Скачиваем программу по ссылке [https://www.anaconda.com/download/](https://www.anaconda.com/download)

**Введение в основные типы данных**

**Шаг 1. Типы чисел в Python**

**ПОСТАНОВКА ЗАДАЧИ**

В первых нескольких шагах мы будем решать следующую задачу:

Проект монетизируется в двух основных регионах — Астана и Алматы. В обоих регионах известны количество пользователей и полученная выручка. Необходимо понять, в каком регионе монетизация более эффективна. Т. е. посчитать отношение выручки к количеству пользователей проекта в каждом регионе и сравнить эти значения (эту метрику называют ARPU — Average Revenue per User).

В Астане проект имеет 4 900 пользователей и выручку за отчетный период 63 000 тенге, в Алмате— 3 500 пользователей и 48 000 тенге выручки.

Мы познакомимся с основными типами данных в Рython и освоим операции форматирования результатов, оформив сравнение в следующем формате:

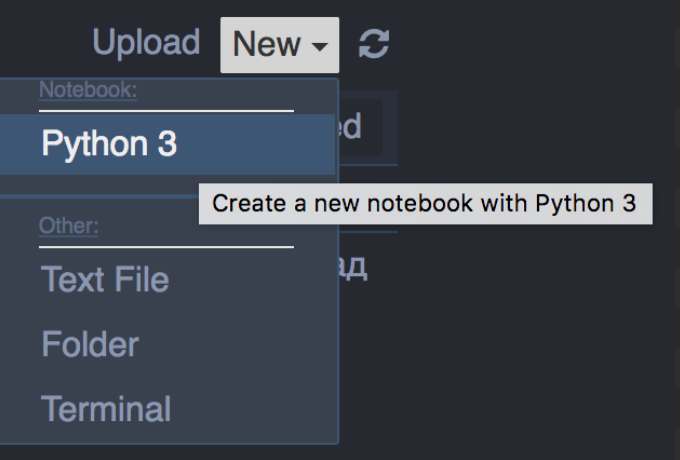
**Выручка на пользователя по регионам: Астана**— **12.86 тенге, Алматы**—**13.71 тенге.**

**Шаг 2. Вывод значений на экран**

**ВЫВОД ЗНАЧЕНИЙ НА ЭКРАН**

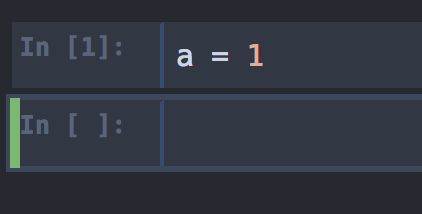
Для подсчета выручки с пользователя необходимо количество пользователей и выручку сохранить в переменных, чтобы в дальнейшем было легко совершать с ними разнообразные операции. В нашей задаче число пользователей и выручка являются целыми числами. Для них в питоне есть отдельный тип, который так и называется — integer.

Откройте в Jupyter notebook новый файл и выполните в нем следующий пример.



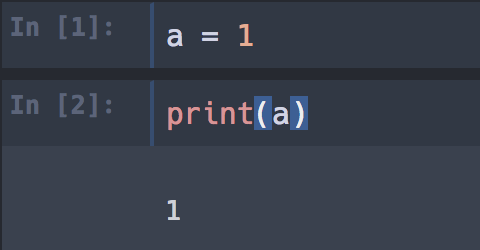
Пример (введите эту строку в Jupyter notebook и нажмите Shift + Enter):

a = **1**



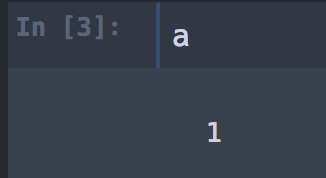
 В переменной a теперь хранится значение 1. Выведем значение на экран:

**print**(a)



**Совет**

В Jupyter notebook для вывода значения переменной на экран можно использовать просто название переменной a, print писать не обязательно:



**Шаг 3. Немного практики**

Вернитесь в свой Jupyter notebook и протестируйте основные математические действия в переменной a:

* + сложение a + 1

a + **1**

* + вычитание a - 20

a - **20**

* + умножение (a + 1) \* 2

(a + **1**) \* **2**

* + деление (a + 5) / 3

(a + **5**) / **3**

* + возведение в степень (a + 1) \*\* 5

(a + **1**) \*\* **5**

Упражнение

Какое значение должно было получиться в последнем примере (a + 1) \*\* 5? Напомним, изначально a = 1

**Подсказка (1 из 1):**значение a в нашем примере было равно 1. Осталось возвести (a + 1) в пятую степень

**Шаг 4. Оформление кода**

**ВЫБИРАЕМ НАЗВАНИЯ ПЕРЕМЕННЫХ**

В простых однострочных примерах название не имеет особого значения. При написании кода переменным лучше давать осмысленные названия. В дальнейшем это очень поможет вам и вашим коллегам понимать, как работает ваш код. Например, в нашей задаче есть 4 переменные: количество пользователей и выручка для Астаны и Алматы. Чтобы не путаться, какая переменная к какому региону относится, используйте более детальные названия этих переменных.

