

WenQuanYi Micro Hei [Scale=0.9]WenQuanYi Micro Hei Mono song-
WenQuanYi Micro Hei sfWenQuanYi Micro Hei "zh" = 0pt plus 1pt

çŁňèŽńă■ęăžă Documentation

ăŘŚăŸČ 1.0

Huang Xinyuan

2019 ăžť 05 æIJŁ 07 æŮě

Contents

7.4	re.sub	27
7.5	re.compile	29
8	Beautiful Soup	31
8.1	èġčæđŘăŜ	31
8.2	âşžæIJñă;ŕçŤÍ	31
8.3	æăĜç■ġéĂĹ'æŊŦ'ăŹÍ	33
8.4	æăĜăĜĖġéĂĹ'æŊŦ'ăŹÍ	41
8.5	CSSéĂĹ'æŊŦ'ăŹÍ	44
8.6	æĂžçzŜ	47
9	PyQuery	48
9.1	ăĹĲăĝŊăŊŨ	48
9.2	âşžæIJñCSSéĂĹ'æŊŦ'ăŹÍ	49
9.3	æşěæĹ;ăĖČçŦ'ă	50
9.4	éA■ăŖĖ	53
9.5	èŖăăŖŨăăŕăæAŕ	55
9.6	DOMæŜ■ă;IJ	57
9.7	ăĲĲçşzēĂĹ'æŊŦ'ăŹÍ	60
9.8	ăŏŸæŨžæŨĜæąč	61
10	çď'ză;ŊĲjŽăĹŦçŤĲĹñèŽñèŖăăŖŨăđŦ'æŕŤăăŕăæAŕ	62
10.1	ăžžăĹăăĹĖæđŘ	62
10.2	çĹŊăžŖçĲjŨăĖŽ	62
10.3	æIJĂçzĹçĹŊăžŖ	64

CHAPTER 1

åŖŠéĂÆèŕŭæśĆ

1.1 äžěGETå¡ćåijŖåŖŠéĂÆæŤŕæ■ŏ

```
import urllib.request

response = urllib.request.urlopen('http://www.baidu.com')
print(response.read().decode("utf-8"))
```

1.2 äžěPOSTå¡ćåijŖåŖŠéĂÆæŤŕæ■ŏ

```
import urllib.parse
import urllib.request

# POSTçşżăđŇéIJĂèēĂăijăèçşdataăŖĆæŤŕ
data = bytes(urllib.parse.urlencode({"world":"hello"}), encoding =
    ↳'utf-8')
response = urllib.request.urlopen('http://httpbin.org/post', data =
    ↳data)
print(response.read())
```

1.3 timeoutåŖĆæŤŕ

ăēĆăđIĲèőēŕŮőēŭĚăŮŕŕijŇăŁŽăĜăijĆăyyăĂĆ

```
import socket
import urllib.request
import urllib.error

try:
    response = urllib.request.urlopen("http://httpbin.org/get",
    ↪ timeout = 0.1)
except urllib.error.URLError as e:
    if isinstance(e.reason, socket.timeout):
        print("TIME OUT")
```

CHAPTER 2

å\$■åžŤèrûæśĆ

2.1 å\$■åžŤçśzådŇ

```
import urllib.request

response = urllib.request.urlopen("http://www.python.org")
print(type(response))
```

2.2 çŁÚæĀAçăAăĀAå\$■åžŤåd't

```
import urllib.request

response = urllib.request.urlopen("http://www.python.org")
print(response.status)
print(response.getheaders())
print(response.getheader('Server'))
```

2.3 èŎûåRŮå\$■åžŤä;ŞåĘăŏż

```
import urllib.request

response = urllib.request.urlopen("http://www.python.org")
print(response.read().decode('utf-8'))
```

2.4 äĭ£çŤÍRequestăřžèśą

```
import urllib.request
```

```
request = urllib.request.Request("http://python.org")
response = urllib.request.urlopen(request)
print(response.read().decode('utf-8'))
```

èŁŽăăŭăŚăĭJăăđčŎřčžŚăđIJăŠŇăŸŁéíćăĖŽčŽĎěŎŭăŔŮăŚăăžŤă;ŚăĖĚăăŏžăăđčŎřčžŚăđIJăŸřăŸĂă

```
from urllib import request, parse
```

```
url = 'http://httpbin.org/post'
headers = {
    'User-Agent': 'Mozilla/4.0', 'Host': 'httpbin.org'
}
dict = {
    'name': 'Germey'
}
data = bytes(parse.urlencode(dict), encoding = 'utf-8')
req = request.Request(url = url, data = data, headers = headers,
    ↪method = 'POST')
response = request.urlopen(req)
print(response.read().decode('utf-8'))
```


CHAPTER 3

ä;£çŦÍHandler

èõÿâĎŽénŸçžgæŞ■ä;IJiijŇæfŦæĈèĎFTPæLŮèĂĚæŞ■ä;IJcacheéIJĂèçAä;£çŦÍHandleræİèè£ZèaŇăĂ

3.1 äžčçŘĚ

```
import urllib.request

proxy_handler = urllib.request.ProxyHandler({
    'http': 'http://127.0.0.1:9743',
    'https': 'https://127.0.0.1:9743'
})
opener = urllib.request.build_opener(proxy_handler)
response = opener.open('http://www.baidu.com')
print(response.read())
```

3.2 Cookie

CookieæŸřăIJlăóçæLûçńřăİçŦŽiijŇçŦlæİèèõřă;ŦçŦlæLûèžñăz;çŽĐæŮĜæIJñæŮĜăzűăĂCăIJlçLñèŽñ

```
import http.cookiejar, urllib.request

cookie = http.cookiejar.CookieJar()
→ #ä;£çŦÍcookieééŮăĚĹéIJĂèçAăčřæŸŮăÿžăÿĂăÿİCookieJarçŽĐăřžèşă
handler = urllib.request.HTTPCookieProcessor(cookie)
opener = urllib.request.build_opener(handler)
response = opener.open('http://www.baidu.com')
```

(ăÿŇéatçžğçž■)

