Моя IT Школа

сертифицированные IT курсы

Строки

Задача №1: Лотерея

Реализуйте свою игру лотереи с выбором числа или набора чисел

Решение

Моя Школа

```python
import random

a = input("Введите название лотереи, которую хотите проверить:  ")

if a == "Суперлото":
    d = int(input("Введите 4 чисел Вашего билета: "))
    k = random.randint(1000, 9999)
    if d == k:
        print("Ваш билет выиграл")
    else:
        print("Попробуйте в другой раз!")
elif a == "Ваше лото":
    q = input("Введите Ваш номер телефона : ")
    w = int(input("Введите Ваш игровой номер из смс: "))
    l = random.randint(10, 999)
    if w == l:
        print("Ваш билет выиграл")
    else:
        print("Попробуйте в другой раз!")
```

Задача №2:

Человек вводит день и месяц своего рождения, выведите, кем он является по знаку зодиака и сколько ему сейчас лет

Решение

Моя Школа

```
a = int(input("Введите день Вашего рождения: "))
b = int(input("Введите номер месяца Вашего рождения: ")
if b == 1:
    if a >= 21:
        print("Вы водолей по знаку зодиака")
    else:
        print("Вы козерог по знаку зодиака")
elif b == 2:
    if a >= 20:
        print("Вы рыбы по знаку зодиака")
    else:
        print("Вы водолей по знаку зодиака")
elif b == 3:
    if a >= 21:
        print("Вы овен по знаку зодиака")
    else:
        print("Вы рыбы по знаку зодиака")
elif b == 4:
    if a >= 21:
        print("Вы телец по знаку зодиака")
    else:
        print("Вы овен по знаку зодиака")
elif b == 5:
    if a >= 22:
        print("Вы близнецы по знаку зодиака")
    else:
        print("Вы телец по знаку зодиака")
```

```
elif b == 6:
    if a >= 22:
        print("Вы рак по знаку зодиака")
    else:
        print("Вы близнецы по знаку зодиака")
elif b == 7:
    if a >= 23:
        print("Вы лев по знаку зодиака")
    else:
        print("Вы рак по знаку зодиака")
elif b == 8:
    if a >= 22:
        print("Вы дева по знаку зодиака")
    else:
        print("Вы лев по знаку зодиака")
elif b == 9:
    if a >= 24:
        print("Вы весы по знаку зодиака")
    else:
        print("Вы дева по знаку зодиака")
elif b == 10:
    if a >= 24:
        print("Вы скорпион по знаку зодиака")
    else:
        print("Вы весы по знаку зодиака")
elif b == 11:
    if a >= 23:
        print("Вы стрелец по знаку зодиака")
```

```
    else:
        print("Вы скорпион по знаку зодиака")
elif b == 12:
    if a >= 23:
        print("Вы козерог по знаку зодиака")
    else:
        print("Вы стрелец по знаку зодиака")
```

План занятия

Моя IT Школа
сертифицированные IT курсы

1. Строки. Их реализация в Python
2. Срезы. Подстроки
3. Встроенные функции строк
4. Форматированный вывод строк

Строки. Их реализация в Python

Моя Школа

Строки — упорядоченные неизменяемые последовательности символов, используемые для хранения и представления текстовой информации, поэтому с помощью строк можно работать со всем, что может быть представлено в текстовой форме.

Строки. Их реализация в Python

Моя Школа

Строки можно создать несколькими способами:

1. С помощью одинарных и двойных кавычек.
Например:

```
first_string = 'Я текст в одинарных кавычках'
second_string = "Я текст в двойных кавычках"
```

Строки в одинарных и двойных кавычках - одно и то же. Причина наличия двух вариантов в том, чтобы позволить вставлять в строки символы кавычек, не используя экранирование. Например вот так(обратите внимание на кавычки внутри строки):

```
first_string = 'Слово "Python" обычно подразумевает змею'
second_string = "I'm learning Python"
```

Строки. Их реализация в Python

Моя Школа

2. С помощью тройных кавычек.