Пользователи Астаны (4 900 пользователей и 63 000 тенге выручки):

* users\_astana — количество пользователей из Астаны;
* revenue\_astana — выручка пользователей из Астаны.

Аналогично для Алматы (3 500 пользователей и 48 000 тенге выручки):

* users\_almaty — количество пользователей из Алматы;
* revenue\_almaty  — выручка пользователей из Алматы.

Набирать в коде длинные названия не проблема, т. к. в Jupyter notebook есть автодополнение с помощью табуляции. Набрав несколько первых букв названия переменной, через табуляцию можно сразу получить готовое название переменной. Пример записи переменной, как usersAstana, называется **CamelCase**. Также можно использовать обозначения переменных в виде users\_astana **(snake\_case).** Вы можете выбрать любой удобный для вас способ.

Упражнение

Задайте в параметрах users\_astana и revenue\_astana значения 4 900 и 63 000. А в параметрах users\_almapb и revenue\_almapb запишите значения 3 500 и 48 000.

Какова сумма выручки обоих регионов?

ВЫВОД РЕЗУЛЬТАТОВ

Для работы со словами и текстами существует строковый тип данных, или string. Под строкой обычно понимают набор символов в кавычках. Например, "привет". При этом можно использовать двойные кавычки ", либо одинарные '. Т. е. запись my\_string = "hello" и my\_string = 'hello' идентичны.

В Python строки можно соединять друг с другом, используя операцию "сложения" +. Давайте присвоим переменной project\_name значение в виде строки.

project\_name = 'Python для анализа данных'

Упражнение

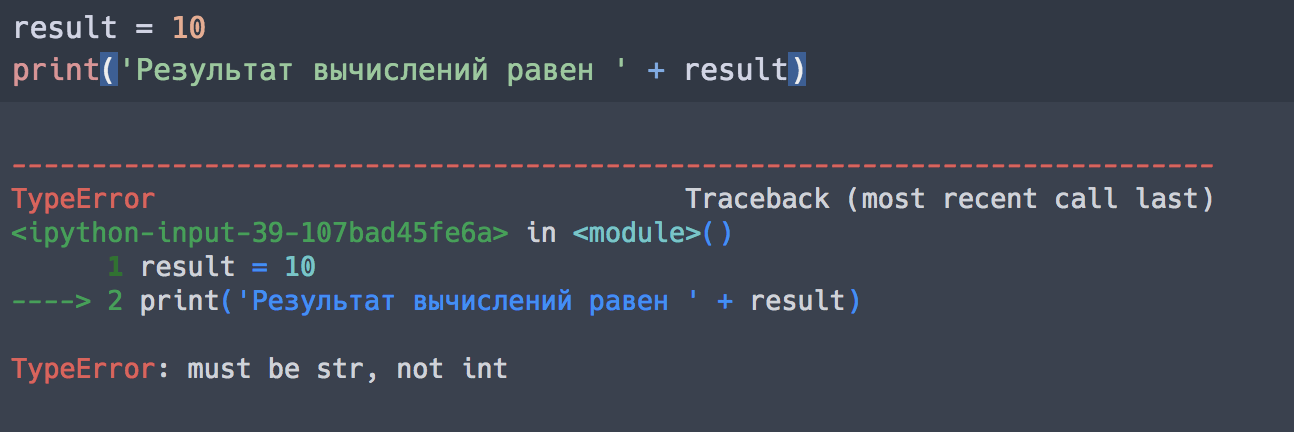
Что выведет система для следующей строки кода?

**print**('Название курса '+ project\_name)

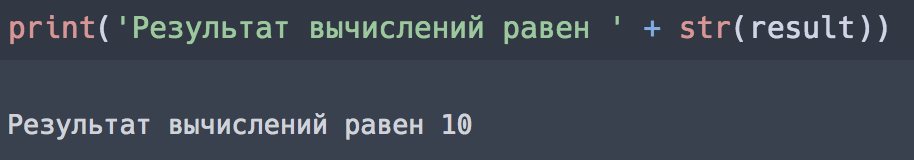
**Шаг 2. Вывод численных результатов**

**ВЫВОД ЧИСЛЕННЫХ РЕЗУЛЬТАТОВ**

Как отмечалось в прошлом шаге, операцию «сложения» можно применять к строкам. Однако, если мы попробуем снабдить комментариями численный результат, то получим ошибку несовместимости типов:



Т. е. мы пытаемся «сложить» переменные строкового и численного типов. Система не понимает, как их складывать, и выдает ошибку. Для того чтобы все-таки вывести численный результат с комментариями, преобразуем переменную result в строку с помощью функции str:



Теперь все корректно: мы «складываем» две строки и получаем желаемый комментарий к результату.

