

Программирование для гуманитариев








Урок 1






Put Interactive Python Anywhere on the Web


Customize the code below and  Share!


Play - Воспроизвести
код

main.py

Powered by  trinket
 Приветики!!!



```

1
2
3 print("Приветики!!!")
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

```

Все что вы напишете в скобках будет выведено на экран после нажатия play

Функция print()

```
print("Ваш текст")
```

```
print (5)
```

```
print (5 + 7)
```

```
print (5 * 7)
```

Операции

Операции с числами:

```
print(6 + 2)  # 8
```

```
print(6 - 2)  # 4
```

```
print(6 * 2)  # 12
```

```
print(6 / 2)  # 3
```



Операции

Используем скобки

* и / (умножение и деление)
выполняются первыми как и в школьной
математике

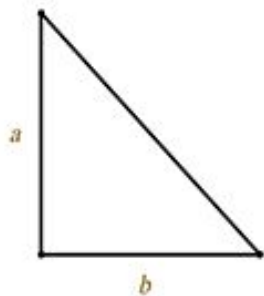
```
print(6 + 2 * 3) → 12
```

```
print((6 + 2) * 3) → 24
```

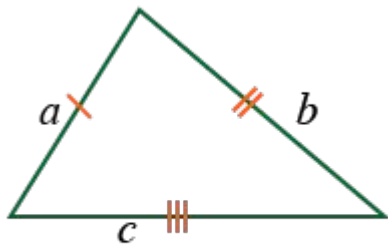
Чтобы первыми выполнить + и - надо воспользоваться
скобками **()**



Страх гуманитария



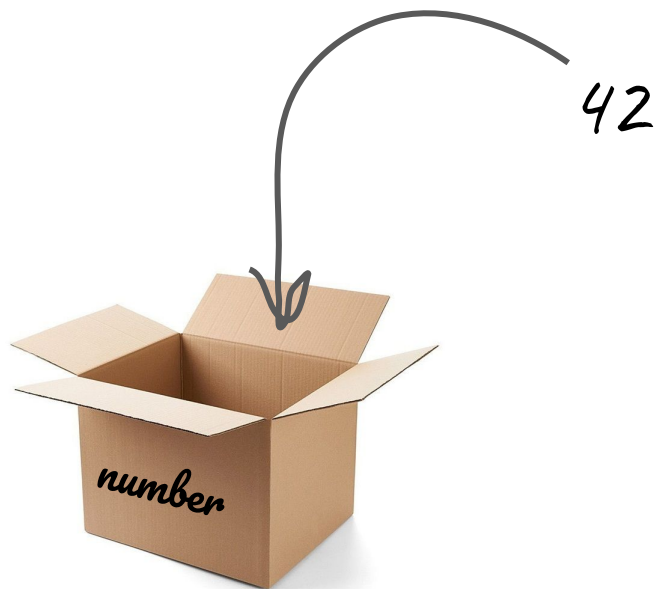
$$S = \frac{ab}{2}$$



$$P = a + b + c$$



Переменная (Variable)



Комментарии

С помощью знака # мы можем оставлять комментарии -

Это часть кода которая будет игнорироваться при выполнении.

Часть строки после # будет игнорироваться



```
6  
7 print (5 + 7) # this will print 12  
8
```



Назначить и переназначить значение

$a = 5$

$a = a + 2$

$c = a$

Какое значение у c ?

$c = 7$

Типы переменных

Каждая переменная имеет свой тип

Имя переменной может отображать, что конкретно в ней хранится:

- возраст можно сохранить в переменную `age`
- имя актера в переменную `name`

```
name = "Leonardo DiCaprio" -> String
```

```
age = 46 -> Integer
```



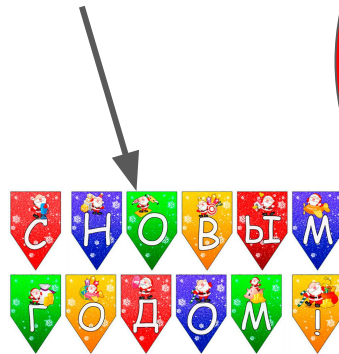
String

Строка (англ. String - тесемка, веревка, нитка)

Так мы называем **фрагменты текста** в программировании, потому что они как последовательность букв на веревке.

```
actor_name = "Leonardo DiCaprio"  
movie = 'Titanic'
```

*Возможны оба варианта скобок " или '



`len()` - проверяет длину слова

`len("Titanic")` → 7 (в слове Titanic 7 знаков)

Работа с текстом

Register

Create your account. It's free and only takes a minute.