Главное достоинство строк в тройных кавычках в том, что их можно использовать для записи многострочных блоков текста. Внутри такой строки возможно присутствие кавычек и апострофов, главное, чтобы не было трех кавычек подряд. Пример:

```
my_string = '''Это очень длинная строка, ей нужно много места''' '''
print(my_string)
```

```
Это очень длинная строка, ей нужно много места'''

Process finished with exit code 0
```

3. С помощью метода str().

Как это работает:

my_num = 12345

my_str = str(my_num)

В данном случае мы создали новую строку путем конвертации переменной другого типа (например, int).

Базовые операции

Моя IT Школа

1. Оператор сложения строк +

+ — оператор конкатенации строк. Он возвращает строку, состоящую из совокупности других строк.

Например:

```
a = 'Вот так работает'
b = ' конкатенация строк'
print(a + b)
```

```
st = 'Строка'
print(5 * st)
```

'Вот так работает конкатенация строк'

2. Оператор умножения строк *

* — оператор создает несколько копий строки. Если str это строка, а n целое число, то будет создано n копий строки str.

```
st = 'Строка'
print(len(st))
```

'СтрокаСтрокаСтрокаСтрокаСтрока’

3. Длина строки (функция len)

Задание №1

Напишите программу, которая запрашивает у пользователя его имя, а затем выводит строку «Привет, ...», где вместо многоточия имя пользователя. А вторая строка выведет имя пользователя с повтором 3 раза.

Решение

Моя Школа

```
1  s1 = input("Введите Ваше имя")
2  s2 = "Привет," + s1
3  s3 = s1*3
4  print(s2)
5  print(s3)
```

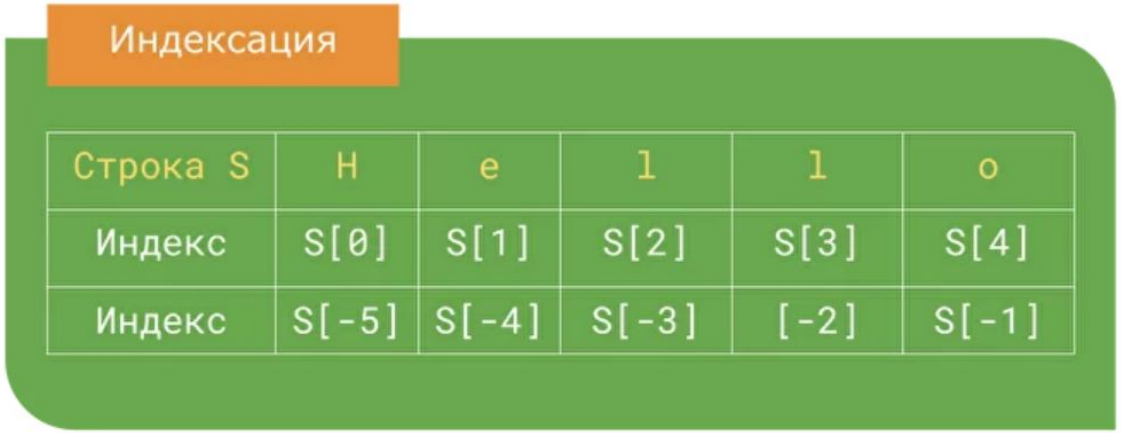
Срезы. Подстроки

Моя Школа

Срезы также относятся к группе общих операций - они используются для всех последовательностей, а значит и для строковых переменных.

Срез (slice) — извлечение из данной строки одного символа или некоторого фрагмента подстроки или подпоследовательности.

Индекс - номер символа в строке (а также в других структурах данных: списках, кортежах, массивах). Обратите внимание, что нумерация начинается с 0. Если указать отрицательное значение индекса, то номер будет отсчитываться с конца, начиная с номера -1.



