

# Языки программирования

## Лекция 4

# Модуль random. Генерация случайных чисел

- Генератор псевдослучайных чисел (ГПСЧ, англ. pseudorandom number generator, PRNG) — алгоритм, порождающий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению (обычно равномерному).
- *«Генерация случайных чисел слишком важна, чтобы оставлять её на волю случая»* — Роберт Кавью
- `import random`

# Модуль random. Генерация случайных чисел

- `random()` – возвращает псевдослучайное число от 0 до 1:
- `random.random()`
- 0.6247985122932341
- 
- `seed()` – настраивает генератор случайных чисел на новую последовательность. Если параметр не указан, в качестве базы для случайных чисел будет использовано системное время. Если значение параметра будет одинаковым, то генерируется одинаковое число:
- `random.seed(2); random.random(); random.seed(2); random.random(); random.random()`
- 0.9560342718892494
- 0.9560342718892494
- 0.9478274870593494

# Модуль random. Генерация случайных чисел

- `uniform(<Начало>,<Конец>)` – возвращает псевдослучайное вещественное число в диапазоне от <Начало> до <Конец>:
- `random.uniform(1,10)`
- 1.2696044276812932
- `randint(<Начало>,<Конец>)` – возвращает псевдослучайное целое число в диапазоне от <Начало> до <Конец>:
- `random.randint(1,10)`
- 5
- 
- `randrange([< Начало>, ]<Конец> [, <Шаг>])` – возвращает случайный элемент из числовой последовательности:
- `random.randrange(3),random.randrange(0,5),random.randrange(1,10,3)`
- 1,3,7

# Модуль random. Генерация случайных чисел

- `choice(<Последовательность>)` – возвращает случайный элемент из заданной последовательности (строки, списка, кортежа):
  - `random.choice("hello")`
  - `e`
- `shuffle(<Список>)` – перемешивает элементы списка случайным образом:
  - `arr = [1,2,3,4,5]`
  - `random.shuffle(arr)`
  - `arr`
  - `[1, 4, 2, 3, 5]`
  - `random.shuffle(arr)`
  - `arr`
  - `[3,1, 2, 5, 4]`

# Модуль random. Генерация случайных чисел

- `sample(<Последовательность>, <Количество элементов>)` – возвращает список из указанного количества элементов, которые будут выбраны случайным образом из заданной последовательности:
- `random.sample("hello",3)`
- `['h', 'o', 'l']`

# Генератор паролей

```
import random # Подключаем модуль random
def passw_generator(count_char=8):
    arr = ['a','b','c','d','e','f','g','h','i','j','k','l','m',
           'n','o','p','q','r','s','t','u','v','w','x','y','z',
           'A','B','C','D','E','F','G','H','I','J','K','L',
           'M','N','O','P','Q','R','S','T','U','V','W',
           'X','Y','Z','1','2','3','4','5','6','7','8','9','0']
    passw = []
    for i in range(count_char):
        passw.append(random.choice(arr))
    return "".join(passw)

# Вызываем функцию
print( passw_generator(10) ) # Выведет что-то вроде ngODHE8J8x
print( passw_generator() )  # Выведет что-то вроде ZxcprkF50
```

# Работа со строками в Python

- строковые типы:
  - `str` – Unicode-строка,
  - `bytes` – неизменяемая последовательность байтов,
  - `bytearray` – изменяемая последовательность байтов.
- 
- `S = "Python"`
  - `S[0]`
  - `'P'`
  - `S[0] = "j"`
  - Traceback (most recent call last):
  - `TypeError: 'str' object does not support item assignment`

# Литералы строк

- `int a1 = 1;`
- `int c = a1;`
- `string cat = "Кот";`
- Экранированные последовательности - служебные символы

Экранированная последовательность	Назначение
<code>\n</code>	Перевод строки
<code>\t</code>	Знак табуляции
<code>\\</code>	Обратный слеш
<code>\'</code>	Апостроф
<code>\"</code>	Кавычка

# Создание строки

- 1) с помощью функции `str()`
- `srt()`, `str([1,2])`, `str((3,4))`
- `("", '[1,2]','(3,4)')`
- 2) указав строку между апострофами или двойными кавычками
- `'строка'`, `"строка"`,
- `print('Строка1\nСтрока2')` # апострофы
- Строка1
- Строка2
- `print("Строка1\nСтрока2")` # кавычки
- Строка1
- Строка2

# “Сырые” строки

- `S = r'C:\newt.txt'`
- `print(r"C:\newt.txt")`
- `C:\newt.txt`
  
- `print("C:\newt.txt")`
- `C:`
- `ewt.txt`
  
- `print("C:\\newt.txt")`
- `C:\newt.txt`

# “Сырые” строки

- `print(r"C:\newt.txt\")`
- `SyntaxError: EOL while scanning string literal`
  
- `print(r"C:\newt.txt\\")`
- `C:\newt.txt\`
  
- `print(r"C:\newt.txt" + "\\")` # конкатенация
- `C:\newt.txt\`
- `print("C:\\newt.txt\\")` # обычная строка
- `C:\newt.txt\`
- `print(r"C:\newt.txt\\"[:-1])` # Удаление слэша
- `C:\newt.txt\`

