



ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ НА PYTHON

ТИПЫ ДАННЫХ

Boolean
(логические типы
данных)

Numbers (числа)

String (строки)

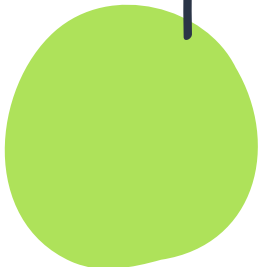
(Тут могла бы
быть ваша
реклама)

List (списки)

Set (множества)

Tuples (кортежи)

Dictionaries
(словари)



ТИПЫ ДАННЫХ ПО ИЗМЕНЯЕМОСТИ

Boolean

(логические типы
данных)



Set (множества)

Numbers (числа)

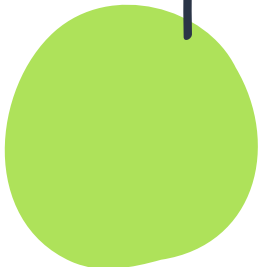
ИЗМЕНЯЕМЫЕ
НЕИЗМЕНЯЕМЫЕ

Tuples (кортежи)

String (строки)

List (списки)

Dictionaries
(словари)



ТИПЫ ДАННЫХ ПО УПОРЯДОЧЕННОСТИ



Set (множества)

УПОРЯДОЧЕННЫЕ
НЕУПОРЯДОЧЕННЫЕ

Tuples (кортежи)

String (строки)

List (списки)

Dictionaries
(словари)

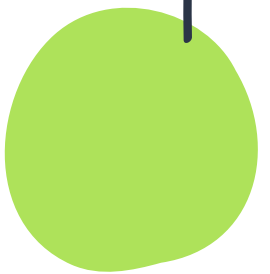
ЧИСЛА

Integer (1)

- X Сложение
- X Вычитание
- X Деление
- X Умножение
- X Преобразование в Float

Float (1.0)

- X Сложение
- X Вычитание
- X Деление
- X Умножение
- X Преобразование в Integer



ОПЕРАТОРЫ ДЛЯ РАБОТЫ С ЧИСЛАМИ

Оператор	Значение
+	Сложение
-	Вычитание
/	Деление
//	Целочисленное деление
*	Умножение
%	Остаток от деления
**	Возведение в степень
abs(число)	Модуль числа
round(число, кол.-во знаков)	Округление числа

In [10]:

```
# Просто вычитание  
1 - 3
```

Out[10]: -2

In [11]:

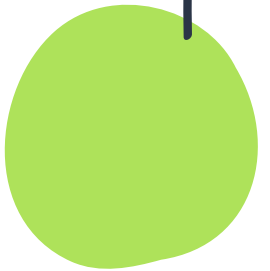
```
#Пример простого умножения  
5 * 6
```

Out[11]: 30

In [12]:

```
#Деление  
30 / 5
```

Out[12]: 6.0



In [6]: Slide Type Fragment ▼

```
# Сложение integer + float.  
1 + 1.0
```

Out[6]: 2.0

In [7]: Slide Type Slide ▼

```
#Integer можно легко преобразовать во float, а float в integer просто прописав перед числом необходимый тип данных  
  
#Преобразование float в integer  
int(2.0)
```

Out[7]: 2

In [8]: Slide Type Fragment ▼

```
#Преобразование integer во float  
float(2)  
  
int(2.5)
```

Out[8]: 2

Slide Type Fragment ▼

Важно понимать, что несмотря на то, что любой integer можно преобразовать во float и наоборот, но при преобразовании float в integer число будет автоматически округляться, по арифметическим правилам. Но есть определенные подводные камни.

In [9]: Slide Type Fragment ▼

```
#Преобразуем float в integer  
round(2.50)
```

Out[9]: 2

In [25]:

Slide Type Slide ▾

```
#Введение в инженерный калькулятор  
sqrt(4)
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-25-6c62a49b74bc> in <module>  
      1 #Введение в инженерный калькулятор  
>>> 2 sqrt(4)  
  
NameError: name 'sqrt' is not defined
```

БИБЛИОТЕКИ



Разработанная
библиотека

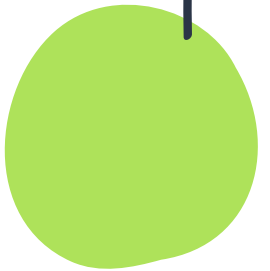


Ручная
реализация
библиотеки

РЕПОЗИТОРИИ ДЛЯ БИБЛИОТЕК

x pip

x anaconda



ИМПОРТ БИБЛИОТЕКИ

In [26]:

```
# Использование sqrt из библиотеки Math
import math
math.sqrt(4)
```