Строки

Моя Школа

1. Самая простая форма среза - взятие одного символа строки - S[i], где S - строка, i - индекс.

```
s = 'Hello'
print(s[0], s[4], s[-5])
```

```
H o H

Process finished with exit code 0
```

2. Второй тип - срез с двумя параметрами. Т. е. S[a:b] возвращает

подстроку, начиная с символа с индексом a до символа с индексом b, не включая его. Если опустить второй параметр (но поставить двоеточие), то срез берется до конца строки.

```
s = 'Hello'

print(s[0:4], s[1:3], s[1:])
```

```
Hell el ello

Process finished with exit code 0
```

3. Срез с тремя параметрами - S[a:b:d]. Третий параметр задает шаг, то есть будут взяты символы с индексами a, a + d, a + 2 * d и т. д. Например, при задании значения третьего параметра, равному 2, в срез попадет каждый второй символ:

```
s = 'Hello'

print(s[0:5:1], s[::1], s[0:5:2], s[::2])
```

```
Hello Hello Hlo Hlo

Process finished with exit code 0
```

И еще раз: строки в Python - это **неизменяемый** тип данных!

Любые операции среза со строкой создают новые строки и никогда не меняют исходную строку. В Питоне строки вообще являются неизменяемыми, их невозможно изменить. Можно лишь в старую переменную присвоить новую строку

Задание №2

Вычислить сумму цифр случайного трёхзначного числа

Решение

Моя Школа

```python
import random

n = random.randint(100, 999)


print(n)

s = str(n)
a = int(s[0])
b = int(s[1])
c = int(s[2])

print(a + b + c)
```

Операции со строками. Дополнительные методы.

Далее давайте рассмотрим методы второй группы, которые были созданы специально для работы с данными типа **str**. Полный и актуальный список методов можно посмотреть на странице официальной документации. И как вы сможете заметить, их там немало. Мы же с вами перечислим самые полезные из них и популярные

Строки

Моя IT Школа

1. Работа с регистром строки	
s.capitalize()	Преобразует первую букву первого слова строки s в букву в верхнем регистре, все остальные буквы преобразуются в буквы в нижнем регистре.
s.title()	Преобразует первые буквы всех слов строки s в буквы верхнего регистра, все остальные буквы слов преобразует в буквы нижнего регистра.
s.upper()	Преобразует все буквы строки s в буквы верхнего регистра.
s.lower()	Преобразует все буквы строки s в буквы нижнего регистра.
s.swapcase()	Преобразует все буквы верхнего регистра в буквы нижнего регистра, а буквы нижнего регистра преобразует в буквы верхнего регистра.
s.isupper()	Возвращает True, если все символы строки, поддерживающие приведение к регистру, приведены к верхнему, иначе — False.
s.islower()	Возвращает True, если все символы строки, поддерживающие приведение к регистру, приведены к нижнему, иначе — False.
s.istitle()	Определяет, начинаются ли слова строки с заглавной буквы. Возвращает True, когда s не пустая строка и первый алфавитный символ каждого слова в верхнем регистре, а все остальные буквенные символы в каждом слове строчные. Иначе - False.

Строки

Моя Школа

```python
s = 'hello everybody'
s_2 = 'Hello Everybody'
s_3 = 'Hello everybody'
# Делаем строку заголовком
print(s.title())

# Начинаем строку с заглавной буквы
print(s.capitalize())

# Переводим строку в верхний регистр
print(s.upper())

# Переводим строку в нижний регистр
print(s.lower())

# Инверсия регистра
print(s_2.swapcase())

# Проверяем, являются ли строки заголовками
print(s.istitle())
print(s_2.istitle())
print(s_3.istitle())
```

Задание №3

На вход подается непустая строка S. В строке хотя бы два символа.
1) В первой строке распечатайте каждый 3-й символ, начиная с нулевого (подряд, не разделяя символы пробелами).
2) Во второй строке распечатайте первый и последний символы (подряд, не разделяя символы пробелами).
3) В третей строке распечатайте S в верхнем регистре.
4) В четвертой строке распечатайте S в обратном порядке.
5) В пятой строке напечатайте True, если все символы в строке S — цифры и False в противном случае.

