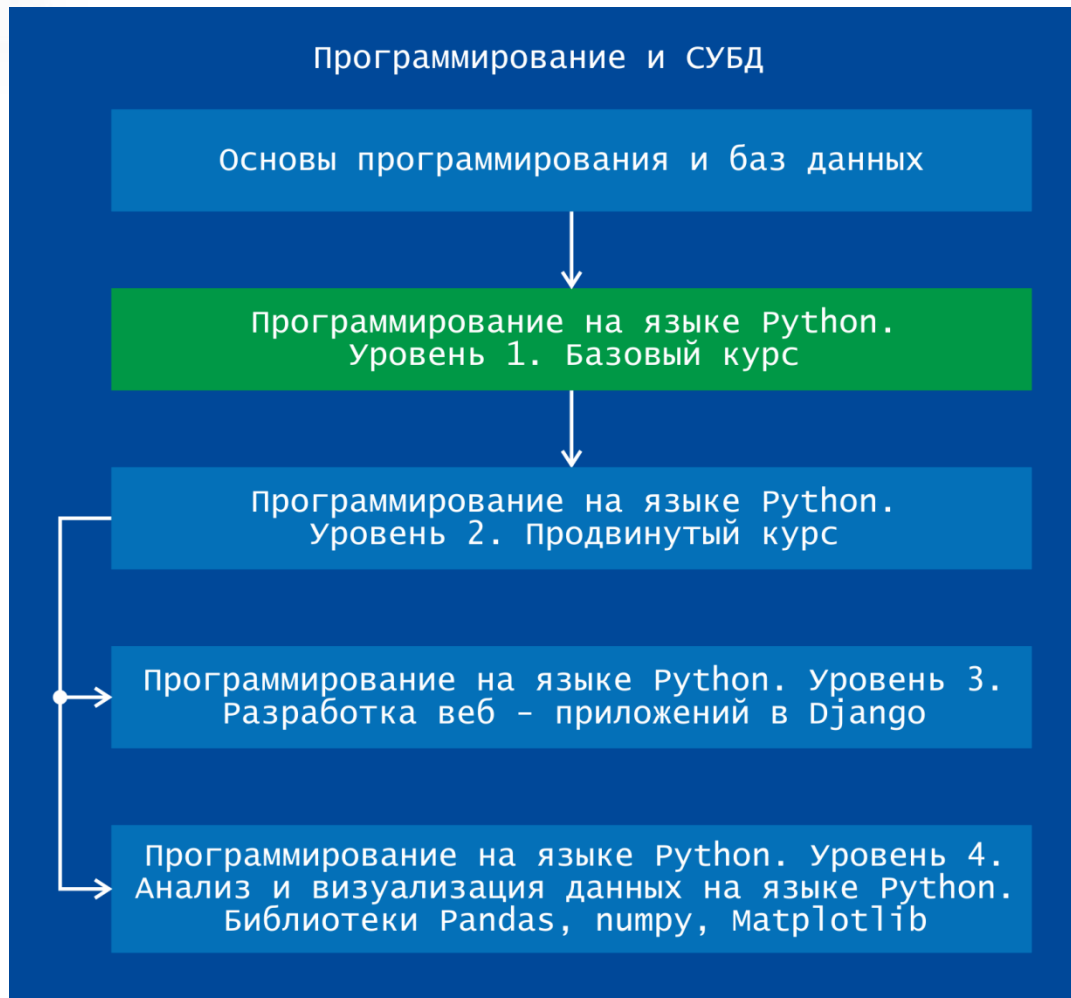


Программирование на языке Python. Уровень 1

О курсе



Модуль 1. Установка Python, IDE, простые типы данных

Модуль 2. Условный оператор if

Модуль 3. Последовательности и циклы

Модуль 4. Список как изменяемая последовательность

Модуль 5. Кортежи, словари и множества

Модуль 6. Функции

Модуль 7. Создание модулей.

Модуль 8. Файлы, работа с файловой системой

Модуль 9. Обработка исключений

Модуль 10. Регулярные выражения

Модуль 11. Получение данных из разных источников

С чем работаем?

