

Языки программирования

Лекция 4

Переменные

```
>a = 2 # переменной присваиваем значение 2
>b = 3
>print(a + b)
>a = 6 # присваиваем новое значение
>print(a + b)
>b = b + 2
>print(b)
>print(c) # ошибка: любая переменная должна
быть проинициализирована перед
использованием
```

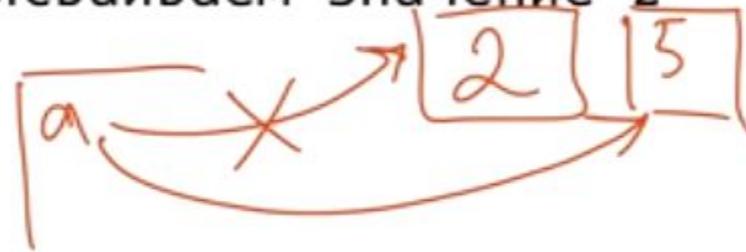
Оператор присваивания

✧ $a = 2$ # переменной присваиваем значение 2

a — имя переменной

= — оператор присваивания

2 — значение



$a = 5$

✧ $2 = a$ # ошибка, имя переменной должно быть слева от оператора присваивания

Несимметричный оператор

Оператор приращения

- > `a = 2` # переменной присваиваем значение 2
- > `a += 3` # увеличиваем значение на 3
- > `a = a + 3`

`+=` `--` `*=` `/=` `//=` `%=` `**=`

Имя переменной

- › `a = 2` # переменной присваиваем значение 2
a — имя переменной
- Имя переменной
 - может состоять из букв (строчных и прописных), цифр, подчеркивания `_`
 - должно начинаться с буквы или подчеркивания
 - не должно являться ключевым словом
 - регистр букв имеет значение

Динамическая типизация

- > a = 2
- > a = 'abacaba'
- > a = foo()

type(a) - ?

Ввод данных

- > `input()` # читает строку с клавиатуры
- > `input('Введите данные')` Пригласительное сообщение

- > `s = input()` # записать в переменную `s` пользовательский ввод
- > `a = int(input())` # прочитать строку с клавиатуры и преобразовать в число

Вывод данных: `print('some output')`

Переменные

```
In [28]: a = 3
```

```
In [29]: a
```

```
Out[29]: 3
```

```
In [30]: a += 4
```

```
In [31]: a
```

```
Out[31]: 7
```

```
In [32]: a  
2 * a # выводится только последнее значение и только при работе в интерактивном  
режиме!
```

```
Out[32]: 14
```

Для вывода значений в своих программах используйте функцию `print()`. Обратите внимание на наличие скобок при вызове функции `print`!

```
In [33]: print(a)
         print(2 * a)
```

```
7
14
```

Можно выводить диалоговые сообщения при 'общении' с пользователем. Но не отправляйте в проверочную систему программы, содержащие лишний вывод

```
In [34]: name = input('Enter your name: ')
         print('Hello ', name)
```

```
Enter your name: Pavel
Hello Pavel
```

```
In [35]: a = int(input())
         print(a * 2)
```

```
12
24
```

```
In [36]: a = int(input())
         b = int(input())
         print(a * b)
```

```
5
7
35
```

Задача

- Напишите программу:
- Тимофей обычно спит ночью X часов и устраивает себе днем тихий час на Y минут. Определите, сколько всего минут Тимофей спит в сутки.
- Внимание, программа принимает значения X и Y из стандартного потока ввода (функция `input`), результат надо выводить в стандартный поток вывода (функция `print`). Обратите внимание на то, что приглашение, переданное в качестве аргумента в функцию `input`, считается выводом вашей программы. Используйте эту функцию без аргументов:
- `values = input()` # без строки приглашения!

Решение

- `X = int(input())`
- `Y = int(input())`
- `print(X*60 + Y)`

Задача

- Коля каждый день ложится спать ровно в полночь и недавно узнал, что оптимальное время для его сна составляет X минут. Коля хочет поставить себе будильник так, чтобы он прозвенел ровно через X минут после полуночи, однако для этого необходимо указать время сигнала в формате часы, минуты. Помогите Коле определить, на какое время завести будильник.
- **Sample Input 1:** 480
- **Sample Output 1:**
 - 8
 - 0
- **Sample Input 2:** 512
- **Sample Output 2:**
 - 8
 - 32