Slide Type Slide ▾

Out[26]: 2.0

In [27]:

```
import math as m
m.sqrt(4)
```

Slide Type Slide ▾

Out[27]: 2.0

In [29]:

```
#Импорт метода sqrt из библиотеки math с названием sq
from math import sqrt as sq
sq(4)
```

Slide Type Fragment ▾

Out[29]: 2.0

ПРИМЕНЕНИЕ БИБЛИОТЕКИ MATH

In [30]:

Slide Type Fragment ▾

```
# Другие функции библиотеки math для работы с числами. Посчитаем косинус пи с помощью math из Python  
math.cos(math.pi)
```

Out[30]: -1.0

In [31]:

Slide Type Fragment ▾

```
# Использование натурального логарифма  
math.log(2)
```

Out[31]: 0.6931471805599453

In [32]:

Slide Type Fragment ▾

```
# Использование 10 го логарифма  
math.log10(10)
```

Out[32]: 1.0

ПЕРЕМЕННЫЕ И ЧИСЛА



ПЕРЕМЕННЫЕ И ЧИСЛА



Альпака 1



Альпака 2



Альпака 3



Альпака 4



Альпака 5



Альпака 6



Альпака 7

Глобальные
переменные



Альпака 1



Альпака 2



Альпака 3



Альпака 4

Локальные переменные



Альпака 5



Альпака 6



Альпака 7

Глобальные
переменные



Альпака 1



Альпака 2



Альпака 3



Альпака 4

Локальные переменные



Альпака 1



Альпака 6



Альпака 7

Глобальные
переменные



Альпака 1



Альпака 2



Альпака 3



Альпака 4

Локальные переменные



Альпака 7



Альпака 6



Альпака 7

НАЗНАЧЕНИЕ ПЕРЕМЕННЫХ

In [33]:

```
x = 1 + 2
```

Slide Type Fragment

In [35]:

```
#Способ 1. Просто назначим переменную y = 8  
y = 8  
y
```

Slide Type Fragment

Out[35]: 8

In [36]:

```
#Назначим переменную y с помощью переменной x, увеличив ее на 5  
y = x + 5  
y
```

Slide Type Fragment

Out[36]: 8

In [37]:

```
#Проверим, чем равна переменная y  
y
```

Slide Type Fragment

Out[37]: 8

In [38]:

```
# Увеличим переменную y на 5, чтобы получить 13  
y = y + 5  
y
```

Slide Type Fragment

Out[38]: 13

ОПЕРАТОРЫ РАБОТЫ С ПЕРЕМЕННЫМИ

Оператор	Значение
<code>+=</code>	эквивалентна <code>x = x + 'что-то'</code>
<code>-=</code>	эквивалентна <code>x = x - 'что-то'</code>
<code>*=</code>	<code>x = x * 'что-то'</code>
<code>%=</code>	<code>x = x % 'что-то'</code>
<code>/=</code>	<code>x = x / 'что-то'</code>
<code>//=</code>	<code>x = x // 'что-то'</code>
<code>**=</code>	<code>x = x ** 'что-то'</code>

ПРИМЕНЕНИЕ ОПЕРАТОРОВ ДЛЯ РАБОТЫ С ПЕРЕМЕННЫМИ

In [4]:

```
#Также данный оператор может вычитать, умножать или делить или возводить в степень  
k = 3  
k *= 5  
k
```

Slide Type Slide ▾

Out[4]: 15

In [42]:

```
l = 10  
l -= 3  
l
```

Slide Type Fragment ▾

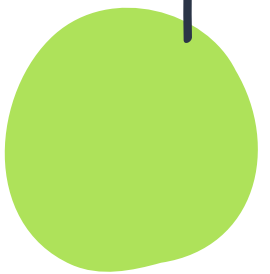
Out[42]: 7

In [43]:

```
n = 3  
n **= 2  
n
```

Slide Type Fragment ▾

Out[43]: 9



ЛОГИЧЕСКИЕ ТИПЫ ДАННЫХ, TRUE И FALSE

In [44]:

```
# проверим равенство чисел  
0 == 1
```

Slide Type Slide ▾

Out[44]: False

In [45]:

```
# Проверим, соответствует тип данных 1 типу Integer  
type(1) == int
```

Slide Type Fragment ▾

Out[45]: True

INPUT

In [48]:

```
# Попробуем получить 2 числа из input и сложить их
first_input = input()
second_input = input()
res = first_input + second_input
res
```

Slide Type Slide ▾

INPUT

In [48]:

```
# Попробуем получить 2 числа из input и сложить их
first_input = input()
second_input = input()
res = first_input + second_input
res
```

