коммерческая выгода в закрытом коде, в коммерческом проекте обменивается на личные предпочтения текущего круга разработчиков, которые могут либо совпадать, либо не совпадать с вашими предпочтениями, с нашими программистами.

[20:12.420 --> 20:13.800] Ну, тут такая…

[20:13.960 --> 20:21.800] Такое дело, ну, как бы, есть специально обученный великодушный диктатор, я надеюсь, что гвида нам поможет.

[20:22.320 --> 20:26.340] Ладно, давай дальше посмотрим, что, какие следующие слайды ты еще подготовил.

[20:27.200 --> 20:29.700] Йо-хоу, системное программирование.

[20:30.300 --> 20:36.260] Давай прямо коротенько поговорим про вот это вот все.

[20:36.260 --> 20:43.260] Ну, системное программирование мы с тобой обсуждали, да, это же у нас всякие вот…

[20:43.960 --> 20:47.960] Интерфейсы, те утилитки, которые позволяют работать с системой.

[20:49.820 --> 21:00.320] То есть, ну, графические интерфейсы, это у него тоже какой-то, мне кажется, в книжке там какая-то старость написана, я почти уверен, что есть и более современные фреймворки.

[21:00.320 --> 21:13.320] Будет классно, кстати, если ребят напишут в комментариях, какими вот, какими фреймворками и какими вещами они используются, то есть для графических интерфейсов, для интернета.

[21:13.960 --> 21:25.100] Для интеграции компонентов, для баз данных, для, ну, вот, прототипирования, это такая вещь, то есть, в общем-то, здесь, наверное, это как-то более объединенное, да.

[21:25.680 --> 21:33.500] Ну, наука, это, скорее всего, MyPy какой-нибудь, то есть, наверное, ничего нового не придумали, хотя будет классно, если вы нам расскажете.

[21:33.500 --> 21:41.480] Ну, игры, роботы и таблицы Excel и всякое вот это вот, там вообще очень большой пласт.

[21:42.200 --> 21:43.500] И здесь, конечно, лучше он…

[21:43.960 --> 21:51.060] Немножко устарел, потому что здесь можно добавить еще и компьютерное зрение, распознавание образов и вот это вот…

[21:51.060 --> 21:52.540] Искусственный интеллект, да, то есть…

[21:52.540 --> 21:59.880] Да, да, да, то есть, как бы, machine learning, вот это вот, всю вот эту вот бодягу можно, можно, в общем-то, сделать.

[22:00.120 --> 22:10.780] Вот народ накидывает, ухоу, киви, кьюти, тинктер, SQL, OpenExcel, OpenPyExcel, нифига себе.

[22:10.800 --> 22:12.860] Кинтер как раз у Луцы и указывается.

[22:12.860 --> 22:13.860] А, слушай, ну, видимо…

[22:13.960 --> 22:15.440] Видимо, кто-то пережил.

[22:15.540 --> 22:22.880] Я у него посмотрел фреймворки, которые он рассказывает про веб.

[22:23.080 --> 22:32.980] Честно говоря, я только Django знаю, но, наверное, остальные куда-то канули в лето, и почему-то нету у него фласка, непонятно.

[22:33.620 --> 22:38.400] То есть, по-хорошему, здесь можно взять и накидать…

[22:38.400 --> 22:40.720] У него, кстати, этого фастапи тоже нет.

[22:40.960 --> 22:43.720] Вот, взять и накидать всяких интересных названий.

[22:43.960 --> 22:48.200] Вот, взять и накидать всяких интересных названий, которые вообще часто используются и используются в вашей жизни.

[22:48.640 --> 22:57.860] Ребят, не стесняйтесь, давайте сделаем книжку Луца еще лучше и еще интереснее тем, что она будет актуальная.

[22:58.940 --> 23:02.100] Да, еще тортой съесть, вот, для баз данных, я точно знаю.

[23:03.200 --> 23:05.940] Поехали дальше, посмотрим, что еще.