- Python актуальной версии
 - VSCode
- (и это всё)



<https://www.python.org/downloads/>

<https://code.visualstudio.com/download/>

Некоторые важные свойства

- Python – интерпретируемый, а не компилируемый язык
- Код сначала преобразуется в промежуточный формат, называемый byte code (это не машинный код!), а затем выполняется интерпретатором.

Hello World

```
print('Hello, world!')
```

всё!

Дзен python (import this)

1. Красивое лучше, чем уродливое.
2. Явное лучше, чем неявное.
3. Простое лучше, чем сложное.
4. Сложное лучше, чем запутанное.
5. Плоское лучше, чем вложенное.
6. Разреженное лучше, чем плотное.
7. Читаемость имеет значение.
8. Особые случаи не настолько особые, чтобы нарушать правила.
9. При этом практичность важнее безупречности.
10. Ошибки никогда не должны замалчиваться.
11. Если они не замалчиваются явно.
12. Встретив двусмысленность, отбрось искушение угадать.
13. Должен существовать один и, желательно, только один очевидный способ сделать это.
14. Хотя он поначалу может быть и не очевиден, если вы не голландец [*Гвидо ван Россум, создатель языка*].
15. Сейчас лучше, чем никогда.
16. Хотя никогда зачастую лучше, чем прямо сейчас.
17. Если реализацию сложно объяснить — идея плоха.
18. Если реализацию легко объяснить — идея, возможно, хороша.
19. Пространства имён — отличная штука! Будем делать их больше!

Простейшие типы данных

type()	тип	примеры
int	целое число	1, -10, 25, 99999999999999999999
float	число с плавающей запятой	1.0, 20., .5, -0.9999
bool	логическая переменная	True, False
str	строка	'Hello world!', "Привет", '''Hello'''

Некоторые важные свойства

- В Python используется динамическая типизация
- За переменной не закреплён какой-то определённый тип данных
- Тип переменной может меняться в ходе выполнения программы

Присваивание

- › Используем знак равенства

`a = 10`

- › Сначала вычисляется значение выражение справа от знака равенства, а потом результат присваивается переменной

`b = a + 5`

- › В версии 3.8 появился ещё один оператор (`:=`, walrus operator), о нем поговорим когда он нам понадобится

Некоторые важные свойства

- Управление памятью (сбор мусора) в Python автоматическое
- Как только уничтожается последняя ссылка, указывающая на какой-то объект, уничтожается и сам объект

Арифметические операции

$x + y$	сложение
$x - y$	вычитание
$x * y$	умножение
x / y	деление
$x // y$	целочисленное деление
$x \% y$	остаток от деления
$x ** y$	возведение в степень
$-x$	смена знака

Логические операции

$x \text{ or } y$	или
$x \text{ and } y$	и
$\text{not } x$	инверсия

Операции сравнения

<	Строго меньше
<=	Меньше или равно
>	Больше
>=	Больше или равно
==	Равно (не путать с присваиванием!)
!=	Не равно
is, is not	Равенство и неравенство) объектов

Битовые операции

$x \mid y$	побитовое или
$x \wedge y$	побитовое исключающее или
$x \& y$	побитовое и
$x \ll n$	сдвиг на n элементов влево (умножение)
$x \gg n$	сдвиг на n элементов влево (деление)
$\sim x$	побитовая инверсия

Сокращённое присваивание

➤ Модификацию переменной можно записывать сокращённо:

$x = x + 5$ равносильно $x \text{ += } 5$

$a = a * b$ равносильно $a \text{ *= } b$

$c = c \% 3$ равносильно $c \text{ \%=} 3$

$d = d \& 3$ равносильно $d \text{ \&}= 3$

и так далее

При этом между знаком оператора и знаком равенства не должно быть пробелов

Полезные ссылки

- Официальная документация: <https://docs.python.org/3/>
- PEP20 Python Zen (Tim Peters) <https://www.python.org/dev/peps/pep-0020/>
- PEP8 Стиль написания кода <https://www.python.org/dev/peps/pep-0008/> (EN)