МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное бюджетное образовательное учреждение**

**науки и высшего образования**

**К Г Э У «КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»**

# (ФГБОУ ВО «КГЭУ»)

**Кафедра Информатики и информационных управляющих систем**

**ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 5  
 дисциплина «Алгоритмизация и программирование»**

|  |  |
| --- | --- |
| **Выполнил:** | Соловьёв Л.А. |
| **Группа:** | ПИ-1-22 |
| **Проверила:** | Шорина Т.В. |
|  |  |

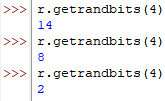
Казань, 2023

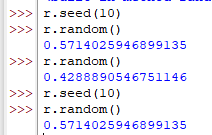
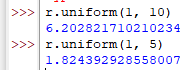
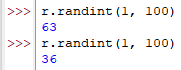
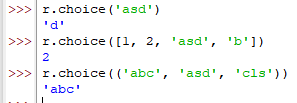
**random**

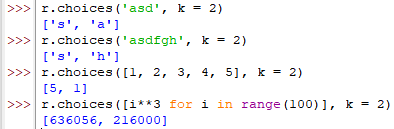
**Постановка задачи.** Изучение работы функций библиотеки random.  
  
Random – это расширение языка Python, позволяющее генерировать псевдослучайные числа

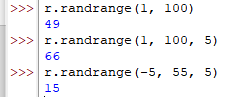
**Основные команды модуля random**

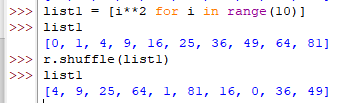
|  |  |
| --- | --- |
| random.random() | Возврат псевдослучайного числа от 0.0 до 1.0 |
| random.getstate() | Возвращает текущее внутреннее состояние генератора псевдослучайных чисел |
| random.setstate() | Восстанавливает внутреннее состояние генератора псевдослучайных чисел |
| random.getrandbits(k) | Возвращает число, которое можно представить из k битов |
| random.seed([seed\_value]) | Настраивает генератор псевдослучайных чисел на новую последовательность(по умолчанию – сис. время) |
| random.uniform(start, end) | Возращает псевдослучайное вещ. число в диапазоне от start до end |
| random.randint(start, end) | Возращает псевдослучайное целое число в диапазоне от start до end |
| random.choice(sequence) | Возвращает случайный элемент из любой последовательности(любой итерируемый объект) |
| random.choices(sequence, k) | Возвращает список из k случайных элементов любой последовательности(любой итерируемый объект) |
| .random.randrange(start, end[, step]) | Возвращает случайное целое число из последовательности |
| random.shuffle(list) | Перемешивает последовательность |
| random.sample(population, k) | Возвращает список состоящий из k элементов последовательности population(k <= len(population)) |
| random.triangular([low, high, mode]) | Возвращает случайное вещ. число из треугольного вероятностного распределения(по умолчанию: low = 0, high = 1, mode = 0.5) |
| random.betavariate(alpha, beta) | Возвращает случайное вещ. число из бета-распределения |
| random.expovariate(lambd) | Возвращает случайное вещ. число из эскпоненциального распределения |
| random.gammavaritate(alpha, beta) | Возвращает случайное вещ. число из гамма-распределения |
| random.gauss(mu, sigma) | Возвращает случайное вещ. число из распределения Гаусса |
| random.normalvariate(mu, sigma) | Возвращает случайное вещ. число из нормального распределения |
| random.lognormvariate(mu, sigma) | Возвращает случайное вещ. число из логнормального распределения |
| random.vonmisesvariate(mu, kappa) | Возвращает случайный угол из интервала [0, 2\*pi] по закону распределения фон Мизеса  mu – средний угол в радианах из интервала выше, kappa >= 0 |
| random.paretovariate(alpha) | Возвращает случайное вещ. число из распределения Парето |
| random.weibullvariate(alpha, beta) | Возвращает случайное вещ. число из распределения Вейбулла |

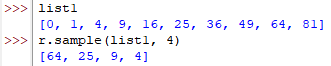
1.­­­­­ Возврат случайного числа в диапазоне от 0.0 до 1.0  
  
2. Возврат случайного числа, которое состоит из k битов  
  
(14 = 1110, 8 = 1000, 2 = 0010)

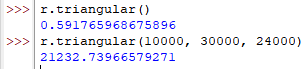
  
3. Настройка генератора на определенную последовательность  
  
4. Возврат псевдослучайного вещественного числа в диапазоне от start до end  
  
5. Возврат псевдослучайного целого числа в диапазоне от start до end  
  
6. Возврат случайного элемента из последовательности  


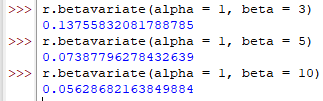
7. Возврат k случайных элементов из последовательности  