CHAPTER 4

4.1. urllib 模块

```
from urllib import request, error
try:
    response = request.urlopen('http://cuiqingcai.com/index.htm')
except error.URLError as e:
    print(e.reason) #ä;£çŤĲe.reasonæL'Şå■řâijĆăÿÿăŎŞăŽă
```

urllib.erroräYÄäËæIJL'äyd'çg■errorijŇăĹEăĹăŇæYŕURLErrorăŞŇHTTPErrorijŇăËäÿ■UR

```
from urllib import request, error
try:
    response = request.urlopen('http://cuiqingcai.com/index.htm')
except error.HTTPError as e:
    print(e.reason, e.code, e.headers, sep = '\n')
except error.URLError as e:
    print(e.reason)
else:
    print('Request Successfully')
```

è£YăŔfäzèçŤĲè£Zçg■æŮzæşŤæĲèĲŇèŕAâijĆăÿÿijŽ

```
import socket
import urllib.request
import urllib.error

try:
    response = urllib.request.urlopen("http://httpbin.org/get",
    ↪timeout = 0.1)
except urllib.error.URLError as e:
    if isinstance(e.reason, socket.timeout):
        print("TIME OUT")
```

èűĖæŮűăijĆăŸŸçŽĎěſle.reasonæŸŕ <class 'socket.
timeout>çśşăđNiiŸŇěĂŽěſĠisinstanceèſZëąŇçśşăđŇăŇzéĖ■æſěéſŇěŕAăijĆăŸŸăĂĆ

CHAPTER 5

URL parsing

5.1 urllib.parse

`urllib.parse.urlparse(urlstring, scheme='', allow_fragments=True)`

urlparse returns a tuple of (scheme, netloc, path, params, query, fragment)

```
from urllib.parse import urlparse

result = urlparse('http://www.baidu.com/index.html;user?id=5#comment
→')
print(type(result), result)
```

`<class 'urllib.parse.ParseResult'> ParseResult(scheme='http', netloc='www.baidu.com', path='/index.html', params='', query='user?id=5', fragment='comment')`

urlparse also accepts a `url` argument

```
from urllib.parse import urlparse

result = urlparse('www.baidu.com/index.html;user?id=5#comment',
→scheme='https')
print(result)
```

urlparse also accepts a `url` argument

```
from urllib.parse import urlparse
```

(`urlparse.urlparse`)

(çž■äŸĹéął)

```
result = urlparse('http://www.baidu.com/index.html;user?id=5#comment
↳', allow_fragments=False)
print(result)
```

èĹŽăăüiijŃfragmentăřsăijŽăŃijăŌěăĹřăĹ■éíççŽĎă;■ç;ŏăŌžiiijŃè;ŚăĠžçzŚăđIJăŸžiiijŽ

```
ParseResult(scheme='http', netloc='www.baidu.com',
path='/index.html', params='user', query='id=5#comment', frag-
ment='')
```

ăĕĆăđIJqueryăžšăŸřçĹ'žçŽĎěřiiijŃfragmentçŽĎăĀijăĕĆăđIJăŸřfalseççŽĎěřiiijŃăĀijăijŽăŃijăŌěăĹ

5.2 urlunparse

çĹĹéĹŽăŸĹăŰžăşŤăĹĹçĹĹĠăŏťăŃijăĹŔăŸĀăŸĥurl

```
from urllib.parse import urlunparse

data = ['http', 'www.baidu.com', 'index.html', 'user', 'a=6',
↳ 'comment']
print(urlunparse(data))
```

5.3 urljoin

èĹŽăŸĹăžšăŸřçĹĹăĹăŃijURLiijŃă;ĒăŸřăŽĹ'ăŸžăijžăđ'ğăĂĆ

```
from urllib.parse import urljoin

print(urljoin('http://www.baidu.com', 'FAQ.html'))
print(urljoin('http://www.baidu.com', 'https://cuiqingcai.com/FAQ.
↳ html'))
print(urljoin('http://www.baidu.com/about.html', 'https://
↳ cuiqingcai.com/FAQ.html'))
print(urljoin('http://www.baidu.com/about.html', 'https://
↳ cuiqingcai.com/FAQ.html?question=2'))
print(urljoin('http://www.baidu.com?wd=abc', 'https://cuiqingcai.
↳ com/index.php'))
print(urljoin('http://www.baidu.com', '?category=2#comment'))
print(urljoin('www.baidu.com', '?category=2#comment'))
print(urljoin('www.baidu.com#comment', '?category=2'))
```

è;ŚăĠžçzŚăđIJ

```
http://www.baidu.com/FAQ.html #èĹŽăŸřăřăĒăŸđ'ăŸłçĹĹĠăŏťăŃijăĹĹăŸŸăĒťă
https://cuiqingcai.com/FAQ.html
↳ #ăŔŌěíéçŽĎă■ŰăŏťăŔ■ăijžèĹăĹíéçĒçŽŰăŐĹ'ăĹ'■éíççŽĎă■Űăŏť
```

(ăŸŃéąţçžçz■)

(çžăŸŁéął)

```
https://cuiqingcai.com/FAQ.html
https://cuiqingcai.com/FAQ.html?question=2
https://cuiqingcai.com/index.php
http://www.baidu.com?category=2#comment
www.baidu.com?category=2#comment
www.baidu.com?category=2
```

ăĽăĹăŕăžěçĬŇăĹŕĭjŇăŦ'ăŸłURLăŕăžěăĹĒăĹŔ6ăŸłăŰăŏŦĭjŇăęĈăđĬăŔŖŖŏĹăĹčŽĐăŰăŏŦăŸăĹăŏ

5.4 urlencode

ăŔŕăžěăĹăŸăŸŸłăŰăĚŸăŕžěsăęĭŇăăĹŔgetĕŕŭăśĈăŔĈăŦŕăĂĈăĹĽ'çĚğ&èĚZăŇăĹĒăĹšăĂĈăŽăă

```
from urllib.parse import urlencode

params = {
    'name': 'germey',
    'age': 22
}
base_url = 'http://www.baidu.com?'
url = base_url + urlencode(params)
print(url)
```

çžŖăđĬĭjŽhttp://www.baidu.com?name=germey&age=22

CHAPTER 6

Requests

RequestsæYřçTÍPythonèr■èlÄçijŮâEZçŽDrijNâšžāžŮurllibrijNä;EæYřāōČæřTurllibæŽřāŁäæŮžä;£rijN

6.1 GETèrúæśĆ

6.1.1 āšžæIJnèrúæśĆ

```
import requests

response = requests.get('http://httpbin.org/get')
print(response.text)
```

