Тема «Основные операции»
Самостоятельная работа
Основные операции в Python. Ввод и вывод значений.

Выполните указанные ниже задания в виде отдельного файла Jupyter notebook, сохраните его (в имени файла укажите вашу фамилию и инициалы, а также номер модуля и название данной темы) и прикрепите к данному заданию в системе.

Решим следующую задачу:

У первого баннера количество показов составило 1 миллион штук, кликов - 983 штуки. У второго - 100 тысяч показов и всего 4 клика.

Необходимо сделать скрипт, выполняющий две операции:

1. Выводит на экран отношение кликов по баннерам к их показам (т. е. CTR) в следующем формате:

CTR первого баннера равен 0.001, второго - 0.004%

2. Проверяет превышает ли средний CTR обеих кампаний определенное значение.

Для справки

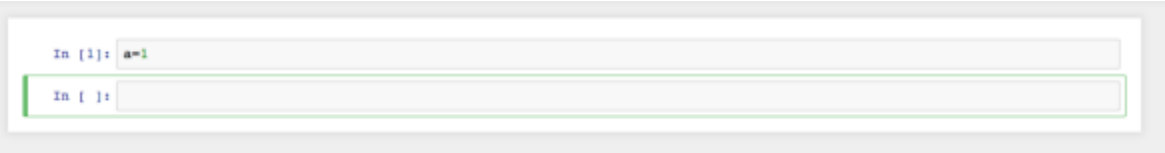
CTR - Click-Through Rate

Для подсчета CTR необходимо количество показов и кликов сохранить в переменных. Чтобы в дальнейшем было легко совершать с ними разнообразные операции. В нашей задаче показы и клики 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

Протестируйте основные математические действия с переменной a:

- сложение a + 1
- вычитание a - 20
- умножение (a + 1) * 2
- деление (a + 5) / 3
- возведение в степень (a + 1) ** 5
- деление нацело a//3
- остаток от деления a%3

При написании кода переменным лучше давать осмысленные названия. Это очень поможет вам и вашим коллегам в дальнейшем понимать, как работает ваша программа. В простых однострочных примерах это не имеет особого значения. Однако в дальнейшем лучше обозначать переменные в соответствии с их смыслом. Например, в нашем упражнении есть 4 переменные: показы и клики для двух баннеров. Чтобы в дальнейшем не путаться какая переменная относится, к какому баннеру давайте использовать более детальные названия этих переменных:

Первый баннер (1 миллион показов и 983 клика):

- banner1_shows - количество показов первого баннера
- banner1_clicks - количество кликов по первому баннеру

Второй баннер (100 тысяч показов и 4 клика):

- banner2_shows - количество показов второго баннера
- banner2_clicks - количество кликов по второму баннеру

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 queriesList называется CamelCase. Также можно использовать обозначения переменных в виде queries_list (snake_case). Вы можете выбрать любой способ, удобный вам.

Задайте в параметрах banner1_shows и banner1_clicks значения 1 миллион и 983. В параметрах banner2_shows и banner2_clicks запишите значения 100 тысяч и 4.

Посчитайте во сколько раз CTR первого баннера больше, чем CTR второго. Ответ округлите до ближайшего целого числа

Значения CTR у нас получались дробными. Для таких чисел в питоне есть отдельный класс float. Естественно, можно было использовать тип float и для целых чисел. Но float требует больше ресурсов компьютера, ведь он содержит гораздо больше информации о числе. В наших небольших примерах эта разница незаметна. Однако при написании скриптов со сложными вычислениями для целых чисел лучше использовать тип int.

Проверьте какие типы переменных мы получали в наших вычислениях:

banner1_shows = 1000000

banner1_clicks = 983

ctr = banner1_clicks / banner1_shows

Определяем типы переменных

Вбейте в новую ячейку type(banner1_shows)

Какой ответ выводит система?

Тип второй переменной?

Тип переменной ctr?

Выведите на экран тип, который получается при вычислении квадратного корня из 2. Квадратный корень идентичен возведению в степень 0.5.

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 myString = "hello" и myString = 'hello' идентичны.

В Python строки можно соединять друг с другом, используя операцию "сложения" +.

Давайте присвоим переменной Name значение в виде строки.

Name = 'Python my love'

Что выведет система для print(Что такое Питон? ' + projectName)

Выполните следующие действия:

- возьмите значение показов и кликов первого баннера из прошлого упражнения (1000000 показов и 983 клика)
- выведите на экран значение CTR этого баннера с комментарием в виде: Значение CTR первого баннера равно 0.000983

Результат в прошлом упражнении выглядит «некрасиво». Для наглядного отображения результатов вычислений 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. В последних версиях питона рекомендуется использовать следующую нотацию. Возьмем наш пример:

```
banner1_shows = 1000000
banner1_clicks = 983
ctr = banner1_clicks / banner1_shows
print( ctr )
```

Давайте оформим результат наших простых вычислений с разными форматами. Попробуем написать фразу: 'Результаты первой кампании: показов 1000000, кликов 983'.

Напишем сначала слова, а вместо значений 1 миллион и 983 поставим две фигурные скобки (пока не будем выполнять этот код):

```
print( 'Результаты первой кампании: показов {}, кликов {}, CTR {}' )
```

В фигурных скобках будут стоять значения переменных a и b, которые мы передадим с помощью метода format:

```
print( 'Результаты первой кампании: показов {}, кликов {}, CTR {}'.format( banner1_shows, banner1_clicks, ctr ) )
```

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:

Результаты первой кампании: показов 1000000, кликов 983, CTR 0.000983

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

в фигурных скобках пишем :.2f, где 2 - необходимо число знаков после запятой, f - указание на тип float
для указания процентного формата вместо f ставим %

Запишем результат вычислений в более наглядном виде:

```
print( 'Результаты первой кампании: показов {}, кликов {:.1f}, CTR {:.3%}'.format( banner1_shows, banner1_clicks, ctr ) )
```

Результаты первой кампании: показов 1000000, кликов 983.0, CTR 0.098%

Упражнение
Вам даны значения показов и кликов двух баннеров:

```
banner1_shows = 1000000
banner1_clicks = 983
banner2_shows = 100000
```

banner2_clicks = 4

Необходимо вывести результаты этих кампаний в следующем виде:

Результаты кампаний: CTR первой 0.0983%, второй - 0.0040%

Какое выражение нужно написать в квадратных скобках, чтобы получить такую запись? Укажите формат вывода CTR, заключенного в фигурные скобки (например, так как мы указывали формат кликов в последнем примере: {:.1f}).

Для проверки значений CTR во втором пункте нашего упражнения 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 равенства значений переменных a и b можно написать:

a = 1
b = 2

Что выдаст система, если написать a==b?
«a<=b?»

Сравниваем CTR
В первой части упражнения мы подсчитали значения CTR первого и второго баннера (ctr1 и ctr2). Как теперь проверить, что значение ctr1 больше, чем ctr2? В качестве результата скрипт должен выдавать True или False.