

Занятие 1

Что такое python ?

Python - **высокоуровневый**, **интерпретируемый** язык программирования общего назначения с **динамической строгой типизацией** и автоматическим управлением памятью. Язык является полностью объектно-ориентированным в том плане, что всё является объектами.

Для чего он используется ?

- Веб разработка
- Анализ данных и машинное обучение
- Игры
- Десктоп-приложения

С чего начать ?

Установка Python

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

Выбор IDLE

- VSCode(моя личная рекомендация) <https://code.visualstudio.com/>
- PyCharm <https://www.jetbrains.com/pycharm/>
- Sublime Text <https://www.sublimetext.com/>

Первая программа

```
print('Hello world')
```

Основные понятия:

Синтаксис

Простая программа

```
a=10#1 строка
b=11#2 строка
word='simpleword'#3 строка
print(a,b,c)#4 строка
# output 10 11 simpleword
```

Переменные

Определение

Переменная в языке программирования это название для зарезервированного места в памяти компьютера, предназначенное для хранения значений. У переменных есть несколько возможностей: **чтение** переменной и **запись** в переменную.

```
a=10 # запись в переменную
print(a)# чтение переменной
```

Типы переменных

- Целые числа (int)
- Числа с плавающей точкой (float)
- Строки (str)
- Булевы значения (bool)
- Другие типы, о которых вы узнаете позже.

Базовые операторы

| Оператор | Название | Объяснение | Примеры |
|----------|-----------------------|--|--|
| „+“ | Сложение | Суммирует два объекта | 3 + 5 даст 8; „a“ + „b“ даст „ab“ |
| „-“ | Вычитание | Даёт разность двух чисел; если первый операнд отсутствует, он считается равным нулю | -5.2 даст отрицательное число, а 50 - 24 даст 26. |
| „*“ | Умножение | Даёт произведение двух чисел или возвращает строку, повторённую заданное число раз. | 2 * 3 даст 6. „la“ * 3 даст „lalala“. |
| „**“ | Возведение в степень | Возвращает число x, возведённое в степень y | 3** 4 даст 81 (т.е. 3 * 3 * 3 * 3) |
| / | Деление | Возвращает частное от деления x на y | 4 / 3 даст 1.3333333333333333. |
| // | Целочисленное деление | Возвращает неполное частное от деления | 4 // 3 даст 1. -4 // 3 даст -2. |
| % | Деление по модулю | Возвращает остаток от деления | 8 % 3 даст 2. -25.5 % 2.25 даст 1.5. |
| << | Сдвиг влево | Сдвигает биты числа влево на заданное количество позиций. (Любое число в памяти компьютера представлено в виде битов - или двоичных чисел, т.е. 0 и 1) | 2 << 2 даст 8. В двоичном виде 2 представляет собой 10. Сдвиг влево на 2 бита даёт 1000, что в десятичном виде означает 8. |
| >> | Сдвиг вправо | Сдвигает биты числа вправо на заданное число позиций. | 11 >> 1 даст 5. В двоичном виде 11 представляется как 1011, что будучи смещённым на 1 бит вправо, даёт 101, а |

| | | | |
|----|-----------------------------|--|---|
| | | | это, в свою очередь, не что иное как десятичное 5 |
| & | Побитовое И | Побитовая операция И над числами | 5 & 3 даёт 1. |
| | Побитовое ИЛИ | Побитовая операция ИЛИ над числами | 5 3 даёт 7 |
| ^ | Побитовое ИСКЛЮЧИТЕЛЬНО ИЛИ | Побитовая операция ИСКЛЮЧИТЕЛЬНО ИЛИ | 5 ^ 3 даёт 6 |
| ~ | Побитовое НЕ | Побитовая операция НЕ для числа x соответствует $-(x+1)$ | ~5 даёт -6. |
| < | Меньше | Определяет, верно ли, что x меньше y. Все операторы сравнения возвращают True или False [1]. Обратите внимание на заглавные буквы в этих словах. | 5 < 3 даст False, а 3 < 5 даст True. Можно составлять произвольные цепочки сравнений: 3 < 5 < 7 даёт True. |
| > | Больше | Определяет, верно ли, что x больше y | 5 > 3 даёт True. Если оба операнда - числа, то перед сравнением они оба преобразуются к одинаковому типу. В противном случае всегда возвращается False. |
| <= | Меньше или равно | Определяет, верно ли, что x меньше или равно y | x = 3; y = 6; x <= y даёт True. |
| >= | Больше или равно | Определяет, верно ли, что x больше или равно y | x = 4; y = 3; x >= 3 даёт True. |

| | | | |
|-----|----------------|--|--|
| == | Равно | Проверяет, одинаковы ли объекты | x = 2; y = 2; x == y даёт True. x = „str“; y = „stR“; x == y даёт False. x = „str“; y = „str“; x == y даёт True. |
| != | Не равно | Проверяет, верно ли, что объекты не равны | x = 2; y = 3; x != y даёт True. |
| not | Логическое НЕ | Если x равно True, оператор вернёт False. Если же x равно False, получим True. | x = True; not x даёт False. |
| and | Логическое И | x and y даёт False, если x равно False, в противном случае возвращает значение y | x = False; y = True; x and y возвращает False, поскольку x равно False. В этом случае Python не станет проверять значение y, так как уже знает, что левая часть выражения 'and' равняется False, что подразумевает, что и всё выражение в целом будет равно False, независимо от значений всех остальных операндов. Это называется укороченной оценкой булевых (логических) выражений. |
| or | Логическое ИЛИ | Если x равно True, в результате получим True, в противном случае получим значение y | x = True; y = False; x or y даёт True. Здесь также может производиться укороченная оценка выражений. |

Управляющие операторы

Условные операторы

Оператор if используется для проверки условий: если условие верно, выполняется блок выражений (называемый “if-блок”), иначе выполняется другой блок

выражений (называемый “else-блок”). Блок “else” является необязательным.

```
a=25
if a > 0:
    print('a>0')
elif a<0:
    print('a<0')
else:
    ???
# output: a > 0
```

Поэкспериментируйте со значением переменной “a”. Что должно быть в блоке else ?



В языке Python блоки разделяются табами или пробелами

Запомните эмпирическое правило: хотя вы можете использовать для отступов пробелы или табуляции, их смешивание внутри блока обычно не будет удачной идеей применяйте либо то, либо другое. Формально табуляция считается достаточным количеством пробелов, чтобы сместить текущую строку на расстояние, кратное 8, и код будет работать в случае согласованного смешивания табуляций и пробелов. Тем не менее, такой код может быть сложнее изменять. Хуже того, смешивание табуляций и пробелов затрудняет чтение кода целиком, не говоря уже о правилах синтаксиса Python — табуляции в редакторе сменившего вас программиста могут выглядеть совсем не так, как в вашем редакторе.

Источник: <https://pylessons.readthedocs.io/ru/latest/basics/operators.html>