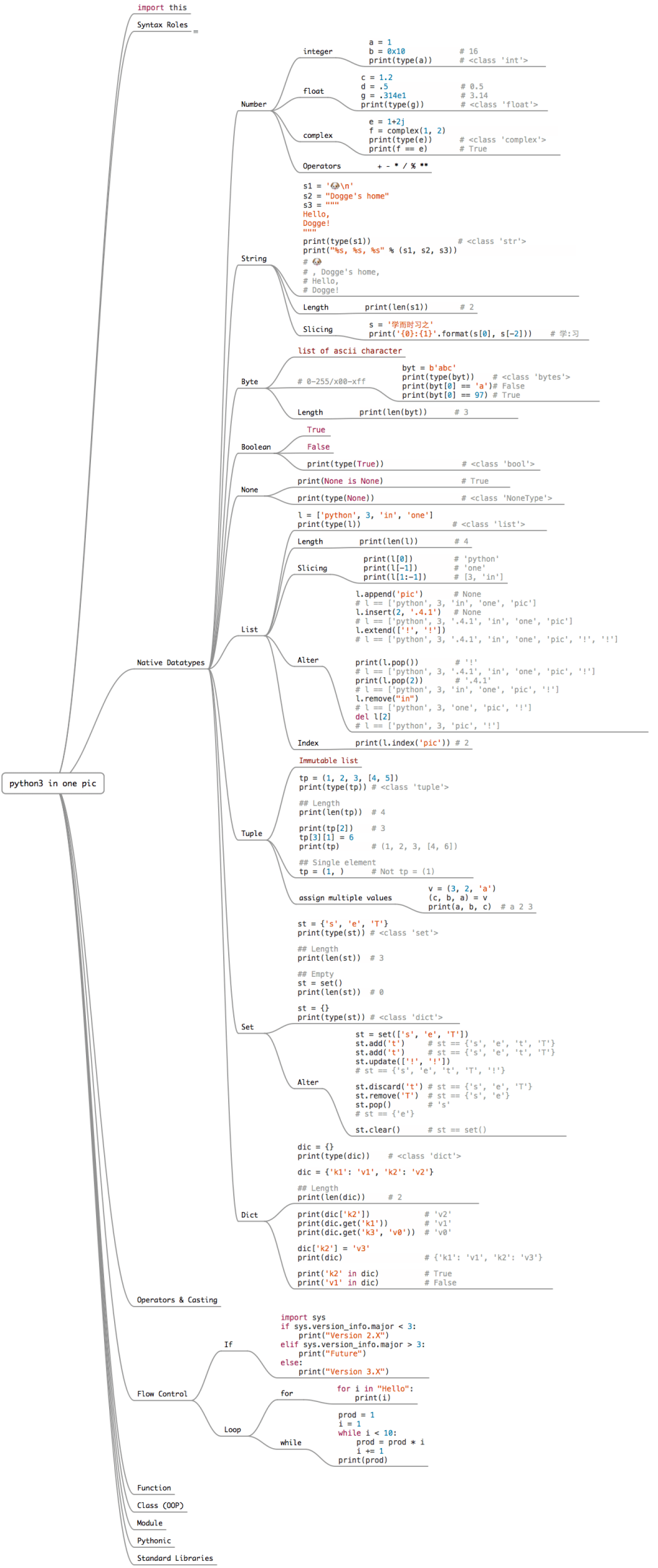
python3 in one pic
import this
Syntax Roles
Native Datatypes
Number
integer
a = 1
b = 0x10
print(type(a))
# 16
# <class 'int'>
float
c = 1.2
d = .5
g = .314e1
print(type(g))
# 0.5
# 3.14
# <class 'float'>
complex
e = 1+2j
f = complex(1, 2)
print(type(e))
print(f == e)
# <class 'complex'>
# True
Operators
+ - * / % **
String
s1 = '🐶\n'
s2 = "Dogge's home"
s3 = """
Hello,
Dogge!
"""
print(type(s1))
print("%s, %s, %s" % (s1, s2, s3))
# 🐶
# , Dogge's home,
# Hello,
# Dogge!
# <class 'str'>
Length
print(len(s1))
# 2
Slicing
s = '学而时习之'
print('{0}:{1}'.format(s[0], s[-2]))
# 学:习
Byte
list of ascii character
# 0-255/x00-xff
byt = b'abc'
print(type(byt))
print(byt[0] == 'a')
print(byt[0] == 97)
# <class 'bytes'>
# False
# True
Length
print(len(byt))
# 3
Boolean
True
False
print(type(True))
# <class 'bool'>
None
print(None is None)
print(type(None))
# True
# <class 'NoneType'>
List
l = ['python', 3, 'in', 'one']
print(type(l))
# <class 'list'>
Length
print(len(l))
# 4
Slicing
print(l[0])
print(l[-1])
print(l[1:-1])
# 'python'
# 'one'
# [3, 'in']
Alter
l.append('pic')
# l == ['python', 3, 'in', 'one', 'pic']
l.insert(2, '.4.1')
# l == ['python', 3, '.4.1', 'in', 'one', 'pic']
l.extend(['!', '!'])
# l == ['python', 3, '.4.1', 'in', 'one', 'pic', '!', '!']
print(l.pop())
# l == ['python', 3, '.4.1', 'in', 'one', 'pic', '!']
print(l.pop(2))
# l == ['python', 3, 'in', 'one', 'pic', '!']
l.remove("in")
# l == ['python', 3, 'one', 'pic', '!']
del l[2]
l == ['python', 3, 'pic', '!']
'!'
'.4.1'
Index
print(l.index('pic')) # 2
Tuple
Immutable list
tp = (1, 2, 3, [4, 5])
print(type(tp)) # <class 'tuple'>
Length
print(len(tp)) # 4
print(tp[2])   # 3
tp[3][1] = 6
print(tp)   # (1, 2, 3, [4, 6])
Single element
tp = (1,)   # Not tp = (1)
assign multiple values
v = (3, 2, 'a')
(c, b, a) = v
print(a, b, c) # a 2 3
Set
st = {'s', 'e', 'T'}
print(type(st)) # <class 'set'>
Length
print(len(st)) # 3
Empty
st = set()
print(len(st)) # 0
st = {}
print(type(st)) # <class 'dict'>
Alter
st = set(['s', 'e', 'T'])
st.add('t')   # st == {'s', 'e', 't', 'T'}
st.add('t')   # st == {'s', 'e', 't', 'T'}
st.update(['!', '!'])
# st == {'s', 'e', 't', 'T', '!'}
st.discard('t') # st == {'s', 'e', 'T'}
st.remove('T') # st == {'s', 'e'}
st.pop()   # 's'
# st == {'e'}
st.clear()   # st == set()
Dict
dic = {}
print(type(dic))   # <class 'dict'>
dic = {'k1': 'v1', 'k2': 'v2'}
Length
print(len(dic))   # 2
print(dic['k2'])   # 'v2'
print(dic.get('k1'))   # 'v1'
print(dic.get('k3', 'v0')) # 'v0'
dic['k2'] = 'v3'
print(dic)   # {'k1': 'v1', 'k2': 'v3'}
print('k2' in dic)   # True
print('v1' in dic)   # False
Operators & Casting
Flow Control
If
import sys
if sys.version_info.major < 3:
    print("Version 2.X")
elif sys.version_info.major > 3:
    print("Future")
else:
    print("Version 3.X")
Loop
for
for i in "Hello":
    print(i)
while
prod = 1
i = 1
while i < 10:
    prod = prod * i
    i += 1
print(prod)
Function
Class (OOP)
Module
Pythonic
Standard Libraries