

The background is a dark blue gradient with a subtle pattern of small white dots. On the left side, there are several overlapping circular elements. A large, semi-transparent scale with numerical markings from 140 to 260 in increments of 10 is visible. Several concentric circles and arcs are drawn in a lighter blue color, some with arrows indicating a clockwise direction. The overall aesthetic is technical and modern.

# PYTHON ДЛЯ БИЗНЕС- АНАЛИТИКОВ

ФЕВРАЛЬ 2018

СИДОХИН Ф.А.

# СОДЕРЖАНИЕ

- Задачи бизнес аналитика
- Python для бизнеса
- Работа с Python
- Парадигма Python
- Парадигма Программирования
- Ваше первое задание



# ЗАДАЧИ БИЗНЕС АНАЛИТИКА

- Сбор данных об операциях компании и давать прогнозы
- Следить за рыком, трендами и давать рекомендации руководителям
- Находить новые направления развития бизнеса
- Производить отчеты об операциях компании
- И многое другое...

# ЗАДАЧИ БИЗНЕС АНАЛИТИКА

- В основе работы аналитика лежат два важные вещи:
  - Обработка данных
  - Делать выводы на основании собранных данных
- Сегодня системы собирают большое кол-во данных которые можно анализировать и получать ценную информацию
- Если раньше аналитики работали с Excel-ом то теперь этого уже не достаточно



# ЗАДАЧИ БИЗНЕС АНАЛИТИКА

- И вот появляется Python
- Что такое Python? "Это язык программирование для чайников"
- Ну не совсем так но Python проще чем Java, C++ и многие другие языки ибо он позволяет пренебрегать многими тонкостями которые не прощают другие языки
- В нем есть много уже заранее восторенных функции которые не надо реализовывать с 0 или как-то запихивать чужой код в свой.
- Чтобы работать в Python не надо страдать как с Java и C++, не надо думать о памяти, железе и других геморроях!

# PYTHON ДЛЯ БИЗНЕСА

- Так что за задачи мне может помочь решить Python? Это же программирование а у меня бизнес проблемы!
- Вот примеры вопросов для которых вполне может быть нужен Python:
  - Мы сделали акцию – хотим понять притянули ли мы новых покупателей?
  - Как праздники влияют на наш продукт? Продаем ли мы больше или меньше?
  - Какие клиенты у нас постоянные? Что у них общего?
- Казалось бы что эти вопросы тривиальны, и что их вполне можно решить в Excel-е...
- Но нет – особенно вопрос 3 теперь уже перешел способности этого продукта – давайте посмотрим пример!



# PYTHON ДЛЯ БИЗНЕСА

- Какие клиенты у нас постоянные? Что у них общего?
- Давайте предположим что у вас есть доступ к базе данных всех транзакции в магазинах (ну или в интернет магазине) вашего предприятия.
- С Python вы сможете например за пару часов увидеть такие вещи как например: у меня 90% клиентов также клиенты Банка Открытие.
- Вы бы сказали – ну и что? А толку мне знать это?
- Для маркетинга и руководства это очень ценная информация! Они могут на основе этих данных согласовать себе более выгодный тариф по обработке платежей, устроить какие-то скидки для клиентов других банков чтобы их притянуть!
- Но в этих данных есть больше!

# PYTHON ДЛЯ БИЗНЕСА

- Так как клиенты чаще всего используют одну карту вы можете посмотреть сколько клиентов у вас стали постоянными покупателями
- Вы можете с помощью машинного обучения найти возможную причину этой конверсии
- Вы можете тогда скрестить эти данные с вашими данными по кассе найти набор товаров которые постоянные клиенты покупают (а еще лучше в какие сезоны)
- Вы можете пойти еще дальше и например найти информацию о том какие товары у вас не пользуются спросом и заказывать их меньше то что приводит к оптимизации финансово состояния компании (и возможность более эффективно договорится с поставщиком)
- В одном предложении: Python позволит вам самостоятельно работать с данными и задавать те вопросы которые вам важны.



# РАБОТА С PYTHON

- Что мы получаем прямо “из-коробки”:
  - Для нашего курса мы будем использовать Anaconda Python 3 (<https://www.anaconda.com/download/>)
  - В ней идут все библиотеки включая инструменты для машинного обучения (sciKitLearn) а также библиотеки для статистики и более сложной работы с данными
  - Бонус: милый web-interface где мы можем программировать и сразу видеть результаты!
- Чтобы установить просто скачайте дистрибутив для вашей системы
- Установите
- Наберите “jupyter notebook” в терминале или в cmd
- И сразу откроется мир Python!

