

# Методические указания

## Урок 2.1 Типы данных

### Задачи урока:

- Познакомить с основными типами данных в python;
- Разобрать работу с типами на задачах.

## 0. Подготовка к уроку

До урока преподавателю необходимо:

- 1) Прочитать методичку до конца.

## 1. Основные типы данных в языке

**Учитель:** Сегодня мы с поговорим о основных типах данных, которые присутствуют в языке программирования Python. Каждый тип в дальнейшем мы разберем более подробно и закрепим работу с ними на практике.

Для начала давайте посмотрим какие типы данных нам доступны при установке Python.

1. integer - целое число
2. float - дробное число
3. decimal - число с фиксированной точностью
4. string - строка
5. boolean - булев тип. Имеет всего два варианта: правда(True) и ложь(False)
6. list - список
7. tuple - кортеж
8. set - множество
9. frozenset - неизменяемое множество
10. dict - словарь

Начнем с одного из самых простых типов - integer.

**Учитель:** Какие основные операции мы знаем с числами? (*задает вопрос ученикам*)

Основные операции с числами:

1. + сложение
2. - вычитание
3. \* умножение
4. / обычное деление
5. // целочисленное деление(без остатка)
6. \*\* возведение в степень
7. % остаток от деления

Разберем каждую операцию на простых примерах

```
print(5 + 5)
```

Данная операция выведет результат сложения двух чисел. Не стоит забывать, что мы не можем складывать, как пример число и строку

```
print(5 + "5")    # выведет ошибку
```

Аналогичным образом работает вычитание, умножение и деление. Стоит обратить внимание, что в Python два типа деления. Обычное и целочисленное. В чем же между ними разница? Так как у нас динамическая типизация (Python определяет сам, к какому типу относится переменная, в зависимости от значения, которое она хранит), то и деление у нас двух типов. Обычное деление у нас позволяет получить дробное число, а целочисленное деление округляет результат, до ближайшего целого числа в меньшую сторону.

```
print(5 - 5)
print(5 * 5)
print(5 / 2)    # выведет 2.5
print(5 // 2)   # выведет 2
print(5 ** 2)   # возводит число 5 в степень 2
```

Очень интересно работает операция нахождения остатка от деления. Она позволяет нам найти остаток, от деления числа нацело. Подобная операция незаменима, например, когда мы хотим узнать четное или нечетное число, узнать на какую цифру оканчивается число и т.п.

Разберем несколько примеров. Например попросим пользователя ввести число и программа, должна будет вывести четное оно или нет. Так как все, что пользователь вводит

через `input()`, будет являться строкой, то для математических вычислений нам потребуется преобразовать строку в число с помощью функции `int()`, которая преобразует переданную в нее строку в целочисленное значение. Также существует конвертация в строку с помощью функции `str()`

```
user_number = int(input('Введите число'))
if user_number % 2 == 0:
    print('Число четное')
```

В данном примере мы используем новую для нас конструкцию - условия(`if`). Условия позволяют нам производить какие то действия если условие верно. `if` переводится как если. Данный пример можно прочитать как если `user_number` делится на 2 без остатка, то выведи нам сообщение в консоль.

Рассмотрим более подробнее как работает остаток от деления в данном примере.

Предположим пользователь ввел число 5. 5 не делится нацело на число 2. Если мы разделим нацело 5 на 2, то получим в остатке 1. Если же у нас например было число 6, то оно бы разделилось без остатка и остаток был бы равен 0. Тем самым, как пример все четные числа при делении на 2 имеют остаток 0, а нечетные остаток 1.

```
user_number = int(input('Введите число'))
if user_number % 2 == 0:
    print('Число четное')
if user_number % 2 == 1:
    print('Число нечетное')
```

Рассмотрим более интересный вариант. Предположим пользователь вводит какое то число, и мы хотим узнать его последнюю цифру. *Как вы думаете как можно это сделать?(спрашивает у учеников)*

Правильным ответом будет найти остаток от деления заданного числа на 10. Тем самым в остатке мы получим именно разряд единиц для данного числа.

```
user_number = int(input('Введите число'))
print(user_number % 10)
```

Рассмотрим на примере числа 2345. Все разряды числа, кроме единиц поделятся нацело на 10. 5 не делится на 10, а если делимое меньше делителя, то остатком будет являться делимое. Тем самым данный код выведет нам число 5.

## 2. Решение задач

### Задача 1.

Разложить любое 4 значное число на отдельные цифры. Пример для числа 1234, должен вывести: тысячи - 1, сотни - 2, десятки - 3, единицы - 4.

### Решение

```
x = 1234
print(x // 1000)
print(x // 100 % 10)
print(x // 10 % 10)
print(x % 10)
```

### Дополнительно

Если на уроке остается время, то ученикам можно предложить начать прорешивать домашнее задание.

### Домашняя работа

В конце урока можно подвести итоги и дать домашнюю работу: