Переменные и типы данных

№ урока:	2	**Курс:**	Python Starter

Средства обучения:	Python 3.4; интегрированная среда разработки (PyCharm 4 (рекомендуется) или Microsoft Visual Studio 2013 + Python Tools for Visual Studio)

Обзор, цель и назначение урока

После завершения урока обучающиеся будут иметь представление о переменных и константах, арифметических и логических операциях, смогут форматировать и выводить информацию на экран.

Изучив материал данного занятия, учащийся сможет:

- Хранить значения в переменных и работать с ними
- Вычислять в своих программах значения арифметических и логических выражений
- Пользоваться форматированным выводом

Содержание урока

1. Переменные и константы
2. Арифметические операции
3. Операции сравнения
4. Логические операции
5. Форматирование и вывод строк на экран

Резюме

- Переменная в императивном программировании — поименованная, либо адресуемая иным способом область памяти, адрес которой можно использовать для осуществления доступа к данным. Данные, находящиеся в переменной (то есть по данному адресу памяти), называются значением этой переменной.
- В других парадигмах программирования, например, в функциональной и логической, понятие переменной оказывается несколько иным. В таких языках переменная определяется как имя, с которым может быть связано значение, или даже как место (location) для хранения значения.
- Тип данных (тип) — множество значений и операций на этих значениях.
- Тип определяет возможные значения и их смысл, операции, а также способы хранения значений типа. Изучается теорией типов. Неотъемлемой частью большинства языков программирования являются системы типов, использующие типы для обеспечения той или иной степени типобезопасности.
- Операция назначения типа информационным сущностям называется типизацией. Назначение и проверка согласования типов может осуществляться заранее (статическая типизация), непосредственно при использовании (динамическая типизация) или совмещать оба метода. Типы могут назначаться «раз и навсегда» (сильная типизация) или позволять себя изменять (слабая типизация).
- Статическая типизация — приём, широко используемый в языках программирования, при котором переменная, параметр подпрограммы, возвращаемое значение функции связывается с типом в момент объявления и тип не может быть изменён позже (переменная или параметр будут принимать, а функция — возвращать значения только этого типа). Примеры статически типизированных языков — C#, Java, C++, Pascal, Haskell, Scala.

CyberBionic systematics

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Некоторые статически типизированные языки также имеют возможность использовать динамическую типизацию при помощи специальных подсистем.
- Динамическая типизация — приём, широко используемый в языках программирования и языках спецификации, при котором переменная связывается с типом в момент присваивания значения, а не в момент объявления переменной. Таким образом, в различных участках программы одна и та же переменная может принимать значения разных типов. Примеры языков с динамической типизацией — Smalltalk, Python, Ruby, PHP, Perl, JavaScript, Lisp, Erlang.
- По одной из классификаций, языки программирования неформально делятся на сильно (строго) и слабо типизированные (англ. strongly and weakly typed), то есть обладающие сильной или слабой системой типов. Эти термины не являются однозначно трактуемыми, и чаще всего используются для указания на достоинства и недостатки конкретного языка. Существуют более конкретные понятия, которые и приводят к называнию тех или иных систем типов «сильными» или «слабыми».
- «Сильной» и «слабой» типизацией называется продукт множества решений, принятых при разработке языка. Более точно языки характеризуются наличием или отсутствием безопасности согласования типов и безопасности доступа к памяти, а также характерным временем осуществления такого контроля (в статике или в динамике).
- Система типов называется сильной, если она исключает возможность возникновения непроконтролированных ошибок времени выполнения, связанных с согласованием типов.
- Примеры языков с сильной типизацией – Haskell, Scala, Java. Примеры языков со слабой типизацией – C, JavaScript, PHP.
- Python – язык с сильной динамической типизацией.
- Константа в программировании — способ адресации данных, изменение которых рассматриваемой программой не предполагается или запрещается. Использование именованных констант — приём, повышающий надёжность и безошибочность программ, позволяя избегать использования «магических чисел».
- В Python нет отдельной синтаксической конструкции для объявления констант. Принято называть их идентификаторами, написанными заглавными буквами (MY_CONSTANT), тогда как переменные в Python объявляются с именами из строчных букв (my_variable).
- Идентификатор – это имя, по которому можно обращаться к какому-либо объекту. Он может состоять из больших и маленьких букв (любого языка, символы которого входят в кодовую таблицу UTF-8, но использовать языки, отличные от английского, является очень плохим тоном), цифр, знаков подчёркивания, не должен начинаться с цифры и не может совпадать с зарезервированными ключевыми словами:

```
False      class      finally    is         return
None       continue   for        lambda     try
True       def        from       nonlocal   while
and        del        global     not        with
as         elif       if         or         yield
assert     else       import     pass
break      except     in         raise
```