āřsāRřāžæL'Sā■řāšžæIJnèrúæśĆČāŁæAř

```
{
  "args": {},
  "headers": {
    "Accept": "*/*",
    "Accept-Encoding": "gzip, deflate",
    "Connection": "close",
    "Host": "httpbin.org",
    "User-Agent": "python-requests/2.20.0"
  },
  "origin": "222.197.179.9",
  "url": "http://httpbin.org/get"
}
```


6.1.2 äŸęăŘĆæŤŕGET

```
import requests
response = requests.get("http://httpbin.org/get?name=germey&age=22")
print(response.text)
```

ăŠŇăŸŇéĹćŽĎăĚŽăŤæŸŕăőŇăĚläŸĂæăŭçŽĎŕĭjŇăŸŇéĹćăĚŽăŤçŽĎăĕ;ăđ'ĎăŸŕăŸ■çŤĹăĚŽăŕĎĎçğ■

```
import requests

data = {
    'name': 'germey',
    'age': 22
}
response = requests.get("http://httpbin.org/get", params=data)
print(response.text)
```

6.1.3 èğçæđŘjson

èĚŽăŸĹăŤ■ăĭJăŕŕăžěçŽŕ'æŐăŕĚèŐăŕŮçŽĎçžŚæđIJèĭŇăŇŮăŸžjsonæăĭjăĭjŕăĂĆ

```
import requests
import json

response = requests.get("http://httpbin.org/get")
print(type(response.text))
print(response.json()) # response.
→jsonăőđéŽĚăŸĹăŕśăŸŕăĹ'ğèăŇăžĚăŸĂăŸĹjson.
→loadsăŮžăŤŕĭjŇăŇŮăŸŇéĹćér■ăŕĚăŤĹăđIJăŸŕăŸĂæăŭçŽĎ
print(json.loads(response.text))
print(type(response.json()))
```

6.1.4 èŐăŕŮăžŇèĚŽăĹăŤŕæ■ő

èŐăŕŮăžŇèĚŽăĹăŤŕæ■őăĭJăŸŷŇèĭ;ăŽĭçĹĠăŠŇăŸŇèĭ;èğĚćŚçŽĎăŮăăĂŽăĭJăŸžăŸŸçŤĹăŮžăŤăŤă

```
import requests

response = requests.get("https://github.com/favicon.ico")
print(type(response.text), type(response.content))
print(response.text)
print(response.content) #ăĭ;ĚçŤĹresponse.
→contentăĹēèŐăŕŮăžĭçĹ'ĠçŽĎăžŇèĚŽăĹăŤăĚăőž
```

ăĚĹă■ŸçŽĎŕĭjŇăŕŕăžăă;ĚçŤĹèĚŽçğ■ăŮžăĭjŕăĹăăĹă■ŸŕĭjŽ

```
import requests

response = requests.get("https://github.com/favicon.ico")
with open('favicon.ico', 'wb') as f:
    f.write(response.content)
f.close()
```

6.1.5 æůžăĹăheaders

```
import requests

headers = {
    'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_4)
    ↳AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116
    ↳Safari/537.36'
}
response = requests.get("https://www.zhihu.com/explore",
    ↳headers=headers)
print(response.text)
```

6.2 POSTèrűæśĆ

ä;ŁçŤĪPOSTèrűæśĆăŸĂăőŽèĚĂăŘŚăŸĂăŸĹForm DataiijŇăĹŚăžňăŘřăžèçŤĪ-
dataă■ŮăĚŸăĹčăiĵăăĚččăĹă■ŤăĂĆ

```
import requests

data = {'name': 'germey', 'age': '22'}
headers = {
    'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_4)
    ↳AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116
    ↳Safari/537.36'
}
response = requests.post("http://httpbin.org/post", data=data,
    ↳headers=headers)
print(response.json())
```

6.3 áŞăăŽŤ

6.3.1 responseăśďæĂğ

ăŸŇéĹăĹŮăĠžăŸŸçŤĪçŽĐresponseăśďæĂğăĂĆ

```
import requests

response = requests.get('http://www.jianshu.com')
print(type(response.status_code), response.status_code)
print(type(response.headers), response.headers)
print(type(response.cookies), response.cookies)
print(type(response.url), response.url)
print(type(response.history), response.history)
```

6.3.2 çŁŮăĀÇăĀăŁđ'æŮ

äŸŇéİççŻăĜçŁŮăĀÇăĀÇăŽĐăŔĐăŸİçİŮăŔŮăŕăžăŤçŽĐăŔăăŮİİjŇăĒĹŖăžăŇăĪĴăăđéŽĚăİçŤİēĚĜçİŇ

```
100: ('continue',),
101: ('switching_protocols',),
102: ('processing',),
103: ('checkpoint',),
122: ('uri_too_long', 'request_uri_too_long'),
200: ('ok', 'okay', 'all_ok', 'all_okay', 'all_good', '\\o/', '✓'),
201: ('created',),
202: ('accepted',),
203: ('non_authoritative_info', 'non_authoritative_information'),
204: ('no_content',),
205: ('reset_content', 'reset'),
206: ('partial_content', 'partial'),
207: ('multi_status', 'multiple_status', 'multi_stati', 'multiple_
    ↳stati'),
208: ('already_reported',),
226: ('im_used',),

# Redirection.
300: ('multiple_choices',),
301: ('moved_permanently', 'moved', '\\o-'),
302: ('found',),
303: ('see_other', 'other'),
304: ('not_modified',),
305: ('use_proxy',),
306: ('switch_proxy',),
307: ('temporary_redirect', 'temporary_moved', 'temporary'),
308: ('permanent_redirect',
    'resume_incomplete', 'resume'), # These 2 to be removed in 3.
    ↳0

# Client Error.
400: ('bad_request', 'bad'),
401: ('unauthorized',),
402: ('payment_required', 'payment'),
403: ('forbidden',),
```

(äŸŇéăŧççğçă)

