1. sequence

string: ‘xyz’/”xyz” | list: [x, y, z] | tuple: (x, y, z)

2. sequence operator

seq[ind] | seq[start:end] | seq[start:end:step] | seq*n | seq1+seq2 | len(seq) | obj in/not in seq

3. list comprehension

[expr for iter_var in iterable if cond_expr]

4. mapping

dict: {a:b, c:d} | dict_var = dict(([a, b], [c, d]))

5. mapping operator

dict[key] | key in/not in dict | len(dict) | dict.keys() | dict.values() | dict.items()

6. command line parameter

import sys | sys.argv | len(sys.argv) | sys.argv[0] (program name)

7. function

def func(param, param = value, *tuple_param, **dict_param):

func(var, var = value, *tuple_var, **dict_var)

expand var to value then assign to param

8. package.module

print sys.path | export PYTHONPATH = ‘/path’ | sys.path.append(‘/path’)

print sys.modules (indicate which modules imported)

module path: LIB/package/module.py

import package.module *as alias*

from package.module import name *as alias*

9. OOP

class class_name(base_name):

```
object = class_name()
```

```
self.attribute | self.method
```

10. childprocess

```
os.system(‘cmd’)
```

11. IO

```
raw_input(‘PlS INPUT:’) | print a, b, c | ‘%s is number %d’ % (‘python’, 1)
```