Решение

Моя Школа

```
8   s = input()
9
10  print(s[::3])                                    # 1
11  print(s[:1] + s[-1:])                            # 2
12  print(s.upper())                                 # 3
13  print(s[::-1])                                   # 4
14  print(s.isdigit())                               # 5
15
```

Строки

Моя Школа

S.isdigit() - Состоит ли строка из цифр.
S.isalpha() - Состоит ли строка из букв.

```
print('A'.isdigit())


print('A'.isalpha())
```

```
False
True


Process finished with exit code 0
```

Строки

Моя Школа

2. Объединение и разбивка строк	
x.join(iterable)	Возвращает строку, собранную из элементов указанного объекта, поддерживающего итерирование(например, список строк).
s.split(x)	Разбивает строку s на части, используя специальный разделитель x, и возвращает эти части в виде списка.

```
print('..'.join(['a', 'b']))
print('1_2_3'.split('_'))
```

```
a..b
['1', '2', '3']

Process finished with exit code 0
```

Задание №4

Вводиться строка. Удалить из неё все пробелы. После этого определить, является ли она палиндромом (перевертышем), т.е. одинаково пишется, как сначала, так и с конца

Решение

Моя IT Школа

```python
s = input()
# Метод split разбивает строку на части списка
s = s.split()
# объединение списка строк с разделителем "пробел"
s = ''.join(s)
print(s)

# Выполняем срез и переворачиваем слово
if s == s[::-1]:
    print("Палиндром")
else:
    print("Не палиндром")
```

```python
s = input()
n_s = s.replace(' ', '')
# Выполняем срез и переворачиваем слово
if n_s == n_s[::-1]:
    print("Палиндром")
else:
    print("Не палиндром")
```

Строки

Моя IT Школа

3. Поиск и замена внутри строки	
s.startswith(prefix)	Возвращает True, если строка s начинается с указанного префикса, иначе - False.
s.endswith(suffix)	Возвращает True, если строка s оканчивается указанным постфиксом, иначе - False.
s.find(sub)	Находит в строке s подстроку sub. Возвращает индекс первого вхождения искомой подстроки. Если же подстрока не найдена, то метод возвращает значение -1.
s.replace(old, new)	Заменяет в строке s все вхождения подстроки old на подстроку new.

Функции поиска

Моя IT Школа

В Python существует две похожих функции для поиска подстроки - **find()**(относится к группе строковых методов) и **index()**. Разница в том, что **find()** вернет **-1**, если не найдет искомое, а **index()** выкинет исключение **ValueError**.

Кроме того, необходимо помнить, что если нужно только удостовериться, что подстрока присутствует внутри строки - можно просто воспользоваться методом **in**:

```
'Py' in 'Python’  # True
```

Задание №5

Дана строка, состоящая ровно из двух слов, разделенных пробелом.
Переставьте эти слова местами. Результат запишите в строку и
выведите получившуюся строку.
Замените в этой строке все цифры 1 на слово one

Примечание: решить без использования функции split

Решение

Моя Школа

```
# Дана строка, состоящая ровно из двух слов, разделенных пробелом.
# Переставьте эти слова местами. Результат запишите в строку и
# выведите получившуюся строку.
# Замените в этой строке все цифры 1 на слово one

s = input()
first_word = s[:s.find(' ')]
second_word = s[s.find(' ') + 1:]
k = second_word + ' ' + first_word
print(k)
print(k.replace('1', 'one'))

```

Домашнее задание

Моя Школа

Задача №1

Дана строка.

Сначала выведите третий символ этой строки.

Во второй строке выведите предпоследний символ этой строки.

В третьей строке выведите первые пять символов этой строки.

В четвертой строке выведите всю строку, кроме последних двух символов.

В пятой строке выведите все символы с четными индексами (считая, что индексация начинается с 0, поэтому символы выводятся начиная с первого).

В шестой строке выведите все символы с нечетными индексами, то есть начиная со второго символа строки.

В седьмой строке выведите все символы в обратном порядке.

В восьмой строке выведите все символы строки через один в обратном порядке, начиная с последнего.

В девятой строке выведите длину данной строки.