First Name

Last Name

Email

Password

Confirm Password

☐ I accept the [Terms of Use & Privacy Policy](#).

Register Now

Already have an account? [Sign in](#)

Работа с текстом

Register

Create your account. It's free and only takes a minute.

First Name

Last Name

Email

Password

Confirm Password

☐ I accept the [Terms of Use & Privacy Policy](#).

Register Now

Already have an account? [Sign in](#)

```
name = "Jack"  
last_name = "Sparrow"  
email = "carrebianLove69@gmail.com"  
password = "ILoveRum1234"
```




Форматируем String

Вставить переменную в текст с помощью %s:

%s - место куда вставим переменную


% - после закрытия скобок, а за ним переменную

```
45
46 number_of_wishes = 3
47
48 print("You have %s wishes" % number_of_wishes)
49
50
```



Вставить **несколько** переменных в текст, нужно использовать скобки

```
16
17 number_of_wishes = 3
18 name = "Leonardo"
19
20 print("%s, you have %s wishes" % (name, number_of_wishes))
21
```



Числа

Integer / целые числа - удобно для отображения количества (жизней, попыток, количество студентов)

```
wishes = 3  
  
lives = 4  
  
num_of_students = 15
```

Целое	
Целое	

Float / числа с плавающей запятой - удобно для измерений: длина, вес, деньги...

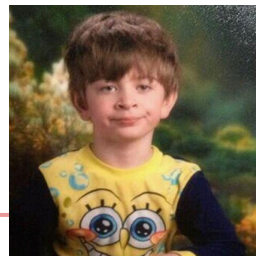
```
length = 1.56  
  
money_in_uah = 100.5  
  
weight_in_kg = 50.64
```

Как называть переменные: Dos and Don'ts

- Используйте маленькие буквы: `lives`, `people`, `books`
- Если название состоит из двух слов - используйте нижнее подчеркивание: `number_of_books`, `weight_in_kg`, `books_on_shelf`
- Следует выбирать понятные названия для переменных
`a = 6` # непонятно, что такое 6, шесть чего??
`num_of_apples = 6` # понятно - количество яблок



- Принято не использовать большие буквы: `Weight`
- Не стоит использовать кириллицу: `количество_жизней`
- Нельзя использовать пробелы: `weight in kg` ---> ошибка! Python воспримет это как отдельные слова
- Нельзя начинать с цифры: `40cats` ---> синтаксическая ошибка!



Переменная

- Хранит только **одно значение**
`number = 5`

- Имеет имя и значение
`number` -> **имя** переменной
`5` -> **значение**

- Имеет какой-то **тип**
`word="house"` -> тип текст
`number=5` -> тип целое число
`distance=5.5` -> тип число с плавающей запятой

- Может **изменять свое значение** по ходу работы программы

`number = 5`

`number = 6`

`number = number + 6` ?? на этом этапе какое значение имеет переменная `number`



User Input

За **ввод в программу данных** с клавиатуры в Python отвечает функция `input()`. Когда вызывается эта функция, программа останавливает свое выполнение и ждет, когда пользователь введет текст.

```
name = input("Введите имя:")
```

```
age = input("Введите возраст:")
```

Что бы не ввел пользователь - тип переменной после использования `input()` будет `string`, то-есть текст.

В будущем это может привести к ошибкам

`input()` - Возвращает string

Числа vs Текст

Integer

a = 400

b = 50

c = a + b

⇒ **c = 450**

Числа vs Текст

Integer

a = 400

b = 50

c = a + b

⇒ c = 450

String

a = "Павло "

b = "Зібров"

c = a + b

⇒ c = "Павло Зібров"

Числа, Текст и +

Integer

a = 400

b = 50

c = a + b

⇒ c = 450

String

a = "Павло "

b = "Зібров"

c = a + b

⇒ c = "Павло Зібров"

Integer and String

a = 450

b = "грн"

c = a + b

Числа, Текст и +

Integer

a = 400

b = 50

c = a + b

⇒ c = 450

String

a = "Павло "

b = "Зібров"

c = a + b

⇒ c = "Павло Зібров"

Integer and String

a = 450

b = "грн"

c = a + b

⇒ ERROR!!!!

Конвертация str - int - float

`str(5) → "5" # делает из текста число`

`int("5") → 5 # делает из текста целое число`

`float("5") → 5.0 # делает из текста число с запятой`

Tea and Pee



Условия



Дрожа от страха, Али-Баба проговорил: „Симсим, открой дверь“. – И дверь открылась.

Проверка значения с if и else

Логические операторы

Оператор	Название
==	равно
!=	не равно
>	больше
<	меньше
>=	больше или равно
<=	меньше или равно

```
3
4 money = 150
5 if money > 100:
6     print("You are rich!")
7 else:
8     print("You are poor!")
9
10
```

Проверка значения с if и else

```
3
4 money = 150
5 if money > 100:
6     print("You are rich!")
7 else:
8     print("You are poor!")
9
10
```

Блок if будет выполняться только если условие будет истинным / правдивым

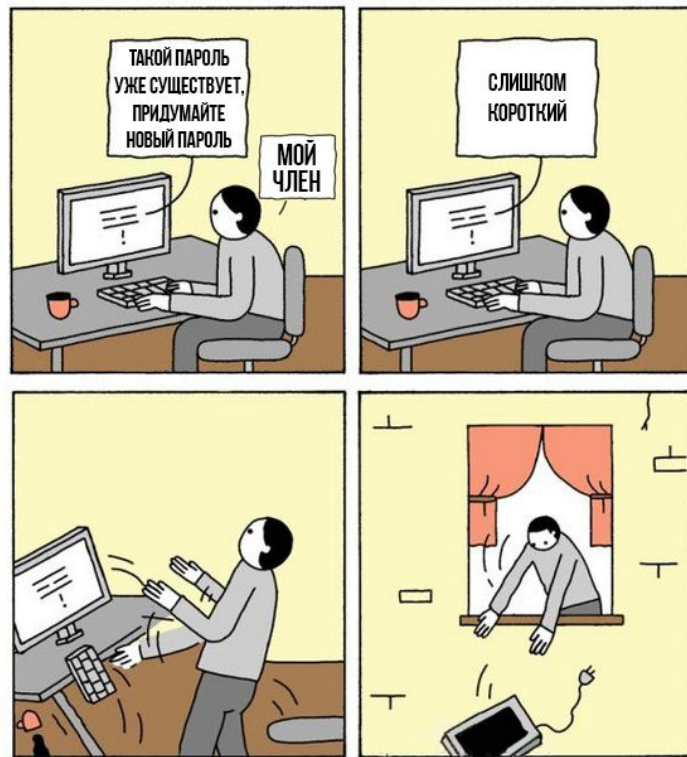
Часть else будет выполняться в любом другом случае

Выполняется только один из блоков !!!
money не может быть одновременно больше и меньше 100

Упражнение "Проверить пароль"

Напишите программу защищенную паролем:

- Придумайте пароль для программы и сохраните его в переменную
- При запуске программы пользователя просят ввести пароль
- Если он совпадает с сохраненным паролем Выведете на экран любое сообщение
- Если не совпадает: вывести "Неправильный пароль!!"



Упражнение с возрастом

Оператор	Название
==	равно
!=	не равно
>	больше
<	меньше
>=	больше или равно
<=	меньше или равно

- Попросите пользователя ввести возраст.
- Если возраст пользователя меньше 18 -> напишите, что мы не можем ему показать “секретное сообщение”.
- В другом случае -> покажите пользователю секретное сообщение.

Два условия должны выполняться

Вход

Логин:

Пароль:

☐ Запомнить меня [Забыли пароль?](#)

Войти

Ещё не зарегистрированы? [Регистрация](#)

Войти можно только когда **и** логин **и** пароль совпадают:

Для этого есть слово **and**

```
6
7 if user_name == "jack_sparrow" and password == "ILoveRun":
8     print("You are logged in!")
9 else:
10    print("Wrong user name or password")
11
```

Условие 1

Условие 2

Или / or

Представим, что у нас есть два любимых фрукта: **киви** и **банан**. Попросите пользователя ввести название любого фрукта.

- Если оно совпадает с одним из наших фруктов -> напишите, что пользователь угадал
- Если нет -> не угадал

```
1 fruit = input('Give me a fruit name')
2
3 if fruit=='banana' or fruit == 'kiwi':
4     print('You guessed right!')
5 else:
6     print('Wrong')
7
8
```


Упражнение с вопросом

Оператор	Название
==	равно
!=	не равно
>	больше
<	меньше
>=	больше или равно
<=	меньше или равно

- Задайте пользователю вопрос, на который можно ответить только “да” или “нет”
(пример: “ Ты меня уважаешь?”)
- Если пользователь ввел “да” или “нет” выведете любое сообщение (прям любое)
- В другом случае напишите, что ответ неправильный

На сегодня все!