- Python чувствителен к регистру: identifier и Identifier – это два разных имени.
- None – специальное значение типа NoneType, которое используется, чтобы указать отсутствие значения.
- int – тип данных, который используется для целых чисел. Примеры: 3, 0, -1, 256123462346523235252 (десятичная система счисления), 0b11101 (двоичная система счисления), 0o730 (восьмеричная система счисления), 0x7de (шестнадцатеричная система счисления).
- bool – логические значения. Есть два допустимых значения: True (истина) и False (ложь).
- float – действительные числа двойной точности. Примеры: 0.3215235, 125256.73, -2.3e8 (экспоненциальная форма записи).
- complex – комплексные числа. Примеры: 2 + 3j, 0.7 - 0.1j
- str – строки.
- Кроме вышеперечисленных типов данных (а также функций и списков, которые будут рассмотрены в этом курсе позже), есть ещё огромное количество других типов данных, которые будут рассмотрены в курсах Python Essential и Python Advanced, а также

CyberBionic systematics

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

имеется возможность создавать свои типы данных. Любая сущность в Python является объектом с определённым типом.

- Переменные в Python – это ссылки, имена, которые привязываются к объектам. Можно привязать несколько разных имён к одному объекту, и тогда при изменении объекта по одному имени, все эти изменения будут доступны и по остальным.
- Объекты бывают изменяемые (mutable) и неизменяемые (immutable). Объекты рассмотренных выше типов являются неизменяемыми.
- Операции с числами:

x + y	сумма
x – y	разница
x * y	произведение
x / y	частное
x // y	операция целочисленного деления
x % y	остаток от деления
-x	число, противоположное x
+x	x
abs(x)	модуль числа x
int(x)	преобразовать x в целое число
float(x)	преобразовать x в комплексное число
complex(re, im)	создать комплексное число re + im*i
c.conjugate()	число, сопряжённое комплексному числу c
x ** y pow(x, y)	x в степени y
round(x) round(x, n)	округлить действительное число x (до n цифр после запятой, если n указано)

- У арифметических операций есть приоритеты (такие же, как и в математике). Чтобы изменить порядок вычислений, следует использовать круглые скобки.
- Арифметические операции, первым операндом которых является та же переменная, что и результат, можно записывать в сокращённой форме, например, x += y вместо x = x + y.
- Если импортировать модуль math:

```
import math
```

то можно также использовать ряд его функций и констант. Среди них:

math.trunc(x)	отбросить дробную часть действительного числа x, возвращает целое число
math.floor(x)	наибольшее целое число (целое в математическом смысле, а не как тип данных), которое не превосходит данное вещественное число
math.ceil(x)	наименьшее целое число, большее или равное данному вещественному
math.pi, math.e	константы π, e

math.sin,
math.cos и т.д.
	математические функции; полный их список можно посмотреть в документации или набрав в консоли интерпретатора:
	`import math`
	`dir(math)`

- Логические операции:

x or y	если x – ложь, то y, иначе x
x and y	если x – ложь, то x, иначе y
not x	если x – ложь, то True, иначе False

- Операции сравнения:

x < y	x строго меньше y
x > y	x строго больше y

CyberBionic systematics

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

x <= y	x меньше или равен y
x >= y	x больше или равен y
x == y	x равен y
x != y	x не равен y
x is y	x и y – это один и тот же объект
x is not y	x и y не являются одним и тем же объектом в памяти

- Можно также использовать двойные сравнения, например, -2 <= x < 3, a < b < c.
- Некоторые операции со строками:

s1 + s2	конкатенация (объединение) строк
s % x s % (x_1, x_2, …, x_n)	форматирование строки в стиле C s – форматная строка, x_1…x_n – значения https://docs.python.org/3/library/stdtypes.html#printf-style-string-formatting
s.format(args)	форматирование строки в стиле C# https://docs.python.org/3/library/stdtypes.html#str.format

