根据2017年1月8日的一项计划，构造一个基于python3的特殊符号查询表

源程序：

help=''' #定义帮助信息

findspec.py --name|-n theta

findspec.py --similar|-s thee

findspec.py --no-original|-j theta , means do not print original name of the symbol

findspec.py -n nabula -j -n theta,oumu

findspec.py --all|-a

findspec.py --help|-h

'''

specs={

"theta":"θ",

"nabula":"▽",

"wmg":"ω",

"lambda":"λ",

"miu":"μ",

"pie":"π",

"oumu":"Ω",

"delta":"Δ",

}

#定义参数处理类型

def arg\_proc\_list(container,key,value,splitor=","):

vals = value.split(splitor)

if not container[key]:

container[key] = []

container[key].extend(vals)

def arg\_proc\_single(container,key,value):

container[key] = value

import sys

import getopt

args = sys.argv[1:] #获取命令行参数

options = [

(("--name","-n"),"name",arg\_proc\_list),

(("--similar","-s"),"do\_similar",arg\_proc\_single),

(("--no-original","-j"),"no\_original",arg\_proc\_single),

(("--help","-h"),"do\_help",arg\_proc\_single),

(("--all","-a"),"print\_all",arg\_proc\_single),

]

values = { #获取解析参数

"name":[],

"do\_similar":None,

"no\_original":None,

"do\_help":None,

"print\_all":None,

}

short\_options = "n:sjha" #定义长短选项

long\_options = ["name=","similar","no-original","help","all"]

opt, val = getopt.gnu\_getopt(args,short\_options,long\_options)

values["name"].extend(val)

for i in opt:

for j in options:

if i[0] in j[0]:

#values[j[1]] = i[1]

j[2](values,j[1],i[1])

break

#print(values)

if values["do\_help"]!=None: #开始处理过程，一般先处理help选项

print(help)

exit()

elif values["print\_all"]!=None:

for k,v in specs.items():

print(k,v)

exit()

NOTFOUND="Not Found"

for i in values["name"]:

if values["no\_original"] == None:

print(i,specs.get(i,NOTFOUND))

else:

print(specs.get(i,"?"),end=" ")