

2



Python

встроенные функции

- `print('hello')` # hello
- `input('Enter your name: ')` # Enter your name _
- `bool(1)` # True
- `round(3.14159265359, 2)` # 3.14
- `list('abcd')` # ['a', 'b', 'd', 'c']
- `int('22')` # 22
- `len('abcd')` # 4

Логические выражения и операторы

Логическое выражение - конструкция языка программирования, результатом вычисления которой является «истина» (**True**) или «ложь» (**False**).

Операторы:

- == равно
- != не равно
- > больше
- < меньше
- >= больше или равно
- <= меньше или равно
- in находится в последовательности
- is ссылается на тот же объект

Логические операции

Над логическими выражениями возможны операции, результатом которых так же являются «истина» (True) или «ложь» (False).

Операции:

- **not** логическое **отрицание**

возвращает обратное значение:

- not 0 == True
- not 1 == False

- **and** логическое **И**

возвращает True, если все выражения True:

- 0 and 1 == False
- 1 and 1 == True
- 0 and 0 == False

- **or** логическое **ИЛИ**

возвращает True, если хотя бы одно выражение True:

- 0 or 1 == True
- 0 or 0 == False
- 1 or 1 == True

Логические операции

Приоритет логических операций без скобок:

1. **not**

2. **and**

3. **or**

`not True and True == False`

`True and False or True == True`

`not False and True or False == True`

`True and not False == True`

if

```
x = int(input("Please enter an integer: "))

if x < 0:
    x = 0
    print('Negative changed to zero')
elif x == 0:
    print('Zero')
elif x == 1:
    print('Single')
else:
    print('More')
```

Задача

1) Дано число. Если оно больше 10, то вывести его на консоль, если же оно меньше 5 но больше 0, то увеличить его на 11 и проверить стало ли оно больше 13 если стало вывести на консоль «стало больше 13» в противном случае вывести число после сложения, в остальных случаях уменьшить число на 100 и проверить что это число находится в промежутке между -200 и -50 то выводим этот промежуток и число иначе выводим просто число с текстом «Что то пошло на право».

2) Выведите на экран **n** раз фразу "Silence 'is' golden". Число **n** вводит пользователь. (циклы не используем)

Цикл while

Цикл с предусловием - пока условие верно, выполняется код из тела цикла.

```
1 x = 0
2 while x < 5:
3     print(x*x)
4     x += 1
5
```



Цикл for

Цикл обхода заданного множества элементов (символов строки, объектов списка или словаря) и выполнения в своем теле различных операций над ними.

```
1 words = ['cat', 'dog', 'horse']  
2  
3 for word in words:  
4     print(word, len(word))  
5
```

Функция range()

Функция, которая генерирует разнообразные последовательности чисел.

`range(5)` -> 0, 1, 2, 3, 4

`range(1, 6)` -> 1, 2, 3, 4, 5

`range(0, 7, 2)` -> 0, 2, 4, 6

`range(5, 0, -1)` -> 5, 4, 3, 2, 1

`range(-1, -5, -2)` -> -1, -3

Задачи

Вывести на экран с 1 по 20 строки. В строках с четными номерами вывести по 10 чисел, равных номеру строки. В строках с нечетными номерами вывести десять единиц.

Создать программу для проверки правильности ввода пароля пользователем. Для этого у пользователя есть 5 попыток на ошибку. На входе юзер вводит пароль, на выходе пользователю выводится количество его попыток и если он ввел правильно пароль выводим на экран «Доступ открыт», если пароль ввели не правильно выводим количество попыток и текст «попробуй еще», когда попытки закончились выводим текст с количеством попыток.

Даны числа ***h*** и ***m***, где ***h*** - количество часов, ***m*** - количество минут некоторого момента времени. Найдите угол между часовой и минутной стрелками в этот момент времени.

Git

Git - распределённая система контроля версий. <https://git-scm.com/download>

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий Git.

.gitignore - файл, указывающий какие папки/файлы не следует отслеживать (.idea, .рус, итп)

Полезные команды:

\$ **git init** (инициализировать репозиторий)

\$ **git status** (проверить статус)

\$ **git add filename** (добавить файлы для отслеживания)

\$ **git commit -m "commit message"** (сохранить версию кода)

\$ **git add remote origin** <https://github.com/repoacc/repo.git> (добавить соединение с удалённым репозиторием)

\$ **git remote -v** (проверить подсоединённые репозитории)

\$ **git push origin master** (отправить сохранённые изменения в репозиторий)

\$ **git pull origin master** (стянуть и слить код из репозитория с локальным кодом)

\$ **git clone <https://github.com/repoacc/repo.git>** (клонировать удалённый репозиторий и создать соединение)

Впишите ваше имя пользователя вместо **user_name** в данной команде:

```
git config --global user.name "user_name"
```

Далее впишите ваш адрес электронной почты вместо **email@domain.com**.

```
git config --global user.email "email@domain.com"
```

Задача на дом 3*

Условие: Дан список с положительными числами и число N. Написать функцию, которая находит N-ую степень элемента в массиве с индексом N. Если N за границами массива, тогда вернуть -1.

*Не забывайте, что первый элемент имеет индекс 0.

Примеры:

- список = [1, 2, 3, 4] и N = 2, тогда результат $3^{**2} == 9$;
- список = [1, 2, 3] и N = 3, но N за границами массива, так что результат -1.

Входные значения: Два аргумента. Список целых чисел и положительное целое число.

Выходные значения: Целое число.

```
index_power([1, 2, 3, 4], 2) == 9
index_power([1, 3, 10, 100], 3) == 1000000
index_power([0, 1], 0) == 1
index_power([1, 2], 3) == -1
```