8. Возврат случайного целого числа из последовательности  


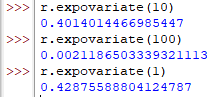
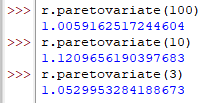
9. Перемешивание списка  
  
10. Возврат списка, состоящего из k элементов исходной последовательности

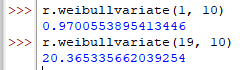


11. Возврат случайного вещ. числа из треугольного вероятностного распределения  


12. Возврат случайного вещ. числа из бета-распределения  




13. Возврат случайного вещ. числа из эскпоненциального распределения  
  
  
14. Возврат случайного вещ. числа из распределения Парето  


15. Возврат случайного вещ. числа из распределения Вейбулла  


**re**

**Постановка задачи.** Изучение работы функций библиотеки re.  
  
Re – это расширение языка Python, позволяющее работать с регулярными выражениями

**Основные команды модуля re**

|  |  |
| --- | --- |
| re.search(pattern, string) | Находит в строке string первую подстроку, подходящую под шаблон pattern |
| re.fullmatch(pattern, string) | Проверяет, подходит ли строка string под шаблон pattern |
| re.split(pattern, string, maxsplit=0) | Аналог метода split(), но разделение происходит по подстрокам, подходящим под шаблон pattern |
| re.findall(pattern, string) | Находит в строке string все непересекающиеся шаблоны pattern |
| re**.**finditer(pattern, string) | Итератор всех непересекающихся шаблонов pattern в строке string(выдаёт match-объект) |
| re.sub(pattern, repl, string, count=0) | Заменяет в строке string все непересекающиеся шаблоны pattern на repl |
| re.compile(pattern, flags=0) | Компилирует шаблон регулярного выражения в объект регулярного выражения, к которому применимы функции match(), search() и т.д. |
| re.sub(pattern, repl, string, count=0, flags=0) | Возвращает строку, полученную заменой крайних левых неперекрывающихся вхождений pattern в string с заменой repl. Если шаблон не найден, string возвращается без изменений |
| re.subn(pattern, repl, string, count=0, flags=0) | Аналогичен sub(), но возвращает кортеэ (new\_string, number\_of\_subs\_made) |
| re.escape(pattern) | Экранирование специальных символов в pattern |
| re.purge() | Очищает кэш регулярных выражений |
| re.error(msg, pattern=None, pos=None) | Вызывает исключение, если переданная строка одной из функций не является допустимым регулярным выражением msg – неформатированное сообщение об ошибке pattern – шаблон регулярного выражения pos – индекс в pattern, в котором произошла ошибка |
| [match-object].group([index of parenthesized subgroup]) | Возвращает одну или несколько подгрупп совпадения 0 – всё совпадение, 1 – первая подгруппа, 2 – вторая подгруппа |
| [match-object].groups(default=None) | Возвращает кортеж, содержащия все подгруппы совпадения, от 1 до количества групп в шаблоне  default – используется для групп, которые не участвовали в совпадении(None – по умолчанию) |
| [match-object].groupdict(default=None)­­­­­­ | Возвращает словарь, содержащий все именуемые подгруппы совпадения, с ключом по имени подгруппы |
| [match-object].\_\_getitem\_\_ или [match-object][index] | Идентичен [match-object].group([index of parenthesized subgroup]), более упрощенный |
| [match-object].start([group]) | Возвращают индексы начала и концы подстроки, соответствующей group |
| [match-object].end([group]) |
| [match-object].span([group]) | Возвращает кортеж ([match-object].start([group]), [match-object].end([group])) |

1. Компилируем шаблон регулярного выражения в объект регулярного выражения

  
  
2. Применяем объект регулярного выражения для нахождения соответствия в строке(получаем match-object)  


  
3. Получаем первый элемент соответствия с помощью .\_\_getitem\_\_  
  
4. Находим индекс начала полученного соответствия  
  
5. Находим индекс следующего элемента после конечного в полученном соответствии  
  
6. Получаем кортеж (Индекс первого элемента, индекс конечного элемента + 1)  


7. Получаем список всех подстрок, соответствующих pattern

  
8. Получаем итератор для всех найденных соответствий + получение начальных индексов с помощью .start()



9. Замена с помощью sub  


10. Получение списка всех совпадений с pattern

