

## Семинар 3-4

Вещественные числа  
Строки

# Вещественные числа

**Хранение float:** 64 бита

1 - знак

52 - мантисса

11 - порядок (экспонента)

**Запись:** 123.123 или 1.23123e2

**Ввод:** `float(input())`    # 0.1

**Вывод:**

```
a = 0.3
print(a)
print('{0:.20}'.format(a))
```

```
a = 0.1
print(a)
print('{0:.20}'.format(a))
```

# Сравнение вещественных чисел

```
x = float(input())  
y = float(input())  
z = float(input())
```

```
if x + y == z:  
    print('Right')  
else:  
    print('Not right')
```

```
# 0.1  
# 0.2  
# 0.3  
# Not right
```

```
x = float(input())  
y = float(input())  
z = float(input())
```

```
eps = 1.0 * 10 ** -5;  
  
if abs(x + y - z) < eps:  
    print('Right')  
else:  
    print('Not right')
```

```
# 0.1  
# 0.2  
# 0.3  
# Right
```

# Округление вещественных чисел

**int** - отбрасывает дробную часть

**round** - округляет до ближайшего целого (если 0.5, то до четного)

**floor** - округляет в меньшую сторону

**ceil** - округляет в большую сторону

Библиотека **math**: <https://docs.python.org/3/library/math.html>

**from** math **import** floor, ceil

```
print(int(1.5))
```

```
print(round(1.5))
```

```
print(round(2.5))
```

```
print(floor(1.5))
```

```
print(ceil(1.5))
```

```
print(int(-1.5))
```

```
print(round(-1.5))
```

```
print(round(-2.5))
```

```
print(floor(-1.5))
```

```
print(ceil(-1.5))
```

# Строки и символы

**Строка** - последовательность символов, заключенная в кавычки (" или ').

**Срез** - подстрока из строки, содержащая отдельные символы или подстроки. При использовании среза создается новая строка, исходная строка остается **без изменений**.

```
str1 = "Hello!"    str1[1:5] # ello!    str1[::-1] # Hello!
str1[0] # H         str1[1:-1] # ello     str1[::-2] # Hlo
str1[1] # e         str1[1:]  # ello!    str1[::-3] # Hl
str1[5] # !         str1[: -1] # Hello   str1[::-1] # !olleH
len(str1) # 6      str1[: -2] # !le
str1[-1] # !        str1[1:5:-1] # "" - начало и конец
str1[-2] # o        должны быть наоборот
str1[-6] # H        str1[5:2:-1] # !ol
```

# Задача 1

Заменить в самом длинном слове строки буквы 'а' на 'б'.

```
text = "Строкаааа для знаний о языке"
```

```
maxWord = ""
```

```
i = 0
```

```
maxLengthOfWord = 0
```

```
while (i < len(text)):
```

```
    word = ""
```

```
    while (i < len(text) and text[i] != ' '):
```

```
        word += text[i]
```

```
        i += 1
```

```
    if (len(maxWord) < len(word)):
```

```
        maxWord = word
```

```
    i += 1
```

```
print(maxWord)
```

# Задача 1

```
i = 0
wordMaxNew = ""
while (i < len(maxWord)):
    if (maxWord[i] == 'a'):
        # нельзя word[i] = 'б'
        wordMaxNew += "б"
    else:
        wordMaxNew += maxWord[i]
    i += 1

print(wordMaxNew)
```

# Задача 1 (Самостоятельно)

Заменить выделение слов на срезы в стиле Python.



# Методы для работы со строками

**Методы** - это функции, которыми "владеют" объекты и применяемые в контексте объекта.

**Метод** вызывается с помощью записи `myObj.methodName(параметры)`.

**Методы** позволяют организовывать хранение и обработку данных в объектах, например, строках.

```
text = "Мой дядя самых честных правил.."
text.find("дядя") # 4 - позиция самого левого входжения подстроки в строку
text.rfind("я") # 7 - позиция самого правого входжения подстроки в строку
text.find("дядя", 5) # -1 - не найдена подстрока, если начинать с 5й позиции
```

```
print(text.replace("дядя", "тетя")) # Мой тетя самых честных правил..
print(text.replace("Вася", "Петя")) # ???
```

```
text = "   ааа   "
text.lstrip()
text.rstrip()
text.strip()
text = "___аааа___"
print(text.strip('_')) # аааа
```

## Задача 2 “Замена подстроки”

Дана строка. Замените в этой строке все цифры 1 на слово one. 2 - two.. 9 - nine, 0 - zero.

```
text = "1 2 3 numbers 2 1 3 5"
text = text.replace("1", "one")
text = text.replace("2", "two")
text = text.replace("3", "three")
text = text.replace("4", "four")
text = text.replace("5", "five")
text = text.replace("6", "six")
text = text.replace("7", "seven")
text = text.replace("8", "eight")
text = text.replace("9", "nine")
text = text.replace("0", "zero")
print(text)
```

*# "\" - так можно говорить интерпретатору, что команда будет продолжена*

```
text = text.replace("1", "one").replace("2", "two") \
    .replace("3", "three").replace("4", "four").replace("5", "five") \
    .replace("6", "six").replace("7", "seven").replace("8", "eight").replace("9", "nine").replace("0", "zero")
```

# Задача 3

Дана строка из слов, разделенных пробелами. Заменить последние три символа каждого слова на "\*\*\*".

```
inputStr = input("Введите слово: ")
res = inputStr[:-3] + "***";
print (res)
```

```
inputStr = input("Введите строку: ")
res = ""
listOfWords = inputStr.split(' ') # список из исходной строки, на основе разделителя.
print (listOfWords)
```

```
for str1 in listOfWords: # применение цикла for - "для каждого" элемента массива
    res += ' ' + str1[:-3] + "***"
print(res)
```

## Задача 4 (Самостоятельно)

Вводится слово и ширина поля (в количестве символов), по которому нужно выровнять его по центру. Получить и вывести строку, в которой введенное слово выровнено по центру.

Например:

Введите слово: Питон

Введите ширину поля: 25

Питон

Т.е. входная строка: “Питон”, выходная - “Питон”.