**УПРАЖНЕНИЕ (БЕЗ ОНЛАЙН-ПРОВЕРКИ)**

Выполните следующие действия:

* возьмите выручку и количество пользователей Астаны из прошлого блока (4 900 пользователей и 63 000 тенге)
* выведите на экран значение ARPU с комментарием в виде:

Значение ARPU пользователей Астаны равно 12.857142857142858

**Шаг 3. Формат вывода**

**ФОРМАТИРОВАНИЕ СТРОК**

Результат в прошлом упражнении выглядит весьма некрасиво. Для наглядного отображения результатов вычислений нам понадобится задавать формат вывода. Например, даже один и тот же результат можно вывести в виде дробного числа и в процентной записи. Для этого есть несколько методов. Мы будем использовать самый популярный и легко запоминающийся вариант. Возьмем данные нашего примера:

users\_astana = **4900**

revenue\_astana = **63000**

arpu\_astana = revenue\_astana / users\_astana

Давайте оформим результат наших простых вычислений с разными форматами. Попробуем написать фразу: 'Астана: пользователей 4900, выручка 63000 тенге, ARPU 12.86'

Напишем сначала слова, а вместо значений 4 900, 63 000 и 12.86 поставим две фигурные скобки (пока не будем выполнять этот код):

**print** ('Астана: пользователей {}, выручка {} тенге, ARPU {}')

В фигурных скобках будут стоять значения переменных users\_astana, revenue\_astana и arpu\_astana, которые мы передадим с помощью метода format:

**print**('Астана: пользователей {}, выручка {} тенге, ARPU {}' .format(users\_astana, revenue\_astana, arpu\_astana))

Соответственно, последовательность желаемых значений на месте фигурных скобок должна совпадать с последовательностью переменных, которые мы указываем в format. Такой формат очень удобен при выводе результатов: при написании кода нам не нужно переводить разные типы переменных в string. К тому же при большом числе переменных этот формат гораздо удобнее, чем наш прошлый вариант со "сложением" строк.

Теперь при выполнении данного кода мы получим строчку с нашими переменными: Какими будут результаты?

**Шаг 4. Задаем формат переменных**

Давайте теперь зададим нашему выводу более читаемый формат. Для чисел количество цифр после запятой задается следующим образом:

* в фигурных скобках пишем: 2f, где 2 — необходимое число знаков после запятой, f — указание на тип float;
* если нужно указать процентный формат, то вместо f ставим %.

 Запишем результат вычислений более наглядно:

**print**('Астана: пользователей {}, выручка {} тенге, ARPU {:.2f}'.format(users\_astana, revenue\_astana, arpu\_astana))

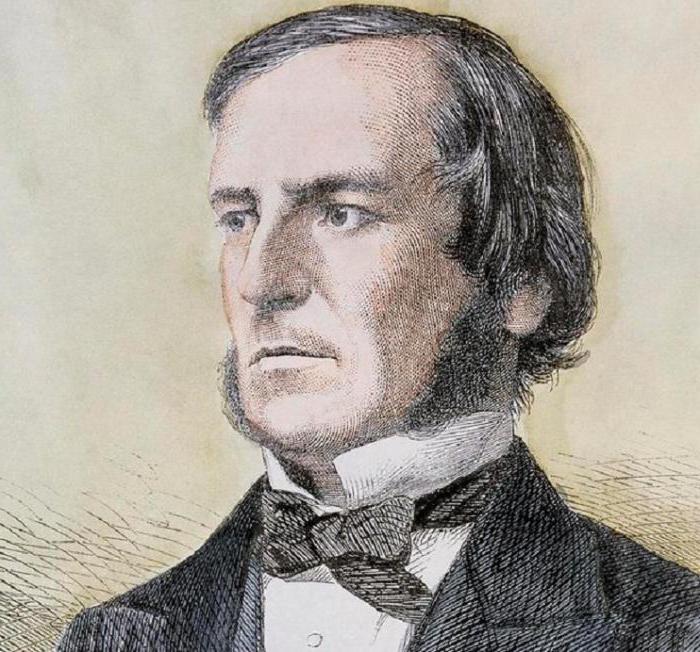
Упражнение

Выведите аналогичный результат для пользователей Алматы, округлив результат до четвертого знака после запятой.

Какое значение ARPU получится для Алматы при округлении до четвертого знака после запятой?

**Шаг 5. Простые проверки**

Нам осталось сравнить между собой эффективность монетизации пользователей Астаны и Алматы. Для этого можно использовать так называемый булевский тип данных (bool). Такое название происходит от фамилии математика Джорджа Буля.



Переменные этого типа принимают всего два значения: True (истина) или False (ложь).

Этот тип переменных удобно использовать в различных проверках на выполнение условий. Конечно, его можно заменить любым другим типом. Например, использовать тип int со значениями 0 и 1 или строковый с аналогичными значениями 'True' и 'False'. Но использование булевого типа переменных может существенно сократить количество кода, сделав его более наглядным.

Например, для проверки неравенства значений переменных a и b можно написать:

a = **1**

b = **2**

Что выдаст система, если написать a==b?

Сравниваем ARPU

Ранее мы посчитали выручку на пользователя для Астаны (arpu\_astana) и Алматы (arpu\_almaty). Как теперь проверить, что значение arpu\_almaty больше, чем arpu\_astana? В качестве результата скрипт должен выдавать True или False.

* float(arpu\_almaty) / float(arpu\_astana)
* print(arpu\_almaty - arpu\_astana)
* arpu\_almaty > arpu\_astana