[23:08.400 --> 23:08.880] Вау.

[23:10.680 --> 23:13.380] Ну, соответственно, недостатки посмотрели.

[23:13.960 --> 23:14.820] Технические превосходства.

[23:16.940 --> 23:18.960] Прекрасно, давай поговорим про технические.

[23:19.300 --> 23:22.420] Объектно-ориентированный, функциональный мы уже разобрали чуть раньше.

[23:23.980 --> 23:30.960] Соответственно, в книжке говорится о том, что он первоначально был…

[23:32.960 --> 23:35.560] Потерялся сейчас секунду.

[23:37.000 --> 23:42.720] В общем, он легкий скриптовый язык.

[23:42.880 --> 23:43.560] Плюс подключение.

[23:43.960 --> 23:46.160] Подключены к нему объектно-ориентированные фишки.

[23:46.640 --> 23:48.440] Плюс добавилась функциональность.

[23:49.140 --> 23:51.300] Ну, то есть он сначала был процедурщиной, да?

[23:51.420 --> 23:55.280] То есть он был процедурным языком, а потом на него всякое нарастили.

[23:55.380 --> 23:57.020] Окей, бесплатно – это круто.

[23:57.280 --> 24:01.920] Слушай, интересно, есть вообще нормальные платные языки по подписке, прикинь?

[24:01.920 --> 24:05.920] То есть ты можешь программировать по подписке каждый месяц за то, что ты программируешь.

[24:07.260 --> 24:12.540] Да, на чудесном языке ты должен бабла платить.

[24:12.660 --> 24:13.920] Я, честно говоря…

[24:13.960 --> 24:16.220] Мне сложно сейчас представить.

[24:16.240 --> 24:19.980] Тут, наверное, все-таки говорится о открытых и закрытых.

[24:19.980 --> 24:24.980] В том смысле, что тот же самый Visual Basic, он пропиетарный.

[24:26.320 --> 24:31.820] Те, которые разрабатываются закрытыми коммерческими предприятиями, компаниями.

[24:31.840 --> 24:33.160] Гошечка, например, да?

[24:34.180 --> 24:37.720] А, слушай, нам нельзя рекламировать продукты этой фирмы.

[24:38.260 --> 24:41.240] Мы не рекламируем, мы просто говорим, что есть такой язык.

[24:41.360 --> 24:43.180] Не пользуйтесь им, это плохой язык.

[24:43.960 --> 24:47.200] Ну, или пользуйтесь, но на свой страх и риск.

[24:47.340 --> 24:49.400] Вы должны понимать и осознасить.

[24:49.720 --> 24:52.820] Слушай, переносимый, то есть есть непереносимый…

[24:52.820 --> 24:59.800] Непереносимый, ну, это такая сложность перевода, соответственно, это о кроссплатформенности.

[25:00.240 --> 25:06.240] Ну да, то есть, наверное, у нас переводчик не программист это все дело переводил,

[25:06.480 --> 25:08.660] поэтому действительно он будет исполнять.

[25:08.660 --> 25:13.760] Хотя, насколько я помню, для винды там нам и интерпретатор все-таки надо ставить.

[25:13.960 --> 25:17.200] Ну да, поэтому ставится интерпретатор.

[25:17.340 --> 25:21.040] Да, он из коробки в винде не идет, ну, как бы, ладно, окей.

[25:21.460 --> 25:26.640] Мощный, слушай, динамическая типизация, честно говоря, здесь он прям вообще попал не туда,

[25:27.360 --> 25:36.720] потому что сейчас питон со всей мочи скачет в сторону тайп-хитинга, вот этой вот аннотации типов,

[25:37.600 --> 25:40.960] потому что, чтобы, видимо, работать быстрее, наверное, так.

[25:40.960 --> 25:43.960] Потому что аннотация типов не отменяет же динамику.

[25:43.960 --> 25:44.840] Динамическая типизация.

