

Outline for March 1

Reading: *text*, §9.2

Assignments: Homework 4, due on March 4 at 11:55pm

1. “Raw” string notation: backslash not handled specially; put “r” before string
2. Useful functions/methods [*recomp.py*, *renocomp.py*, *regroup.py*]
 - a. `re.compile(str)` compiles the pattern into *pc* (that is, `pc = re.compile(str)`)
 - b. `pc.match(str)` returns None if compiled pattern *pc* does not match beginning of string *str*
 - c. `pc.search(str)` returns None if pattern *pc* does not match any part of string *str*
 - d. `pc.findall(str)` returns a list of substrings of the string *str* that match the pattern *pc*
 - e. `pc.group(str)` returns the substring of the string *str* that the pattern *pc* matches
 - f. `pc.start(str)` returns the starting position of the match
 - g. `pc.end(str)` returns the ending position of the match
 - h. `pc.span(str)` returns tuple (start, end) positions of match
3. Useful abbreviations
 - a. `\d` matches any digit; same as `[0-9]`
 - b. `\s` matches any space character; same as `[\t\n\r\f\v]`
 - c. `\w` matches any alphanumeric character and underscore; same as `[a-zA-Z0-9_]`
 - d. `\D` matches any character *except* a digit; inverse of `\d`
 - e. `\S` matches any character *except* a space character; inverse of `\s`
 - f. `\W` matches any character *except* an alphanumeric character or underscore; inverse of `\w`
 - g. `\b` matches a word boundary — a word is a sequence of alphanumeric characters