Решение

- `X = int(input())`
- `print(X // 60)`
- `print(X % 60)`

Задача

- Катя узнала, что ей для сна надо X минут. В отличие от Коли, Катя ложится спать после полуночи в N часов и M минут. Помогите Кате определить, на какое время ей поставить будильник, чтобы он прозвенел ровно через X минут после того, как она ляжет спать.
- На стандартный ввод, каждое в своей строке, подаются значения X , N и M . Гарантируется, что Катя должна проснуться в тот же день, что и заснуть. Программа должна выводить время, на которое нужно поставить будильник: в первой строке часы, во второй — минуты.
- **Sample Input 1:** 480 1 2
- **Sample Output 1:**
- 9
- 2
- **Sample Input 2:** 475 1 55
- **Sample Output 2:**
- 9
- 50

Решение

- `X = int(input())`
- `H = int(input())`
- `M = int(input())`
- `print(X // 60 + H + (X % 60 + M) // 60)`
- `print((X % 60 + M) % 60)`

Логические значения

False (0)

True (1)

Операции:

Унарная операция

<u>x</u>	<u>y</u>	<u>x or y,</u>	<u>x and y,</u>	<u>not x</u>
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

Истина, если хотя бы один из операндов является истиной

Истина, если оба операнда являются истиной

Операции сравнения

< строго меньше

<= меньше или равно

> строго больше

>= больше или равно

== равно

!= не равно

5 < 7
True
5 == 2 + 3

Логические операции

```
In [37]: a = int(input())  
print(a > 0)
```

```
-234  
False
```

Проверим, является ли число
положительным

```
In [38]: a = int(input())  
print(a >= 10 and a < 100)
```

```
10  
True
```

Проверим, является ли число
двухзначным

```
In [39]: a = int(input())  
print(10 <= a < 100)
```

```
23  
True
```

Совместили 2 условия

```
In [40]: x1, x2, x3 = False, True, False  
not x1 or x2 and x3
```

1) not, 2) and, 3) or

```
Out[40]: True
```

Добавляя скобки в выражения, можно изменить порядок вычисления и значение результирующего выражения. Если не уверены в приоритете операций, смело добавляйте скобки, чтобы быть уверенными в том, что выражение вычисляется именно так, как вы хотите

```
In [41]: ((not x1) or x2) and x3
```

```
Out[41]: False
```

Задача

- Расставьте скобки в выражении
- **a and b or not a and not b**
- в соответствии с порядком вычисления выражения (приоритетом операций).
Всего потребуется 5 пар скобок
(внешние скобки входят в их число).

Решение

- $((a \text{ and } b) \text{ or } ((\text{not } a) \text{ and } (\text{not } b)))$

Задача

- $x = 5$
- $y = 10$
- $y > x * x$ or $y \geq 2 * x$ and $x < y$
- Помните, что любые арифметические операции выше по приоритету операций сравнения и логических операторов.

Решение

- True

Задача

- $a = \text{True}$
 $b = \text{False}$
 $a \text{ and } b \text{ or not } a \text{ and not } b$

Решение

- False

УСЛОВИЯ

```
if x % 2 == 0:  
    print('Четное')  
else:  
    print('Нечетное')
```

Проверка на четность

Обратите внимание на
отступы!

УСЛОВИЯ

`if` условие_1:

- Блок 1

`elif` условие_2:

- Блок 2

`else`:

- Блок 3

Блок 3 – если ни одно из условий не выполнилось!

Блоков `elif` и `else` может не быть

Найти наибольшее из двух чисел

```
a = 4
```

```
b = 7
```

```
if a >= b:
```

```
    print(a)
```

```
else:
```

```
    print(b)
```

```
a = 4
```

```
b = 7
```

```
m = a    m - максимум
```

```
if b > m:
```

```
    m = b
```

```
print(m)
```

Условия

```
In [42]: a = int(input())
         b = int(input())
         print(a / b)
```

```
5
10
0.5
```

```
In [43]: a = int(input())
         b = int(input())
         if b != 0:
             print(a / b)
         else:
             print('Деление невозможно')
```

```
5
0
Деление невозможно
```

```
In [44]: a = int(input())
         b = int(input())
         if b != 0:
             print(a / b)
         else:
             print('Деление невозможно')
             b = int(input('Введите ненулевое значение '))
             print(a / b)
```

```
4
0
Деление невозможно
Введите ненулевое значение 2
2.0
```

In [45]:

```
a = int(input())
b = int(input())
if b != 0:
    print(a / b)
else:
    print('Деление невозможно')
    b = int(input('Введите ненулевое значение '))
    if b == 0:
        print('Вы не справились!')
    else:
        print(a / b)
```

5

0

Деление невозможно

Введите ненулевое значение 0

Вы не справились!

In [47]:

```
x = int(input())
if x % 2 == 0:
    print('Четное')
else:
    print('Нечетное')
```

25

Нечетное