(čžăŸĹéął)

```

404: ('not_found', '-o-'),
405: ('method_not_allowed', 'not_allowed'),
406: ('not_acceptable',),
407: ('proxy_authentication_required', 'proxy_auth', 'proxy_
    ↪authentication'),
408: ('request_timeout', 'timeout'),
409: ('conflict',),
410: ('gone',),
411: ('length_required',),
412: ('precondition_failed', 'precondition'),
413: ('request_entity_too_large',),
414: ('request_uri_too_large',),
415: ('unsupported_media_type', 'unsupported_media', 'media_type'),
416: ('requested_range_not_satisfiable', 'requested_range', 'range_
    ↪not_satisfiable'),
417: ('expectation_failed',),
418: ('im_a_teapot', 'teapot', 'i_am_a_teapot'),
421: ('misdirected_request',),
422: ('unprocessable_entity', 'unprocessable'),
423: ('locked',),
424: ('failed_dependency', 'dependency'),
425: ('unordered_collection', 'unordered'),
426: ('upgrade_required', 'upgrade'),
428: ('precondition_required', 'precondition'),
429: ('too_many_requests', 'too_many'),
431: ('header_fields_too_large', 'fields_too_large'),
444: ('no_response', 'none'),
449: ('retry_with', 'retry'),
450: ('blocked_by_windows_parental_controls', 'parental_controls'),
451: ('unavailable_for_legal_reasons', 'legal_reasons'),
499: ('client_closed_request',),

# Server Error.
500: ('internal_server_error', 'server_error', '/o\\', 'âĹĴŮ'),
501: ('not_implemented',),
502: ('bad_gateway',),
503: ('service_unavailable', 'unavailable'),
504: ('gateway_timeout',),
505: ('http_version_not_supported', 'http_version'),
506: ('variant_also_negotiates',),
507: ('insufficient_storage',),
509: ('bandwidth_limit_exceeded', 'bandwidth'),
510: ('not_extended',),
511: ('network_authentication_required', 'network_auth', 'network_
    ↪authentication'),

```

ăŸŇéĹăŸď'čĝăĕŽăŝŤăđđčŖăĹŝěČĵăŸăŸĂăăŭčŽďĹĵŽ

```
import requests
```

(ăŸŇéąłčžĝčžă)


```
import requests
from requests.exceptions import ReadTimeout #âĲTăĚěTimeoutçśż
try:
    response = requests.get("http://httpbin.org/get", timeout = 0.5)
    print(response.status_code)
except ReadTimeout:
    print('Timeout')
```

```
import requests
from requests.auth import HTTPBasicAuth

r = requests.get('http://120.27.34.24:9001', auth=HTTPBasicAuth(
    ↪ 'user', '123'))
print(r.status_code)
```

```
import requests

r = requests.get('http://120.27.34.24:9001', auth=('user', '123'))
print(r.status_code)
```

```
import requests
from requests.exceptions import ReadTimeout, ConnectionError,
↳RequestException
try:
    response = requests.get("http://httpbin.org/get", timeout = 0.5)
    print(response.status_code)
except ReadTimeout:
    print('Timeout')
except ConnectionError:
    print('Connection error')
except RequestException:
    print('Error')
```


CHAPTER 7

æ■čāŁŻèąłè¿¿ąįŖ

æ■čāŁŻèąłè¿¿ąįŖæŸřąřą■ŮčņęäŸşçŽďäŸĂçğ■éĂzè¿ŚăĖņăįŖĭįŃăŕşæŸŕçŦlăžŃăĖĹăőŽăžĹ'ăë¿çŽďäŸ
ăĬĴçžŧætŃērŦæ■čāŁŻèąłè¿¿ąįŖ

7.1 re.match

re.matchăŕĭĕŕŦăžŎă■ŮčņęäŸşçŽďĕŦăğŃă¿ç¿őăŃzéĚ■ăŸĂăŸłăĹăįŖĭĭŃăĕĆăđĬăŸ■æŸŕĕŦăğŃă¿ç¿
re.match(pattern, string, flags = 0)

7.1.1 æĬĂăŸŸĕğĐçŽďăŃzéĚ■

```
import re

content = 'Hello 123 4567 World_This is a Regex Demo'
print(len(content))
result = re.match('^Hello\s\d\d\d\s\d{4}\s\w{10}.*Demo$', content)
print(result)
print(result.group()) #èĤăŽďăŃzéĚ■çžşæđĬĴ
print(result.span()) #è¿ŚăĜžăŃzéĚ■çžşæđĬĴçŽďĕŃčăžŦ'
```

7.1.2 æşŽăŃzéĚ■

ăžŃăĹ■ăĖŽçŽďăŃzéĚ■èŧĞăžŎăđ■ăĬĤĭĭŃăĕĆăđĬăĈşĕĕĂăőđçŎŕăŕŃăăŭçŽďăŦĹăđĬăŕŕăžĕăĬĴ
*ăĬăŃzéĚ■ăĹĂăĬĴçŽďă■ŮčņęäŸşăĂĆ

```
import re

content = 'Hello 123 4567 World_This is a Regex Demo'
result = re.match('^Hello.*Demo$', content) #ă;ĚčŤÍ.
→*ăŕśăŔŕăžěăĹĹăŸ■éŮť'čŽĎăĹ'ĂăĪĴĴ'čžŚăđĪĴéČ;ăŇžéĚ■èĚĞăŌžăžĚ
print(result)
print(result.group())
print(result.span())
```

7.1.3 ăŇžéĚ■čŽŏăăĞ

ăŕŤăĚĆĕŕť'ăĪĴăŸŇéĪčŽĎă■ŮčňăŸšéĞŇéĪăĈšĕĕAĕŌăăŔŮ1234567iijŇăĹŚăžňăŕŕăžěăĹĹ1234567č
1234567čŽĎăŮĕčňŕčĆžăŸŕăŸĂăŸĴčĴ'žčŽ;ă■ŮčňĕiijŇăĹĂăžĕčŤĴsĭijŇăžŇăŔŌĕĕAăŇžéĚ■čŽĎăŸŕăŸĂ

```
import re

content = 'Hello 1234567 World_This is a Regex Demo'
result = re.match('^Hello\s(\d+)\sWorld.*Demo$', content)
print(result)
print(result.group(1))
print(result.span())
```

7.1.4 èŤĴăĴ'ĴăŇžéĚ■

```
import re

content = 'Hello 1234567 World_This is a Regex Demo'
result = re.match('^He.*(\d+).*Demo$', content)
print(result)
print(result.group(1))
```

èĴŚăĜžčžŚăđĪĴiijŽ

```
<_sre.SRE_Match object; span=(0, 40), match='Hello 1234567 World_
→This is a Regex Demo'>
7
```

ăžŌăŽĴăŸ■ăŕŕăžĕčĪŇăăĜžĭijŇĕť'ĴăĴ'ĴăĴăĴiijŔčŽĎĕŕĴiijŇăĹ■éĪč.
*ăiijŽăŸĂčŽť'ăŇžéĚ■ăŤŕă■ŮiijŇăŸĂčŽť'ăĴŕ7ăĴ■čĴiijŇăŽăăŸžăŔŌéĪăĪĴ'(\d+)iijŇăĴĚăžèĴŚăĜžăŸĂăŸĴă

7.1.5 éĪđĕť'ĴăĴ'ĴăŇžéĚ■

éĪđĕť'ĴăĴ'ĴăŇžéĚ■iijŇăŕŕăžěăĪĴĴ. *ăŔŌéĪăĴăăŸĂăŸĴ?iijŇĕĴŽăăăăŇžéĚ■čŽĎčžŚăđĪĴăŕśăŸŕéĪđĕť'ĴăĴ'Ĵă

```
import re

content = 'Hello 1234567 World_This is a Regex Demo'
result = re.match('^He.*?(\\d+).*Demo$', content)
print(result)
print(result.group(1))
```

èĭŚăĜžăÿžĭijŽ

```
<_sre.SRE_Match object; span=(0, 40), match='Hello 1234567 World_
↳This is a Regex Demo'>
1234567
```

ĭijŝăĖŖŌéĬcăIJĹăÿĂăÿĤd+ĭijŇèĖŕ' æÿŖŌèçAăijĂăĝŇăŇzéĖ■æŤră■ŪăžEĭijŇăĖŕ;ăĖŖèĈ;ăŇzéĖ■ăĖŖŤçŽDăeŤ

7.1.6 âŇzéĖ■ăĖĭăĭijŖ

```
import re

content = '''Hello 1234567 World_This
is a Regex Demo
'''
result = re.match('^He.*?(\\d+).*?Demo$', content, re.S)
↳#ăŇzéĖ■ăĖĭăĭijŖăĖŖŤçŽDăeŤre.S
print(result.group(1))
```

èĖŽéĜŇçŽDcontentæÿŖăÿd'èăŇĭijŇăÿ■éŮŕ' éŽŤçĬĂăÿĂăÿĹæ■cèăŇçĖĭijŇăçCădĬăÿ■ăĹăre .
séĈcăžĹăĹŖă■cèăŇçĖă;■ç;ŏăŖŝăAĬJæ■căžEĭijŇăÿ■ăĭijŽăŇzéĖ■ăĖŖŌéĬçŽDăEĖŖăŏžăĂĈăĹăăžEĖēŽăÿĹăŇzéĖ

7.1.7 èĭŇăžĹ'

ăĖŖŤăžŇăIJĹçŽDăŮŭăĂŽèĖÿæĈŝăĭŮăĹŖăÿĂăžŽçĹ'žăŏĹă■ŮçĖĭijŇăĖŖŤăçĈĖŕŕ' \$5.00ĭijŇăĖŖŤăžŇăçŝă

```
import re

content = 'price is $5.00'
result = re.match('price is \\$5\\.00', content)
print(result)
```

ăĂççŝĭijŽăŖ;éĖŖă;éçŤĹăŝŽăŇzéĖ■ăĂăă;éçŤĹăŇŇăŖăŭăŮăĹŖăŇzéĖ■çŽŏăăĜăĂăăŖ;éĖŖă;éçŤ

7.2 re.search

re.matchæIJĹăÿĂăÿĹăÿ■ăd'ĹæŮžăĭççŽDăĬJŖæŮžæÿŖăžŮăijŽăžŖçŇŇăÿĂăÿĹă■ŮçĖăĭijĂăĝŇăŇzéĖ■ĭijŇ
re.searchăĭijŽăĹ'ŇăĖŖŖæŤŕ'ăÿĹă■ŮçĖăÿŝăžŭèĖŤăŽdçŇŇăÿĂăÿĹăĹŖăĹŝçŽDăŇzéĖ■ăĂĈăĹ'ĂăžĖĭijŇăÿ

```
import re

html = '''<div id="songs-list">
  <h2 class="title">čžŘăĚÿèĂăĕŃ</h2>
  <p class="introduction">
    čžŘăĚÿèĂăĕŃăĹŮèăĹ
  </p>
  <ul id="list" class="list-group">
    <li data-view="2">ăÿĂèŭăÿŁăIJL'ă;ă</li>
    <li data-view="7">
      <a href="/2.mp3" singer="ăžžèt'd'é;Ř">ăšğætŭăÿĂăčřčňŚ</
→a>
    </li>
    <li data-view="4" class="active">
      <a href="/3.mp3" singer="é;Řçğę">ăĹĂăžŇéŽŘéčŎ</a>
    </li>
    <li data-view="6"><a href="/4.mp3" singer="beyond">
→ăĚĹ'è;Ĺ'ăšĂăIJĹ</a></li>
    <li data-view="5"><a href="/5.mp3" singer="éžĹăĚğçŘș">
→èőřăžŇăIJň</a></li>
    <li data-view="5">
      <a href="/6.mp3" singer="éĆšăÿ;ăŘŽ"><i class="fa fa-user
→"></i>ă;ĒăĎăžžéĤăžĚ</a>
    </li>
  </ul>
</div>'''
result = re.search('<li.*?active.*?singer="(.*?)">(.*?)</a>', html,
→re.S)
if result:
    print(result.group(1), result.group(2))
→#group(1) äžčèăĹčňňăÿĂăÿĹăřŘăŇăăŖŭăŇžéĚčŽĎăĒăőžiiĴŇgroup(2) äžčèăĹčňňăžŇăÿĹăřŘă
```

èĹŚăĜžčžŚăđIJăŸřijŽé;Řçğę âĹĂăžŇéŽŘéčŎ

èĚŽéĜŇčŽĎactiveèăĹčđ'žčŽĎăŸřăŎžăŇžéĚăÿęăIJĹactivečŽĎăăĜçăĹăĂĆ

ăÿŇéĹćIJŇăÿĂăÿŇăĕĆăđIJăšăăIJĹactivečŽĎăČĚăĒřijŇHTMLèĚŸăŸřăžŇăĹăčŽĎHTMLăĂĆ

```
import re

html = '''<div id="songs-list">
  <h2 class="title">čžŘăĚÿèĂăĕŃ</h2>
  <p class="introduction">
    čžŘăĚÿèĂăĕŃăĹŮèăĹ
  </p>
  <ul id="list" class="list-group">
    <li data-view="2">ăÿĂèŭăÿŁăIJL'ă;ă</li>
    <li data-view="7">
      <a href="/2.mp3" singer="ăžžèt'd'é;Ř">ăšğætŭăÿĂăčřčňŚ</
→a>
```

(ăÿŇéăĹčžğçžă)

(čžăŷĽéąŧ)

```

        </li>
        <li data-view="4" class="active">
            <a href="/3.mp3" singer="éĭŘčġę">ăĹĂăžŇéŽŘéčŎ</a>
        </li>
        <li data-view="6"><a href="/4.mp3" singer="beyond">
→ăĚĽ'èĹ'ăšĂæIJĽ</a></li>
        <li data-view="5"><a href="/5.mp3" singer="éŽĽăĚġčŘš">
→èöřăžŇæIJň</a></li>
        <li data-view="5">
            <a href="/6.mp3" singer="éĆŞăŷĭăŘŽ">ăĭĚăĎĽăžžéŧĽăžĚ</a>
        </li>
    </ul>
</div>'''
result = re.search('<li.*?singer="(.*?)">(.*?)</a>', html, re.S)
if result:
    print(result.group(1), result.group(2))

```

èĭŞăĠčžčŞăđIJăŸřijŽăžžèťđ'éĭŘ æšġætŭăŷĂăčřčňŚ

7.3 re.findall

re.searchăŸřăĽĭăĽřăŷĂăŷĭčžŞăđIJĭijŇre.findallăŸřèĽŧăŽđăĽĂăIJĽăŇzéĚăčžŞăđIJăĂĆ

```

import re

html = '''<div id="songs-list">
    <h2 class="title">čžŘăĚŷëĂĂă■Ň</h2>
    <p class="introduction">
        čžŘăĚŷëĂĂă■ŇăĽŮèăĭ
    </p>
    <ul id="list" class="list-group">
        <li data-view="2">ăŷĂăèŭřăŷĽăIJĽ'ăĭă</li>
        <li data-view="7">
            <a href="/2.mp3" singer="ăžžèťđ'éĭŘ">æšġætŭăŷĂăčřčňŚ</
→a>
        </li>
        <li data-view="4" class="active">
            <a href="/3.mp3" singer="éĭŘčġę">ăĹĂăžŇéŽŘéčŎ</a>
        </li>
        <li data-view="6"><a href="/4.mp3" singer="beyond">
→ăĚĽ'èĹ'ăšĂæIJĽ</a></li>
        <li data-view="5"><a href="/5.mp3" singer="éŽĽăĚġčŘš">
→èöřăžŇæIJň</a></li>
        <li data-view="5">
            <a href="/6.mp3" singer="éĆŞăŷĭăŘŽ">ăĭĚăĎĽăžžéŧĽăžĚ</a>
        </li>
    </ul>

```

(ăŷŇéąŧčžġčž■)

(çžăÿĹéął)

```
</div>'''
results = re.findall('<li.*?href="(.*?)".*?singer="(.*?)">(.*?)</a>
→', html, re.S)
print(results)
print(type(results))
for result in results:
    print(result)
    print(result[0], result[1], result[2])
```

èĹŚăĜžçžŚăđĹĹijŽ

```
[('/2.mp3', 'äzzèt'd'é;Ř', 'æšğætűäÿĂăčřčňŚ'), ('/3.mp3', 'é;Řçğę',
→'ăĹĂăžŇéŽŘéčŮ'), ('/4.mp3', 'beyond', 'ăĚĹ'èĹ'ăšĂæĹĹ'), ('/5.mp3
→', 'éŽĹæĚğçŘş', 'èõřăžŇæĹĹ'), ('/6.mp3', 'éĆŚăÿ;ăŘŽ',
→'ă;ĒăĎĹăžžéŤĹăžĚ')]
<class 'list'>
('/2.mp3', 'äzzèt'd'é;Ř', 'æšğætűäÿĂăčřčňŚ')
/2.mp3 äzzèt'd'é;Ř æšğætűäÿĂăčřčňŚ
('/3.mp3', 'é;Řçğę', 'ăĹĂăžŇéŽŘéčŮ')
/3.mp3 é;Řçğę äĹĂăžŇéŽŘéčŮ
('/4.mp3', 'beyond', 'ăĚĹ'èĹ'ăšĂæĹĹ')
/4.mp3 beyond âĚĹ'èĹ'ăšĂæĹĹ
('/5.mp3', 'éŽĹæĚğçŘş', 'èõřăžŇæĹĹ')
/5.mp3 éŽĹæĚğçŘş èõřăžŇæĹĹ
('/6.mp3', 'éĆŚăÿ;ăŘŽ', 'ă;ĒăĎĹăžžéŤĹăžĚ')
/6.mp3 éĆŚăÿ;ăŘŽ ä;ĒăĎĹăžžéŤĹăžĚ
```

ăÿŇéĹăĹĹăĜžăŽĹ'énÿĕĹăşĆăĂĈăĹŤăĈĕĹ'ăĂĹăÿĂĕŭăĹĹ'ă;ăăĂĹĹijŇăşşăĹĹ'æ■ŇăŽşă;ç;őăŠ

```
import re

html = '''<div id="songs-list">
    <h2 class="title">çžŘăĚÿèĂă■Ň</h2>
    <p class="introduction">
        çžŘăĚÿèĂă■ŇăĹŮèăĹ
    </p>
    <ul id="list" class="list-group">
        <li data-view="2">ăÿĂĕŭăÿĹăĹĹ'ă;ă</li>
        <li data-view="7">
            <a href="/2.mp3" singer="äzzèt'd'é;Ř">æšğætűäÿĂăčřčňŚ</
→a>
        </li>
        <li data-view="4" class="active">
            <a href="/3.mp3" singer="é;Řçğę">ăĹĂăžŇéŽŘéčŮ</a>
        </li>
        <li data-view="6"><a href="/4.mp3" singer="beyond">
→ăĚĹ'èĹ'ăšĂæĹĹ</a></li>
        <li data-view="5"><a href="/5.mp3" singer="éŽĹæĚğçŘş">
→èõřăžŇæĹĹ</a></li>
```

(ăÿŇéăĹçžğçž■)


```
import re

content = 'Extra stings Hello 1234567 World_This is a Regex Demo_
↪Extra stings'
content = re.sub('\d+', 'Replacement', content)
print(content)
```

èĹŖăĜžçzŖăđĪăŷžĭĭžExtra stings Hello Replacement World_This is a Regex Demo Ex-
tra stingsăĈ

ăŔŖăžçĭĪŃăĹŕĭĭjŃ123457ăŖŝèçńăŽŖă■ăŷžăĹŖăžŃăĈŝèçAăŽŖă■çŽĐăŔĈăŤŕăžEăĈ

ăĭEăŸŖăĪĹăŷĂăŷĹŮŮèçŸĭĭjŃăçĈăđĪăĹŖăžŃăĈŝèçAçŽĐăŸŖăŕăžăŎŝçŤŖă■ŮçŃăŷŝçŽĐăŤŕăžăŽăĹ

```
import re

content = 'Extra stings Hello 1234567 World_This is a Regex Demo_
↪Extra stings'
content = re.sub('(\d+)', r'\1 8910', content)
print(content)
```

èĹŖăĜžçzŖăđĪăŸŕĭĭjžExtra stings Hello 1234567 8910 World_This is a Regex Demo
Extra stings

ăĹŖăžŃăŕăžă■ăĹŽăĹèĹăĭjŔèçAăŃzéĚ■çŽĐăĖĚăŏžăĹăŷĂăŷĹăŃăŕŮĭĭjŃăĈŽèçĜăžŃăĹ■çŽĐăĭŃă■

ăĭŖçŤĭsubăžŃăŔŎăŕăžăžŃăĹ■findallăŮžăŝŤăŏđăŮăĭĪĹăŷĂăŷĹăŮŕçŽĐăĂĭèŮĭĭjŃăŖŝăŸŖăĒĹăĹĹăē

```
import re

html = '''<div id="songs-list">
  <h2 class="title">çžŔăĖŷèĂăă■Ń</h2>
  <p class="introduction">
    çžŔăĖŷèĂăă■ŃăĹŮèăĹ
  </p>
  <ul id="list" class="list-group">
    <li data-view="2">ăŷĂăŮŕăŷĹăĪĹăĭă</li>
    <li data-view="7">
      <a href="/2.mp3" singer="ăžžèt'd'ėĭŔ">ăŝğăŧŮăŷĂăçŕçŃŖŖ</
↪a>
    </li>
    <li data-view="4" class="active">
      <a href="/3.mp3" singer="ėĭŔçğç">ăĭĂăžŃéžŔéçŎ</a>
    </li>
    <li data-view="6"><a href="/4.mp3" singer="beyond">
↪ăĒĹăĖĹăŖăŖăĪĹăĭă</a></li>
    <li data-view="5"><a href="/5.mp3" singer="ėžĹăĖğçŔŖ">
↪èŏŕăžŃăĪĭŃ</a></li>
    <li data-view="5">
      <a href="/6.mp3" singer="ėĈŖăŷăĭăŔŽ">ăĭEăĎăžžéçŤŖăžĚ</a>
    </li>
```

(ăŷŃăēăŧçžğç■)

(çžăŸĹéąŧ)

```

    </ul>
</div>' '
html = re.sub('<a.*?>|</a>', '', html)
print(html)
results = re.findall('<li.*?>(.*?)</li>', html, re.S)
print(results)
for result in results:
    print(result.strip())

```

çžŚăđĬăŸřijŽ

```

<div id="songs-list">
  <h2 class="title">çžŘăĚŸĕĂăĕŃ</h2>
  <p class="introduction">
    çžŘăĚŸĕĂăĕŃăĹŮĕăĹ
  </p>
  <ul id="list" class="list-group">
    <li data-view="2">ăŸĂĕŭŕăŸĹăĬĴĹ'ăĭă</li>
    <li data-view="7">
      æšğætŭăŸĂăčřçŇŚ
    </li>
    <li data-view="4" class="active">
      âĴĂăžŇĕžŘĕčŎ
    </li>
    <li data-view="6">ăĚĹ'ĕĴĹ'ăšĂăĬĴĹ</li>
    <li data-view="5">ĕőřăžŇăĬĴŇ</li>
    <li data-view="5">
      äĭĒăđĴăžžĕŦĴăžĚ
    </li>
  </ul>
</div>
['ăŸĂĕŭŕăŸĹăĬĴĹ'ăĭă', '\n          æšğætŭăŸĂăčřçŇŚ\n',
 → '\n          âĴĂăžŇĕžŘĕčŎ\n', 'ăĚĹ'ĕĴĹ'ăšĂăĬĴĹ',
 → 'ĕőřăžŇăĬĴŇ', '\n          äĭĒăđĴăžžĕŦĴăžĚ\n']
ăŸĂĕŭŕăŸĹăĬĴĹ'ăĭă
æšğætŭăŸĂăčřçŇŚ
âĴĂăžŇĕžŘĕčŎ
ăĚĹ'ĕĴĹ'ăšĂăĬĴĹ
ĕőřăžŇăĬĴŇ
äĭĒăđĴăžžĕŦĴăžĚ

```

7.5 re.compile

ĕĚŽăŸĹăŮŹăŸŦăŸřăřĒăĕăĴĴăžăĹĕĴĴăĭĴŦĕĭĴŮĕŕŚăĴŦăĕăĴĴăžăĹĕĴĴăĭĴŦăŕŕžĕśăăĂĴăĴŚăžŇăŦŕăžĕăĴĴă

```
import re

content = '''Hello 1234567 World_This
is a Regex Demo'''
pattern = re.compile('Hello.*Demo', re.S)
result = re.match(pattern, content)
#result = re.match('Hello.*Demo', content, re.S)
print(result)
```

7.5.1 čžČăžă

```
import requests
import re
content = requests.get('https://book.douban.com/').text
pattern = re.compile('<li.*?cover.*?href="(.*?)".*?title="(.*?)".*?
    ↳more-meta.*?author">(.*?)</span>.*?year">(.*?)</span>.*?</li>',
    ↳re.S)
results = re.findall(pattern, content)
for result in results:
    url, name, author, date = result
    author = re.sub('\s', '', author)
    date = re.sub('\s', '', date)
    print(url, name, author, date)
```