[25:45.040 --> 25:49.580] Нет, она не отменяет, но получается какая-нибудь интересная штука, на двух стульях сидим,

[25:49.680 --> 25:57.300] то есть, с одной стороны, мы как бы типизируем динамически, но, в общем-то, мы при инициализации,

[25:57.400 --> 26:05.640] мы явно задаем класс и, в общем-то, сильно ругаемся, если это все, задаем тип и сильно ругаемся,

[26:05.640 --> 26:13.240] если посредством MyPy, если у нас этот вот тип, он как-то...

[26:13.960 --> 26:15.460] Не совпадает.

[26:15.760 --> 26:16.200] Не зашел.

[26:16.580 --> 26:17.360] Да, не зашел.

[26:17.440 --> 26:25.360] Слушай, автоматическое управление памяти, блин, ну, тут тоже много копий сломано, такое дело, окей.

[26:26.040 --> 26:27.320] Да, сборщика мусора.

[26:27.700 --> 26:35.680] Да, то есть, гербочколлектор, ну, в принципе, у него неплохой, насколько я могу судить, да, и это удобнее, чем руками.

[26:35.980 --> 26:38.540] Но, тем не менее, иногда он и лажает.

[26:38.640 --> 26:42.820] Я помню, что кто-то даже отключал в каких-то проектах, что-то там читал.

[26:43.400 --> 26:43.880] Да, вот бестификат.

[26:43.960 --> 26:48.560] И степизация как-то, ну, достаточно сложная и неудобная.

[26:48.680 --> 26:57.940] Так что есть ощущение, что они раскаялись и постепенно ведут в сторону, ну, путем таким вот хитрым введением тайп-хинтов.

[26:58.420 --> 27:00.460] Слушай, что такое программирование в большом?

[27:00.940 --> 27:12.460] Ну, это как раз о том, что он является как бы надстройкой над другими, может являться надстройкой над другими языками, интегрирует в себе...

[27:13.960 --> 27:21.420] Скриптовый легкий язык, с которого можно вызывать там C, C++, другие языки программирования.

[27:21.420 --> 27:34.960] Окей, то есть, вот это программирование в большом, то есть, вот, по сути, это клей, то есть, он может быть такой универсальной прокладкой между джабой, да, то есть, и вот это все.

[27:35.440 --> 27:41.300] Окей, встроенные типы объектов, это речь идет про всякие там листы, дикты и вот это все, да?

[27:41.320 --> 27:42.080] Да, да, абсолютно.

[27:42.160 --> 27:43.640] Отлично. Инструменты?

[27:43.960 --> 27:44.200] Да.

[27:44.980 --> 27:51.320] О том, что они изменяются динамически в процессе и легко встроены?

[27:51.320 --> 28:04.700] Да, что с ними удобно работать, я согласен. В общем-то, у Python достаточно приличный вот этот вот кусок для работы с данными, прям все хорошо.

[28:05.320 --> 28:07.460] Ну, инструментики, это речь...

[28:07.460 --> 28:09.500] О встроенных библиотеках.

[28:10.140 --> 28:13.060] Ну, да, то есть, как бы достаточно много на все случаи жизни.

[28:13.060 --> 28:13.860] Те самые батарейки...

[28:13.960 --> 28:19.960] Батарейки, о которых так мы немножко ругались, но на самом деле они как раз дают как раз мощь.

[28:21.080 --> 28:22.380] Да, радости к программированию.

[28:22.820 --> 28:26.960] Согласен, та самая радость. Так, ладно, утилиты, это...

[28:26.960 --> 28:30.960] Это сторонник разработчиков, соответственно, то, что большое количество инструментов...

[28:31.720 --> 28:35.100] Ну, то есть, сел и написал, то есть, по сути...

[28:35.100 --> 28:37.260] Выложил на гид, пользуйтесь.

[28:38.140 --> 28:41.520] Ну, логично, он же, в общем-то, открытый язык, почему нет?

[28:41.520 --> 28:42.520] Окей, смешиваемый.

