

Основы Python

In [1]:

```
1 + 2
```

Про выполнение команд

- Shift + Enter - выполнить код в ячейке и перейти на следующую
- Ctrl + Enter - выполнить и остаться

Немного полезностей и про markdown:

- для автозаполнения длинных названий используйте табуляцию
- перевести ячейку из кода в комментарии можно с помощью 'Esc + m' или просто 'm', если вы не еще не успели сделать ее активной
- добавить новую ячейку можно аналогичным образом через 'b'
- Jupyter понимает эти команды и в русской раскладке
- все это можно сделать и через верхнюю панель под меню File, Edit, ...
- список горячих клавиш Ctrl+Shift+P (Command+Shift+P для мака)

In [2]:

```
a = 1
```

Поддержка HTML

Jupyter умеет выполнять HTML-код и понимает формулы:

$$\vec{a} \cdot \vec{b} = |\vec{a}| |\vec{b}| \cos(\vec{a}, \vec{b})$$

Виджеты

In [3]:

```
from ipywidgets import IntProgress
from IPython.display import display
import time
```

In [4]:

```
progress = IntProgress(min=0, max=100, value=0)
display(progress)

for i in range(100):
    time.sleep(0.05)
    progress.value = i
```

Арифметические операции

In [5]:

```
print(1 + 1)
```

2

In [6]:

```
print(2 - 1, 5 + 6)
```

1 11

In [7]:

```
print(2-1)
```

1

In [8]:

```
# ...
1 + 1 # для вывода в jupyter notebook не обязательно использовать print
```

Out[8]:

2

In [9]:

```
1 + 1
2 - 1
```

Out[9]:

1

In [10]:

```
print(1 + 1)
print(2 - 1)
```

2

1

In [11]:

```
5 * 6
```

Out[11]:

30

In [12]:

```
10 / 4
```

Out[12]:

2.5

In [13]:

```
10 // 4
```

Out[13]:

2

In [14]:

```
10 = 4*2 + 2
```

File "<ipython-input-14-8918154505c8>", line 1

```
10 = 4*2 + 2
```

^

SyntaxError: cannot assign to literal

In [15]:

```
2 ** 100
```

Out[15]:

1267650600228229401496703205376

In [16]:

```
2*2*2*2*2
```

Out[16]:

32

In [17]:

```
-4 % 3
```

Out[17]:

2

In [18]:

```
5 - (3 + 2) * 6
```

Out[18]:

-25

Переменные

In [19]:

```
a = 10 + 20  
b = a * 30  
c = a / b  
print('Ответ: ', a, b, c)
```

Ответ: 30 900 0.03333333333333333

In [20]:

b

Out[20]:

900

Переменная должна объявляться до того, как использоваться

In [21]:

```
x = 10  
print(x)
```

10

Строки

In [22]:

```
my_string_1 = 'He said: "Hello"'
```

In [23]:

```
my_string_2 = "Hello \"World!\""
```

In [24]:

```
my_string_1
```

Out[24]:

'He said: "Hello"'

Операторы сравнения

In [25]:

```
2 > 1
```

Out[25]:

True

In [26]:

```
10 < 9
```

Out[26]:

False

In [27]:

```
10 > 10
```

Out[27]:

False

In [28]:

```
10 >= 10
```

Out[28]:

True

In [29]:

```
7 == 7
```

Out[29]:

True

In [30]:

```
a = 1
```

In [31]:

```
a == 1
```

Out[31]:

True

In [32]:

```
<>
```

File "<ipython-input-32-ae3fe16fc7d9>", line 1

```
<>
  ^
```

SyntaxError: invalid syntax

In [33]:

```
15 != 15
```

Out[33]:

False

In [34]:

```
5 > 4 < 0
```

Out[34]:

False

In [35]:

```
# сравнение строк  
print('c' > 'a')
```

True

In [36]:

```
print('b' > 'B')
```

True

In [37]:

```
print('abc' > 'aba')
```

True

In [38]:

```
len('abc') > len('dfhjdfj')
```

Out[38]:

False

Условные конструкции

In [39]:

```
a = 1  
b = 2  
  
if a < b:  
    print('Значение переменной a меньше b')
```

Значение переменной a меньше b

In [40]:

```
x = 6
if x % 2 == 0:
    print(x, '- четное число')
```

6 - четное число

In [41]:

```
x = 5
if x % 2 == 0:
    print(x, '- четное число')
else:
    print(x, '- нечетное число')
```

5 - нечетное число

In [42]:

```
'2020-05-01' < '2020-05-09'
```

Out[42]:

True

In [43]:

```
'2019-12-31' < '2020-01-01'
```

Out[43]:

True

In [44]:

```
'logs_2020-02-29.csv' < 'logs_2020-03-01.csv'
```

Out[44]:

True

Пишем программу для определения куда направить призывника

In [45]:

```
height = 180
if height < 170:
    print('В танкисты')

print('Конец программы')
```

Конец программы

In [46]:

```
if height < 170:  
    print('В танкисты')  
else:  
    print('В другие войска')  
print('Конец программы')
```

В другие войска
Конец программы

In [47]:

```
# условия выполняются по порядку, как только одно выполнено - проверки останавливаются  
height = 210  
if height < 170:  
    print('В танкисты')  
elif height < 200:  
    print('В десантники')  
else:  
    print('В другие войска')  
print('Конец программы')
```

В другие войска
Конец программы

In [48]:

```
# блоков elif может быть хоть сколько  
height = 210  
if height < 170:  
    print('В танкисты')  
elif height < 180:  
    print('На флот')  
elif height < 200:  
    print('В десантники')  
else:  
    print('В другие войска')  
print('Конец программы')
```

В другие войска
Конец программы

Логические операторы and и or

In [49]:

```
True and True
```

Out[49]:

True

In [50]:

```
True and False
```

Out[50]:

False

In [51]:

```
True or False
```

Out[51]:

True

In [52]:

```
not True
```

Out[52]:

False

In [53]:

```
((9 > 7) and (2 < 4))
```

Out[53]:

True

In [54]:

```
(8 == 8) or (6 != 6)
```

Out[54]:

True

In [55]:

```
not(3 <= 1)
```

Out[55]:

True

In [56]:

```
age = 15
height = 180

if height < 170 and age >= 18:
    print('В танкисты')
elif height < 185 and age >= 18:
    print('На флот')
elif height < 200 and age >= 18:
    print('В десантники')
else:
    print('В другие войска')
print('Конец программы')
```

В другие войска
Конец программы

In [57]:

```
age = 21
if age >= 18:
    user_input = int(input('Введите рост'))
    if user_input < 170:
        print('В танкисты')
    elif user_input < 185:
        print('На флот')
    elif user_input < 200:
        print('В десантники')
    else:
        print('В другие войска')
else:
    print('Непризывной возраст')
print('Конец программы')
```

Введите рост181
На флот
Конец программы

In []: