|  |  |
| --- | --- |
| a = b + c | concatenate b and c |
| a = b \* c | b repeated c times |
| a[0] | the first character of a |
| len(a) | the number of characters in a |
| min(a) | the smallest element in a (alphabetically first) |
| max(a) | the largest element in a (alphabetically last) |

|  |  |
| --- | --- |
| zoo = ["monkey", "tiger", "eagle"] | # define a list |
| zoo[0] | # a single element |
| zoo[0] = "gorilla" | # change an element |
| zoo.append("parrot") | # add elements at end of list |
| zoo = ["zebra","lion"] + zoo | # add elements at beginning |

|  |  |
| --- | --- |
| del zoo[0] | # delete an element by index number |
| zoo.remove("tiger") | # delete an element by value |
| zoo.insert(1, "elephant") | # insert an element |
| len(zoo) | # length of zoo |
| max(zoo) | # alphabetically first or smallest element |
| min(zoo) | # alphabetically last or largest element |
| new\_zoo = zoo[:] | # create a new copy of zoo |
| zoo.reverse() | # change the list to reverse order |
| zoo.sort() | # change the list to alphabetical order |

|  |  |
| --- | --- |
| file = open("alice.txt","r") | open for reading |
| file = open("output.txt","w") | open for writing |
| file = open("output.txt","a") | open for appending |
| file.close() | close file |
| text = file.readlines() | read all lines into a list called "text" |
| line = file.readline() | read next line into a string called "line" |
| file.writelines(text) | write list to file |
| file.write(line) | write one line to file |

**Special Characters**

|  |  |
| --- | --- |
| . | Any single character except a newline |
| ^ | The beginning of the line or string |
| $ | The end of the line or string |
| \* | Zero or more of the last character |
| + | One or more of the last character |
| ? | Zero or one of the last character |
| {5,10} | Five to ten times the previous character |
|  | for example: \* equals {0, }; + equals {1, } |
|  | ? equals {0,1} |

|  |  |
| --- | --- |
| [qjk] | Either q or j or k |
| [^qjk] | Neither q nor j nor k |
| [a-z] | Anything from a to z inclusive |
| [^a-z] | No lower case letters |
| [a-zA-Z] | Any letter |
| [a-z]+ | Any non-zero sequence of lower case letters |
| jelly|cream | Either jelly or cream |
| (eg|le)gs | Either eggs or legs |
| (da)+ | Either da or dada or dadada or... |
| \n | A newline |
| \t | A tab |
| \w | Any alphanumeric (word) character. |
|  | The same as [a-zA-Z0-9\_] |
| \W | Any non-word character. |
|  | The same as [^a-zA-Z0-9\_] |
| \d | Any digit. The same as [0-9] |
| \D | Any non-digit. The same as [^0-9] |
| \s | Any whitespace character: space, |
|  | tab, newline, etc |
| \S | Any non-whitespace character |
| \b | A word boundary, outside [] only |
| \B | No word boundary |

**Escapes for special characters**

|  |  |
| --- | --- |
| \| | Vertical bar |
| \[ | An open square bracket |
| \) | A closing parenthesis |
| \\* | An asterisk |
| \^ | A carat symbol |
| \/ | A slash |
| \\ | A backslash |