# РАБОТА С PYTHON

- Есть также более мощные IDE которые больше подходят для программистов (например IntelliJ PyCharm) – в них много дополнительных инструментов для профиляции, и работы именно с кодом
- Для совсем бородатых можно использовать VIM, Sublime Text, или даже notepad!
- Что важно: В Jupyter (Anaconda) нету никаких промежуточных шагов – код сразу исполняется, однако если вы потом хотите уже писать программу в Python и запускать ее вне Jupyter то обычно это делается следующей командой: `python nameOfFile.py`
- Хммм но это же похоже на Java? `java -jar myapp.jar`
- ДА!



# РАБОТА С PYTHON

- Python обладает свойством: write once, run anywhere (напиши один раз, исполняй везде)
- За исключением уникальных программ ваша программа взлетит на любом компьютере и на любой операционной системе
- Это прелесть Python – даже с Java не все так просто... а с C++ вы даже не хотите знать
- Это не делает эти языки хуже – у них своя область применения.
- Так что Python это от части язык для тех кто не программисты

# ПАРАДИГМА PYTHON

- Python относится к языкам которые называются объектно ориентированными
- Что это значит?
- Например вы хотите написать программу которая читает файл в котором есть данные о пользователе и вам печатает эту информацию на экран.
  - Первый подход – можно просто написать читатель файла и выводить на экран...
  - Но тут прояснилось что есть еще данные в какой-то базе данных... черт!
  - Второй путь был определить объект “пользователь” с некоторым набором полей и методом “напечатать на экран”
  - Тогда нам бы просто надо было добавить способ как данные из базы должны наполнять объект пользователь который мы так-же успешно бы печатали



# ПАРАДИГМА PYTHON

- В Python все что мы вводим (переменные, подключения, и т.п.) это объекты
- Объект это набор свойств и методов
  - Свойства это просто данные (представьте себе поля на формуляре или цвет, порода кота)
  - Методы это то что объект может выполнять (либо над самим собой – если опять взять пример кота то шипеть, либо над собой и другим объектом – например дать мелкого котика)
- Идея объектов это не совсем идея абстракции (хотя многие так говорят), это больше идея о создании элементов которые потом можно использовать заново, но вероятно с небольшими модификациями.

# ПАРАДИГМА PYTHON

- Исполнение кода в Python имеет две вариации
  - Straight through
  - If `__name__ == '__main__':`
- Straight through означает что все что вы написали в файле исполнится (и это всегда так)
- If `__name__ == '__main__':`: этот блок исполнится если вы запускаете программу
- Оба подхода одинаково хороши для наших целей второй не будет испоэооватся



# ПАРАДИГМА PYTHON

- Самое важное о Python:
  - Tabs (logic separation and sequencing)
  - None Type (x is None, x = None)
  - <,>==,<=,>=
  - If/elif/else
  - While ( while(true):)
  - For

# ПАРАДИГМА ПРОГРАММИРОВАНИЯ

- Теперь самая важная и наименее интересная часть нашей встречи
- Надо помнить самое важное: компьютер ОЧЕНЬ тупой – он реально тупее любого животного, даже таракана
- Его мощь тупо в количестве операций которые он может исполнить
- Ему надо очень четко ставить задачи – как двух-летнему ребенку



# ПАРАДИГМА ПРОГРАММИРОВАНИЯ

- Правильный подход
  - Что на входе?
  - Что на выходе?
  - Какие операции надо сделать и в какой последовательности чтобы из входных данных получить выход?
  - Заранее иметь набор входов с известными выходами чтобы проверить свою программу
- Внимание: Есть уникальные ситуации где один и тот-же вход может дать разные выходы, у нас с вами таких программ не будет но в природе это законно.
- Для тех кто сверх педантичны есть схемы UML которые как раз сделаны чтобы в них описать такие процессы и логику программы.

# ПАРАДИГМА ПРОГРАММИРОВАНИЯ

- Слово о скоростях: Python не самое быстрое что есть (точнее он один из самых медленных языков) – причины мы не будем обсуждать, но это важно понимать и особенно при работе с большими данными можно ожидать задержки.



# ВАШЕ ПЕРВОЕ ЗАДАНИЕ

- Найти бизнес задачу которую вы бы хотели решить и где вы думаете что можно применить Python
- Описать ее входы и выходы
- Описать последовательность действий которые вход превращают в выход