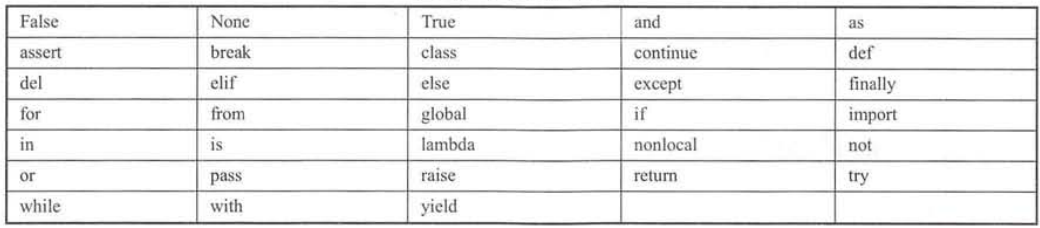
https://www.cnblogs.com/xueweihan

HelloDjango:

git clone https://github.com/HelloGitHub-Team/HelloDjango-blog-tutorial.git

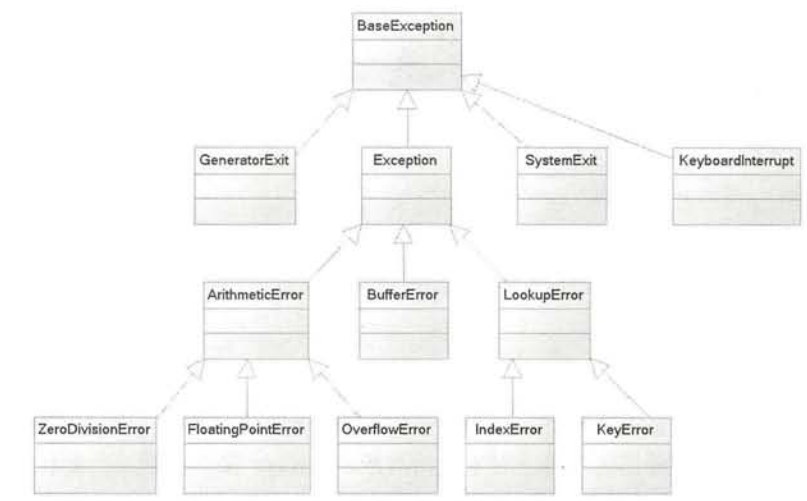
# python关键字

https://www.cnblogs.com/xueweihan/p/4518022.html



# python3内置函数

https://www.runoob.com/python3/python3-built-in-functions.html



python常见异常类的继承关系

# 编解码

https://www.cnblogs.com/zz22--/p/8799071.html

https://www.cnblogs.com/yes5144/p/11523516.html

https://blog.csdn.net/zhuan\_long/article/details/112303276

转码的方法：

str.encode(‘utf-8’).decode(‘unicode\_escape’)

with open(filename, encoding='gbk') as f:

return response.read().decode('utf8')

# split的用法

str = "Line1-abcdef \nLine2-abc \nLine4-abcd";

print(str.split()) #以空格为分隔符进行拆分，包含\n

print(str.split(' ', 1)) #拆成1+1段

print(str.split(' '))

root@2004a:/home/test#

root@2004a:/home/test# python3 test\_split.py

['Line1-abcdef', 'Line2-abc', 'Line4-abcd']

['Line1-abcdef', '\nLine2-abc \nLine4-abcd']

['Line1-abcdef', '\nLine2-abc', '\nLine4-abcd']

root@2004a:/home/test#

#先以空格为分隔符拆成列表，再合并成字符串

text = "".join(html.split())

# 正则表达式 .\*?

表达式 .\* 就是单个字符匹配任意次，即贪婪匹配。

表达式 .\*? 是满足条件的情况只匹配一次，即最小匹配。

root@2004a:/home/test# cat test\_re.py

import re

str='<img src="test.jpg" width="60px" height="80px"/>'

#print(str)

pattern = re.compile('.\*')

table = re.findall(pattern, str)

print(len(table))

print(table[0])

pattern = re.compile('.\*?')

table = re.findall(pattern, str)

print(len(table))

print(table[1])

root@2004a:/home/test# python3 test\_re.py

2

<img src="test.jpg" width="60px" height="80px"/>

97

<

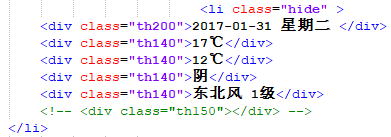
# 正则匹配进阶

虽然XPath更好用，我们仍然使用正则表达式





…



root@2004a:/home/test# cat test.py

import re

from datetime import datetime

from datetime import timedelta

with open('./201701.html', 'r', encoding='utf-8') as f:

html = f.read()

#print(html)

#去掉所有空格

text = "".join(html.split())

#print(text)

#pattern = re.compile('<divclass="tian\_three">(.\*?)</div>')

#table = re.findall(pattern, text)

#print(len(table))

#print(len(table[0]))

pattern = re.compile('<divclass="tian\_three">(.\*?)</div><script>')

table = re.findall(pattern, text)

#print(len(table[0]))

#print(table[0])

pattern1 = re.compile('<ulclass="thrui">(.\*?)</ul>')

table1 = re.findall(pattern1, table[0])

#print(len(table1))

#pattern2=re.compile('<li>(.\*?)</li>')

#pattern2=re.compile('(<li>(.\*?)</li>)|(<liclass="hide">(.\*?)</li>)')

pattern2=re.compile('>(.\*?)</li>')

lis=re.findall(pattern2, table1[0])

#print(len(lis))

#print(lis[0], "\n\n", lis[17])

#divs=re.findall('>(.\*?)</div>', lis[0])

#print(divs)

#divs=re.findall('>(.\*?)</div>', lis[17])

#print(divs)

for li in lis:

divs=re.findall('>(.\*?)</div>', li)

#print(divs[0], divs[1], divs[2])

d\_str=re.findall('(.\*)星期', divs[0])[0]

low\_str=re.findall('(.\*)℃', divs[1])[0]

high\_str=re.findall('(.\*)℃', divs[2])[0]

print(d\_str, low\_str, high\_str)

cur\_day=datetime.strptime(d\_str, '%Y-%m-%d')

high=int(high\_str)

low=int(low\_str)

print(cur\_day,high,low)

root@2004a:/home/test#