[28:43.960 --> 28:46.300] С кем? А, слушай, смешать, но не взбалтывать.

[28:46.720 --> 28:50.100] Или, наоборот, взболтать, но не смешивать. Не помню, как там правильно.

[28:51.760 --> 28:55.620] Ну, это как раз, наверное, повтор про программирование в большом,

[28:55.620 --> 29:04.460] о том, что самыми разнообразными способами можно вызывать IP-интерфейс на C, имеющийся в Python,

[29:05.100 --> 29:08.620] и, наоборот, там, как бы...

[29:08.620 --> 29:10.480] Ну, да, слушай, ну, возможно.

[29:10.480 --> 29:11.540] Не вызывать, да.

[29:12.040 --> 29:13.880] Да, так, ну, простой в использовании.

[29:13.960 --> 29:16.840] И изучение, все признается в сравнении.

[29:16.960 --> 29:17.640] Да, относительно.

[29:18.120 --> 29:21.760] Basic намного проще, а уж Pascal-то, ба...

[29:21.760 --> 29:25.920] Ну, у Basic есть другие недостатки.

[29:27.160 --> 29:33.440] Да, слушай, ну, везде, то есть, к ним любой язычок можно взять и недостатков-то доковырять.

[29:33.860 --> 29:36.300] Слушай, группа Мойти Пайтон, ты что-нибудь смотрел из нее?

[29:38.320 --> 29:42.200] Узнал только, когда начал приобщаться к Python, не смотрел ничего.

[29:43.100 --> 29:43.660] Такие.

[29:43.960 --> 29:45.900] Поверхностные знания.

[29:46.180 --> 29:46.620] Ясно.

[29:46.680 --> 29:51.780] Ну, там, знаешь, такой интересный английский юмор, достаточно специфический.

[29:51.880 --> 29:55.300] Слушай, это не сериал, это группа как раз комиков.

[29:55.320 --> 29:55.840] Комик-группа.

[29:56.060 --> 30:00.180] Да, и они делали в том числе и сериал, там, про Грааль что-то.

[30:01.340 --> 30:07.280] Кстати, там очень классный, он очень легкий язык, сильно рекомендую его для изучения английского.

[30:07.440 --> 30:11.400] Ну, например, с подстрочником, у них хорошие произношения, мне очень нравится.

[30:12.300 --> 30:13.760] Теперь, внимание, вопрос.

[30:13.960 --> 30:19.940] Ну, назван он в честь группы Мойти Пайтон, а почему на логотипе у него питон?

[30:22.900 --> 30:32.480] Возможно, чтобы как-то зашифровать это, чтобы не серьезно как-то в честь группы Мойти Пайтон комиков называть.

[30:32.680 --> 30:35.240] Да нет, слушай, на самом деле там интересная история.

[30:35.340 --> 30:42.140] Я погуглил КС Грешин, но это было давно, потому что мне тоже было интересно, почему Пайтон, а вообще питон.

[30:42.600 --> 30:43.940] Просто Гвида, он тот еще...

[30:43.960 --> 30:48.180] Он еще раздолбай оказался, ему, в общем-то, было все равно на нейминг абсолютно.

[30:48.920 --> 31:00.140] И когда ему нужно было придумать язык, там что-то там с ABC у них было, он взял и сказал, ну, пусть будет Пайтон в честь Мойти Пайтона.

[31:00.960 --> 31:01.180] Вот.

[31:01.400 --> 31:05.720] А логотип просто взял первый попавшийся шрифт и написал слово питон.

[31:05.880 --> 31:06.040] Все.

[31:06.640 --> 31:07.400] Больше ничего.

[31:07.860 --> 31:13.460] Но в итоге язык начал немножко стать популярным.

[31:13.960 --> 31:21.960] И люди, которые там писали книжки, еще что-то, начали извращаться и рисовать всяких змей на обложках.

[31:22.920 --> 31:25.480] Там на журналах, на сайтах и везде.