# Операции над строками

- `s = "Python"`
- `s[0],s[1]`
- `('P', 'y')`
  
- `s[10]`
- `IndexError: string index out of range`
  
- `s[-1], s[len(s)-1]`
- `('n', 'n')`

# Срезы

- [`<Начало>`:`<Конец>`:`<Шаг>`]
- `s = "Python"`
- Получим копию строки:
- `s[:]` # фрагмент от позиции 0 до конца строки
- `'Python'`
  
- Выведем символы в обратном порядке:
- `s[::-1]` # отрицательное значение в параметре `<Шаг>`
- `'nohtyP'`
  
- Заменяем первый символ в строке:
- `"J" + s[1:]` # фрагмент от символа 1 до конца строки
- `'Jython'`

# Срезы

- Удалим последний символ:
- `s[:-1]` # фрагмент от 0 до `len(s)-1`
- `'Pytho'`
  
- Получим первый символ в строке:
- `s[0:1]` # Символ с индексом 1 не входит в диапазон
- `'P'`
  
- Получим последний символ:
- `s[-1:]` # фрагмент от `len(s)-1` до конца строки
- `'n'`

# Срезы

- Выведем символы с индексами 2,3 и 4:
- `s[2:5]`
- `'tho'`
  
- Узнать количество символов в строке:
- `len("Python"), len("\n\t"), len(r"\n\n")`
- `(6,2,4)`

# Операции над строками

- Перебрать все символы строки:
- `s = "Python"`
- `for i in range(len(s)): print(s[i], end = " ")`
- `P y t h o n`
  
- `for i in s: print(i, end = " ")`
- `P y t h o n`
  
- `print("строка1" + "строка2")`
- `строка1строка2`
  
- `print("строка1" "строка2")`
- `строка1строка2`

# Операции над строками

- `"string" + str(10)`
- `'string10'`
  
- `"_" * 5`
- `'-----'`
  
- `"yt" in "Python"`
- `True`
  
- `"yt" in "Perl"`
- `False`
  
- `"PHP" not in "Python"`
- `True`

# Функции и методы для работы со строками

- Удаление указанных символов в начале и в конце строки:
  - `s = "strstrstrokstrstrstr"`
  - `s.strip("tsr")`
  - `'ok'`
- Разбиение строки по разделителю. Если разделитель не указан, то используется символ пробела:
  - `s = "str1 str2 str3"`
  - `s.split()`
  - `['str1','str2','str3']`

# Функции и методы для работы со строками

- `S.join(<Список>)` - Сборка строки из списка с разделителем `S`
- `" => ".join(["str1", "str2", "str3"])`
- `'str1 => str2 => str3'`
  
- `S.find(str,[start],[end])` - поиск подстроки в строке. Возвращает номер первого вхождения или `-1`. Метод зависит от регистра
- `s = "string1string2String3"`
- `s.find("str"), s.find("Str"), s.find("123")`
- `(0,14,-1)`

# Функции и методы для работы со строками

- `S.replace(шаблон,замена)` - замена шаблона
- `s = "Hello, Tom"`
- `s.replace("Tom","Bob")`
- `'Hello, Bob'`

# Функции и методы для работы со строками

- `S.capitalize()` - Переводит первый символ строки в верхний регистр, а все остальные в нижний
- `S.swapcase()` - Переводит символы нижнего регистра в верхний, а верхнего – в нижний
- `S.title()` - Первую букву каждого слова переводит в верхний регистр, а все остальные в нижний
- `S.upper()` - Преобразование строки к верхнему регистру
- `S.lower()` - Преобразование строки к нижнему регистру
- `S.isupper()` - Состоит ли строка из символов в верхнем регистре
- `S.islower()` - Состоит ли строка из символов в нижнем регистре
- `S.lstrip()` - Удаление пробельных символов в начале строки
- `S.rstrip()` - Удаление пробельных символов в конце строки
- `S.strip()` - Удаление пробельных символов в начале и в конце строки
- `S.strip(<Символы>)` - Удаление указанных символов в начале и в конце строки

# Функции и методы для работы со строками

- `S.isdigit()` Состоит ли строка из цифр
- `S.isalpha()` Состоит ли строка из букв
- `S.isalnum()` Состоит ли строка из цифр или букв
- `S.join(<Список>)` Сборка строки из списка с разделителем `S`

# Упражнения и контрольные вопросы

- 1. Что представляют собой строки в Python?
- 2. Какая последовательность символ предназначена для знака табуляции?
- 3. Какая последовательность символ предназначена для перевода строки?
- 4. Какой результат получим после выполнения следующего кода:
  - `s = 'String'`
  - `s[:-1]`
- 5. Какой результат получим после выполнения следующего кода:
  - `s = 'String'`
  - `J + s[1:]`

# Упражнения и контрольные вопросы

- 6. Какой результат получим после выполнения следующего кода:
  - `s = 'String'`
  - `s[0:1]`
- 7. Какой результат получим после выполнения следующего кода:
  - `s = 'String'`
  - `s[-1:]`
- 8. Какой результат получим после выполнения следующего кода:
  - `s = 'String'`
  - `s[2:5]`
- 9. Какой результат получим после выполнения следующего кода:
  - `len("\n\t")`
- 10. Какой результат получим после выполнения следующего кода:
  - `print("string1" "string2")`
- 11. Какой результат получим после выполнения следующего кода:
  - `s = "str1 str2 str3"`
  - `s.title()`