Каждый *объект данных* (переменная или константа) имеет тип, который описывает, как сохранить его в памяти, какие операции могут быть применены к этому объекту и как их вычислить.

Аналогией типов в реальном мире могут быть биологические виды или любой другой абстрактный атрибут, общий для конкретных объектов. Все собаки, которых вы видели, имеют тип *dog*, но каждый из них является индивидуальным объектом. Думая о *собаке* как о типе, вы можете предположить, что некоторые операции доступны, например, собака может лаять.

В этом разделе мы рассмотрим лишь несколько простейших типов данных, которые обычно используются в практике программирования.

**Строки**

Всякий раз, когда вы хотите работать с какой-либо текстовой информацией в своей программе, вам придется работать со **строками**. Вызывается строковый тип str. Строки чрезвычайно распространены и полезны в Python. Строковые литералы могут быть заключены в одинарные или двойные кавычки.

* Примеры строк в двойных кавычках:

print("") # empty string

print("string") # one word

print("Hello, world!") # a sentence

* Примеры строк в одинарных кавычках:

print('a') # single character

print('1234') # a sequence of digits

print('Bonjour, le monde!') # a sentence

В реальной программе строка может представлять электронную почту человека или организации.

print('hello@hyperskill.org') # printing an email

Как вы можете видеть, строки очень просты в использовании!

**Числовые типы**

Числа - это самое важное для любого программиста. Вряд ли существует какая-либо серьезная программа, которую вы могли бы написать, не используя числа, поэтому давайте обсудим некоторые основные **числовые типы**:

* int**(целые числа со знаком)**. Называемые целыми числами или целыми числами, они представляют собой целые числа (положительные, отрицательные или нулевые), не имеющие десятичной точки;
* float**(числа с плавающей запятой)**. Называемые числами с плавающей запятой, они представляют действительные числа и имеют десятичную точку.

Вы можете начать работать с числом, просто распечатав его.

print(11) # prints 11

print(11.0) # prints 11.0

Хотя 11 и 11.0 имеют одинаковое число, первое является целым числом, а второе - с плавающей точкой. Самый простой способ отличить их заключается в том, что числа с плавающей запятой имеют **десятичную точку**, а целые числа - нет. Обратите внимание!

Вы также можете использовать отрицательные числа, а также нули:

print(0) # prints 0

print(-5) # prints -5

print(-1.03) # prints -1.03

Целые числа можно использовать для подсчета в реальном мире, в то время как числа с плавающей запятой являются хорошим выбором для статистических и научных расчетов.

**Типы печати**

У нас также есть способ наглядно продемонстрировать типы различных объектов с помощью type() функции, которая является частью Python.

print(type('hello')) # <class 'str'>

print(type("world")) # <class 'str'>

print(type(100)) # <class 'int'>

print(type(-50)) # <class 'int'>

print(type(3.14)) # <class 'float'>

print(type(-0.5)) # <class 'float'>

Как вы можете видеть из приведенных выше примеров, type() функция указывает тип данных передаваемого значения после слова *класс*.

**Краткие сведения**

Мы надеемся, что теперь у вас есть некоторое представление о концепции типов данных. Вы должны помнить простейшие типы, называемые str, int и float, и как писать их литералы. В следующих разделах мы изучим специфические особенности каждого из этих типов. Если вам нужно знать тип объекта, просто распечатайте его с помощью type() функции.