[31:25.760 --> 31:28.180] То есть шрифт, в общем-то, мало кто пользовался.

[31:28.700 --> 31:32.260] Если посмотрите Вику, то, в общем-то, можно понять, почему.

[31:32.340 --> 31:33.960] Он такой страшненький, из шариков состоит.

[31:35.060 --> 31:36.540] Он ужасен на самом деле.

[31:36.840 --> 31:37.900] Настолько пипец.

[31:38.340 --> 31:40.680] Почему у него крыса нарисована, а не питон?

[31:40.960 --> 31:43.580] Ну, потому что, наверное, питон ест крыс.

[31:43.960 --> 31:45.280] Это логично.

[31:45.820 --> 31:51.760] И вот они в 2006-м придумали логотип как раз с двумя желтыми-синими питонами.

[31:52.500 --> 31:53.820] Ну, и обновленный шрифт.

[31:54.180 --> 31:58.920] То есть просто люди заставили взять и все-таки...

[31:58.920 --> 32:02.120] Люди проголосовали не за Мойти Пайтона, а за змей.

[32:02.660 --> 32:03.360] За ребрендинг.

[32:03.600 --> 32:04.600] За ребрендинг, да.

[32:04.760 --> 32:06.380] Ну что, слушай, звучит как тост.

[32:06.440 --> 32:07.640] Ну что, за ребрендинг.

[32:07.920 --> 32:08.880] Как тост.

[32:09.600 --> 32:10.920] Это у тебя последний слайд.

[32:11.460 --> 32:12.020] Еще один.

[32:12.020 --> 32:13.020] Оу-хоу.

[32:13.960 --> 32:17.140] Так, ладно, смотрите.

[32:17.320 --> 32:21.560] Здесь, наблюдая я много вымерших языков...

[32:21.560 --> 32:24.960] Ну, абсолютно, да, если книжка-то старенькая.

[32:25.240 --> 32:25.720] Вот.

[32:25.820 --> 32:27.240] Ну, как бы, Перл...

[32:27.240 --> 32:29.400] Кстати, интересно, на Перле кто-нибудь пишет?

[32:30.220 --> 32:31.380] Прямо вот активно сейчас.

[32:31.660 --> 32:32.340] Я не уверен.

[32:33.740 --> 32:35.960] Это можно посмотреть в этом...

[32:36.560 --> 32:37.500] Как он называется-то?

[32:37.600 --> 32:40.720] Есть сравнение языков программирования по популярности.

[32:41.000 --> 32:42.140] Ой, слушай.

[32:42.840 --> 32:43.720] Там достаточно столь...

[32:43.960 --> 32:56.020] Странные у них метрики, то есть там они меряются, например, количеством запросов, еще что-то, количеством реп в гите на этом языке.

[32:56.020 --> 33:00.540] Я бы не стал прямо вот говорить, что это вот прям вот так, вот так.

[33:01.040 --> 33:08.320] Так, друзья мои, кто нас смотрит, напишите языки, на которых вы разрабатываете.

[33:08.640 --> 33:13.820] Будет очень классно, если найдутся люди, которые пишут на Алоа.

[33:13.960 --> 33:16.240] На Перле, на ТСЛ.

[33:16.660 --> 33:17.600] Это будет круто.

[33:17.680 --> 33:19.860] Это будет вообще не мейнстрим ни разу.

[33:20.080 --> 33:21.920] Это будет вообще очень круто.

[33:22.480 --> 33:23.960] Вот, Visual Basic тоже, это уважаемо.

[33:24.720 --> 33:25.620] Вопросов нет.

[33:25.820 --> 33:27.820] Остальные вроде ничего так.

[33:28.540 --> 33:33.160] Ну, давай как-то мы, наверное, будем...

[33:33.160 --> 33:36.960] Я для работы поописываю иногда на Visual Basic.

[33:37.600 --> 33:38.420] А, ну видишь.

[33:38.520 --> 33:39.860] По необходимости.

[33:40.580 --> 33:42.300] Слушай, а что Visual Basic делает?

[33:43.960 --> 33:45.620] Ну, он просто в...

[33:45.620 --> 33:51.420] Также можно скриптовым его использовать в ВБС для винды.

[33:53.040 --> 33:55.320] Ну, окей.

[33:55.460 --> 33:56.820] Ладно, это очень круто.

[33:57.720 --> 34:02.820] Так, ну давай стопшеринг и начнем подводить итоги.

[34:06.140 --> 34:08.380] Задаем вопросы, но вопросы не из книжки.

[34:08.500 --> 34:13.480] Вопросы из книжки были бы слишком простые.

[34:13.960 --> 34:15.100] Что скажешь?

[34:15.120 --> 34:17.260] В чате, что работает на СЛО.

[34:17.840 --> 34:19.580] Ну, в общем-то, круто.

[34:19.760 --> 34:20.720] Слушай, оказывается, нет.

[34:21.720 --> 34:27.120] Есть у нас, среди нас еще люди, которые умеют, любят и практикуют всякое.

[34:29.120 --> 34:31.700] Итого, питон нормальный, как по-твоему?

[34:33.000 --> 34:33.440] Лучший.

[34:33.860 --> 34:36.620] У меня последний слайд не показал.

[34:36.700 --> 34:38.940] Там написано, что он best of the best of the best.

[34:38.940 --> 34:42.940] Слушай, это нельзя, потому что это с точки зрения...

[34:42.940 --> 34:43.940] Это нарушение...

[34:43.960 --> 34:45.020] Это нарушает закон о рекламе.

[34:46.040 --> 34:48.960] Боюсь, что нас могут взять и...

[34:49.560 --> 34:51.920] Взять и это самое.

[34:52.440 --> 34:56.580] Окей, народ пишет, что есть параграф H, строка А.

[34:57.300 --> 34:57.700] Вопрос.

[34:59.020 --> 34:59.480] Кошмар.

[34:59.880 --> 35:01.200] Я не буду это все делать.

[35:01.360 --> 35:04.080] То есть, друзья мои, изучайте питон.

[35:04.220 --> 35:06.220] Питон классный, питон современный.

[35:06.620 --> 35:07.760] Питон развивается.

[35:08.440 --> 35:11.240] Читайте Лутца, слушайте Лутца вместе с нами.

[35:11.240 --> 35:13.240] Мы постараемся...

[35:13.960 --> 35:15.960] Мы постараемся копать немножко глубже, чем Лутц,

[35:16.040 --> 35:19.700] потому что книжка написана немножко давно.

[35:20.440 --> 35:26.620] И в следующих сериях мы обязательно будем, в общем-то, спрашивать себя,

[35:27.080 --> 35:30.960] а как по-современному мы будем смотреть на дедушку...

[35:32.160 --> 35:34.160] Как, кстати, у него имя?

[35:34.620 --> 35:35.240] Марк.

[35:35.480 --> 35:35.800] Вот.

[35:35.920 --> 35:37.060] На дедушку Марка.

[35:37.700 --> 35:38.300] Вот.

[35:38.300 --> 35:43.800] И, соответственно, немножко помогать ему сделать книжку актуальной.

[35:43.960 --> 35:48.960] Дальнее, внося новые современные модные техники.

[35:49.640 --> 35:51.220] Ну что, а на сегодня это все.

[35:51.760 --> 35:53.640] Спасибо большое, что пришел.

[35:54.000 --> 35:55.020] Было очень круто.

[35:55.120 --> 35:55.980] Мне понравилось.

[35:56.360 --> 35:56.880] Динамично.

[35:56.980 --> 35:57.580] Тебе тоже понравилось.

[35:58.680 --> 36:01.100] Ну что, живите долго и процветайте.

[36:02.200 --> 36:02.840] До свидания.

[36:07.980 --> 36:08.080] Продолжение следует.