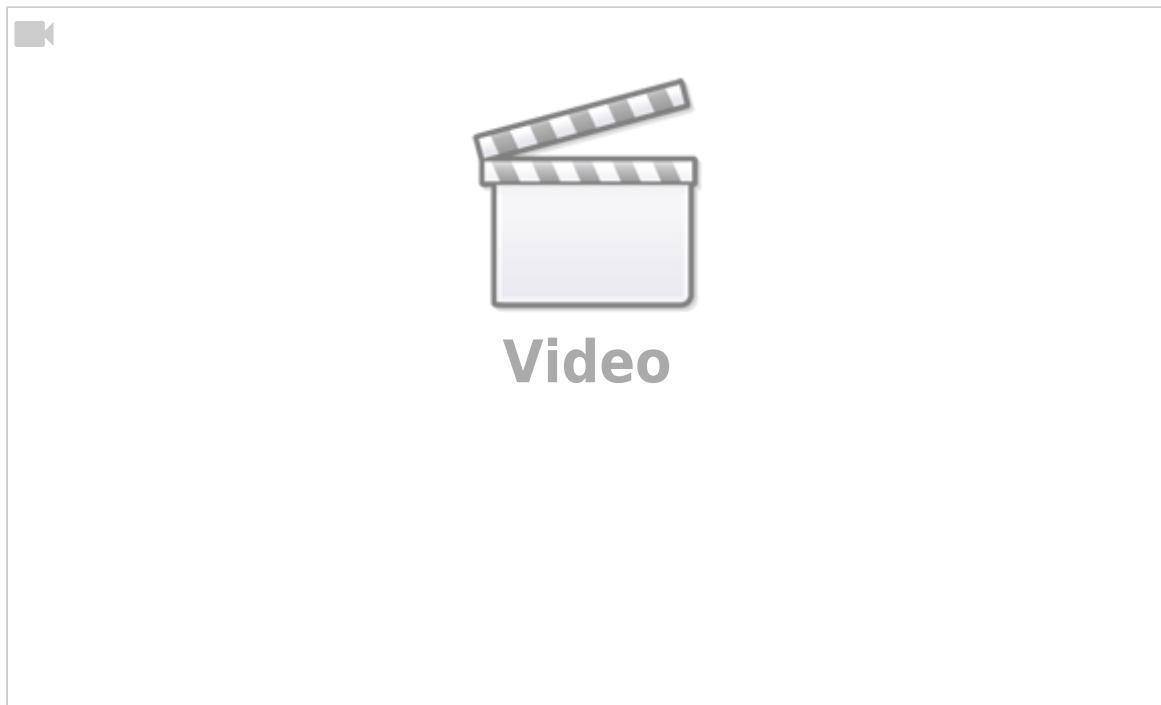


Урок 3. Типы данных



Типы данных в языках программирования - это способ классификации и организации информации, которую компьютер может обрабатывать. Они определяют, какие типы значений и операции могут быть применены к этим значениям.

Давайте рассмотрим аналогию с реальной жизнью. Представьте, что вы храните свои вещи в различных ящиках. Каждый ящик имеет свою метку, которая описывает тип содержимого внутри. Некоторые ящики могут содержать одежду, другие - книги, еще другие - еду и т.д. В данном случае, метка на ящике - это тип данных.



Типы данных в программировании работают аналогичным образом. Они определяют, как компьютер будет хранить и обрабатывать информацию. Например, в языке программирования могут быть типы данных, такие как целые числа (например, 5), десятичные числа (например, 3.14), строки (например, «Привет!») и булевы значения (например, true или false).

Как и в реальной жизни, разные типы данных имеют свои особенности и правила, которые определяют, какие операции могут быть выполнены с этими данными. Например, еду нужно хранить в определенных условиях (в холоде), можно взять два небольших контейнера с помидорами и огурцами и объединить их в один, назвав «овощи». Подобно и с языками программирования, например, вы можете складывать числа, объединять строки, сравнивать булевы значения и т.д.

Понимание типов данных в программировании важно, поскольку это позволяет разработчику правильно объявлять переменные, выполнять операции и обрабатывать данные в программе.

Язык python имеет динамическую типизацию. То есть не нужно каждый раз указывать тип значения, он будет выведен автоматически. Например, когда в прошлом уроке мы писали инструкцию `print(5 + 2)`, то подразумевалось что мы складываем два числа и хотим в результате видеть тоже число. И увидели, что в результате мы получили именно число - 7.

Тем не менее все данные делятся на типы, чтобы было понятно что с ними делать в результате различных операций. Если мы будем уже складывать не числа, а строки, то нам бы хотелось видеть в результате их сложения какое-то слово или предложение. Например, сложив строки "пи" + "тон", мы явно хотели бы увидеть "питон" в результате, а не что-то другое ещё. Также в зависимости от типа данных определённые операции выполнять можно, а другие нельзя. Например, вычитание двух строк "пи" - "тон" бессмысленно, поэтому операции вычитания над данными типа строка не существует.

Итак, первый тип данных, это **число**. Причем целое число - **int** и вещественное (в более простом понимании - дробное число) - **float**, это два разных типа. Но арифметические операции над ними выполняются примерно одинаково.

Второй тип - это **строка (string)**. Строкой является даже один символ, например "я" - это тоже строка, просто состоит из одного знака (символа). Строки используются для хранения и представления текстовой информации. Для того чтобы указать что это строка, а не инструкция, следует поместить строку в кавычки (одинарные или двойные - мы будем использовать двойные, так как это привычнее для русскоговорящих людей).

Следующий тип - это **Булево (bool)**. Он может иметь одно из двух значений *True* или *False* (истина или ложь). Данный тип очень удобен когда, например, нам надо узнать состояние светодиода - ведь он может находиться только в двух состояниях: включен или выключен. То есть узнать светодиод включен - это правда или ложь? True или False?

Помимо этих основных типов данных, существуют и другие, которые мы рассмотрим уже позднее, по мере прохождения нашего курса.

Запомнить:

- Любые данные имеют свой тип, который определяет какие операции над этими данными выполнять можно, а какие нет.

- Число - это тип данных для хранения количественных данных, который позволяет выполнение математических операций.
- Строка - это тип данных для хранения и представления текстовой информации.
- Булево - это тип данных, который может принимать одно из двух значений True или False (истина или ложь). Он нужен для логических вычислений.
- Существуют другие типы, а также можно создавать и свои типы данных.

[Предыдущий урок](#)

[Следующий урок](#)

From:
<https://know.gikkon.ru/> -

Permanent link:
https://know.gikkon.ru/main/gikkon_start/p1_I3

Last update: **2023/08/03 07:36**