1
2

Out[48]: '12'

Slide Type Slide ▾

INPUT ДЛЯ РАБОТЫ С ЧИСЛАМИ

In [49]:

```
# Попробуем получить 2 числа из input и сложить их с учетом преобразования
first_input = int(input())
second_input = int(input())
res = first_input + second_input
res
```

1
2

Out[49]: 3

PRINT

In [50]:

```
# напечатать строку res  
print(res)
```

3

Slide Type Slide



ПАРАМЕТРЫ, КОТОРЫЕ МОЖНО ПЕРЕДАТЬ PRINT

In [54]:

Slide Type Slide ▾

```
# Использование параметра sep для разделения переменных в print(). Разделим переменные
# с помощью символа переноса строки \n
n1 = 2
n2 = 5
n3 = 6
print(n1, n2, n3, sep='\n')
```

```
2
5
6
```

In [55]:

Slide Type Fragment ▾

```
# Тоже самое без разделителя
n1 = 2
n2 = 5
n3 = 6
print(n1, n2, n3)
```

```
2 5 6
```

In [56]:

Slide Type Slide ▾

```
# Использование параметра end для установления последнего символа в выводимом сообщении
print(n1, n2, n3, end='|тут заканчивается строка|')
```

```
2 5 6|тут заканчивается строка|
```

ОБЪЕДИНЕНИЕ INPUT И PRINT

In [53]:

Slide Type Fragment ▾

```
print('Введите первое число')
first_input = int(input())
print('Первое введенное число =', first_input)
print('Введите второе число')
second_input = int(input())
print('Второе введенное число =', second_input)
res = first_input + second_input
print('Результат сложения first_input и second_input =', res)
```

Введите первое число

2

Первое введенное число = 2

Введите второе число

3

Второе введенное число = 3

Результат сложения first_input и second_input = 5

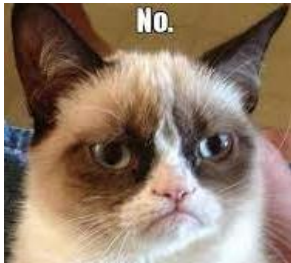
ЛОГИЧЕСКИЕ КОНСТРУКЦИИ IF, ELIF, ELSE



ОБЩАЯ СХЕМА РАБОТЫ ЛОГИЧЕСКИХ ОПЕРАТОРОВ

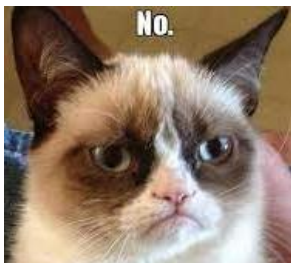
if

Выполнение условия
для if



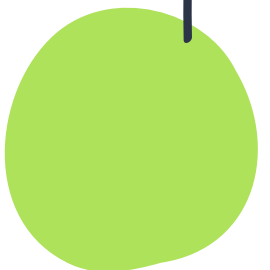
elif

Выполнение условия
для elif



else

Выполнение условия
при несовпадении if
и elif



РАБОТА ОПЕРАТОРА If

In [58]:

Slide Type Fragment ▾

```
year = int(input())
if (2021 - year) > 18:
    print('Действительно больше 18')
```

1998
Действительно больше 18

In [59]:

Slide Type Fragment ▾

```
year = int(input())
age = 2021 - year
if (age > 18) and (age < 150):
    print('Действительно больше 18 и меньше 150')
```

1950
Действительно больше 18 и меньше 150

In [60]:

Slide Type Fragment ▾

```
# Применение слова or
year = int(input())
age = 2021 - year
if (age > 18) or (age == 13):
    print('Действительно больше 18 или равно 13')
```

2008
Действительно больше 18 или равно 13

КОНСТРУКЦИЯ IF ELSE

In [62]:

```
inp_number = int(input())  
  
if inp_number > 10:  
    print('Больше')  
else:  
    print('Меньше')
```

3
Меньше

Slide Type Slide ▾

КОНСТРУКЦИЯ IF, ELIF, ELSE

In [63]:

```
a = 2
if a > 0:
    print(1)
elif a == 2:
    print(2)
```

Slide Type Slide ▾

In [64]:

```
# Использование конструкции if elif else
inp_num = int(input())
if inp_num > 100:
    print('Слишком большое число')
elif inp_num > 50 and inp_num < 90:
    print('Число между 50 и 90')
elif inp_num > 10 and inp_num < 50:
    print('Число между 10 и 50')
else:
    print('Число явно меньше 10')
```

Slide Type Slide ▾

13

Число между 10 и 50