- Для вывода значений на экран служит функция print:

  `print`(**objects*, *sep=' '*, *end='\n'*, *file=sys.stdout*, *flush=False*)

  где
  - objects – это объекты, которые необходимо вывести (символ * в объявлении функции значит, что при вызове функции они все передаются как отдельные параметры функции, а интерпретатор Python объединяет их в список и передаёт функции как один аргумент)
  - sep – разделитель, который вставляется между выводом отдельных объектов (по умолчанию пробел)
  - end – строка, которую необходимо вывести после всех объектов (по умолчанию – символ новой строки \n)
  - file – файл, в который необходимо вывести данные (по умолчанию – стандартный поток вывода, который связан с консолью; в рамках этого курса мы не будем использовать этот параметр, так как файлы рассматриваются в курсе Python Essential
  - flush – нужно ли сразу после вывода сбросить содержимое буфера в файл. Если вы выводите информацию на одной строке через продолжительные промежутки времени, и она не появляется на экране, пока вы не выведете символ новой строки, добавьте параметр flush=True

  Ни один из этих параметров не является обязательным.

- Для ввода данных с клавиатуры можно использовать функцию input:

  input(prompt)
  или
  input()

  Если параметр prompt задан, то он выводится, как поясняющий текст, приглашение ко вводу.

  Функция приостанавливает выполнение программы, пока пользователь не введёт строку текста, считывает её и возвращает.

  Обратите внимание, что она возвращает именно строку, то есть значение типа str, поэтому если необходимо ввести число, то нужно воспользоваться одной из функций, которые были рассмотрены выше, чтобы сконструировать число из его текстового представления.

Закрепление материала

- Что такое переменная?
- Что такое тип данных?
- В чём разница между статической и динамической типизацией?

CyberBionic systematics

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- В чём разница между слабой и сильной (строгой) типизацией?
- Охарактеризуйте типизацию в языке Python.
- Что такое идентификатор?
- Какие основные числовые типы данных есть в Python?
- Что такое None?
- Какие два способа форматирования строк есть в Python?
- Какие аргументы может принимать функция print?
- Каким будет результат выражения `3 != 4 and not ("test" != "test" or "Python" == "Python")`?

Дополнительное задание

Задание
Напишите программу, которая запрашивает у пользователя радиус круга и выводит его площадь. Формула площади круга S= πr^2.

Самостоятельная деятельность учащегося

Задание 1
Напишите программу, которая спрашивает у пользователя два слова и выводит их разделёнными запятой.

Задание 2
Напишите программу, которая запрашивает три целых числа a, b и x и выводит True, если x лежит между a и b, иначе – False.

Задание 3
Напишите программу, которая решает квадратное уравнение $ax^2 + bx + c = 0$ по формулам $x_{1,2} = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$. Значения a, b и c вводятся с клавиатуры. Для извлечения корня используйте оператор возведения в степень, а не функцию math.sqrt, чтобы получить комплексные числа в случае, если подкоренное выражение отрицательно.

Рекомендуемые ресурсы

Документация по Python о стандартных типах данных и операциях с ними
https://docs.python.org/3/reference/datamodel.html#the-standard-type-hierarchy
https://docs.python.org/3/library/stdtypes.html

Документация Python о операциях над строками и их форматировании
https://docs.python.org/3/library/stdtypes.html#text-sequence-type-str
https://docs.python.org/3/library/stdtypes.html#printf-style-string-formatting
https://docs.python.org/3/library/stdtypes.html#str.format
https://docs.python.org/3/library/string.html#formatstrings

Статьи в Википедии о ключевых понятиях, рассмотренных на этом уроке
https://ru.wikipedia.org/wiki/Переменная_(программирование)
https://ru.wikipedia.org/wiki/Константа_(программирование)
https://ru.wikipedia.org/wiki/Тип_данных
https://ru.wikipedia.org/wiki/Статическая_типизация
https://ru.wikipedia.org/wiki/Динамическая_типизация
https://ru.wikipedia.org/wiki/Сильная_и_слабая_типизация
https://ru.wikipedia.org/wiki/Целое_число
https://ru.wikipedia.org/wiki/Вещественное_число
https://ru.wikipedia.org/wiki/Комплексное_число
https://ru.wikipedia.org/wiki/Логический_тип

CyberBionic systematics
CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com