CHAPTER 8

Beautiful Soup

èŁZæYřäYÄäYıçAıæt'zâRLæŰzä;ŁçŽDç;ŠéaęèğçæđŘâžŠiijŇad'ĐçŘEénYæŤLüijŇæŤræŇAđ'Žçğ■èğç
åŁL'èčĚiijŽpip install beautifulsoup4

8.1 èğçæđŘâžŠ

解析器	使用方法	优势	劣势
Python标准库	BeautifulSoup(markup, "html.parser")	Python的内置标准库、执行速度适中、文档容错能力强	Python 2.7.3 or 3.2.2)前的版本中文容错能力差
lxml HTML 解析器	BeautifulSoup(markup, "lxml")	速度快、文档容错能力强	需要安装C语言库
lxml XML 解析器	BeautifulSoup(markup, "xml")	速度快、唯一支持XML的解析器	需要安装C语言库
html5lib	BeautifulSoup(markup, "html5lib")	最好的容错性、以浏览器的方式解析文档、生成HTML5格式的文档	速度慢、不依赖外部扩展

image

8.2 åžšæIŇä;ŁçŤÍ

```
html = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title" name="dromouse"><b>The Dormouse's story</b></p>
<p class="story">Once upon a time there were three little sisters;
↳and their names were
<a href="http://example.com/elsie" class="sister" id="link1"><!--
↳Elsie --></a>,
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</
↳a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie
↳</a>;
```

(äYŇéatçžğçz■)

(çz■äyŁéął)

```

and they lived at the bottom of a well.</p>
<p class="story">...</p>
"""
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(soup.prettify())
↪ #æïjâijRâÑŨăžččăĀiijÑĖĠlâĹlêŁŻèaÑăžččăĀeăeăĒĹiijÑăőzéŤžăd'ĎĈŘĚ
print(soup.title.string) #ăřĚtitleăĚĚăőžăžěstringă;ćăijRĚ;ŚăĠž

```

è¿ŠăǦžæȚŁæđIăęĆăŸŃiijŽ

```
<html>
  <head>
    <title>
      The Dormouse's story
    </title>
  </head>
  <body>
    <p class="title" name="dromouse">
      <b>
        The Dormouse's story
      </b>
    </p>
    <p class="story">
      Once upon a time there were three little sisters; and their_
      ↪names were
      <a class="sister" href="http://example.com/elsie" id="link1">
        <!-- Elsie -->
      </a>
      ,
      <a class="sister" href="http://example.com/lacie" id="link2">
        Lacie
      </a>
      and
      <a class="sister" href="http://example.com/tillie" id="link3">
        Tillie
      </a>
      ;
      and they lived at the bottom of a well.
    </p>
    <p class="story">
      ...
    </p>
  </body>
</html>
The Dormouse's story
```

8.3 æăĜčĹéĂĹæŇĹăŽĪ

8.3.1 éĂĹæŇĹăĚČťă

```
html = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title" name="dromouse"><b>The Dormouse's story</b></p>
<p class="story">Once upon a time there were three little sisters;
↳and their names were
<a href="http://example.com/elsie" class="sister" id="link1"><!--
↳Elsie --></a>,
<a href="http://example.com/lacie" class="sister" id="link2">Lacie<
↳a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie
↳</a>;
and they lived at the bottom of a well.</p>
<p class="story">...</p>
"""
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(soup.title) #éĂĹæŇĹă'titleæăĜčĹ;
print(type(soup.title))
print(soup.head) #éĂĹæŇĹă'headæăĜčĹ;
print(soup.p) #éĂĹæŇĹă'pæăĜčĹ;
```

èĹŚăĜžčŚăđĪĲĲž

```
<title>The Dormouse's story</title>
<class 'bs4.element.Tag'>
<head><title>The Dormouse's story</title></head>
<p class="title" name="dromouse"><b>The Dormouse's story</b></p>
```

ăŘřăžčĲŇăĹăŕæăĜčĹéĂĹæŇĹăžĲĲĲžŇăŕžăžŌăĈăđĪæĲĲăđŽăŷĹæăĜčĹčŽĎĕŕĲĲĲžŇăŕĲăĲĲžèĹŚăĜžč

8.3.2 èŌŭăŔŪăŔĲčĝř

èŌŭăŔŪăŔĲčĝřăŇĜčŽĎăŸŕèŌŭăŔŪăăĜčĹčŽĎăŔĲčĝřăĂĆ

```
html = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title" name="dromouse"><b>The Dormouse's story</b></p>
<p class="story">Once upon a time there were three little sisters;
↳and their names were
<a href="http://example.com/elsie" class="sister" id="link1"><!--
↳Elsie --></a>,

```

(ăŷŇéăĲčžĝčžĲ)

(çžăŸŁéął)

```
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie
</a>;
and they lived at the bottom of a well.</p>
<p class="story">...</p>
"""
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(soup.title.name)
```

èŁŚăĜžijŽtitle

èŁŽăŸŕăĹĹăĪĴăđŰăśĆăăĜçŁčŽĎăŘčĝŕăĹŰăŕèŁŚăĜžăĜžăĹăăĆ

8.3.3 èŰăŔŰăśđăĜ

```
html = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title" name="dromouse"><b>The Dormouse's story</b></p>
<p class="story">Once upon a time there were three little sisters;
and their names were
<a href="http://example.com/elsie" class="sister" id="link1"><!--
Elsie --></a>,
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie
</a>;
and they lived at the bottom of a well.</p>
<p class="story">...</p>
"""
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(soup.p.attrs['name'])
print(soup.p['name'])
```

èŁŚăĜžçžŚăđĪĴijŽ

dromouse
dromouse

ăŸđŰçĝăŰăŔŰăśđăĜçŁăśđăĜçŽĎăŦĹăđĪăŸŕăŸĂăăŭçŽĎăĂĈijĹèŰăŔŰăśđăĜçŽĎăĂijij

8.3.5 ąŃăŮéĂL'æNÍ'

8.3.6 ěŘěŁĆçĆźǎŠǺǎŘǎŽěŁĆçĆź

8.3. æǎĞç■¿éĂĹæŃĴǎŽÍ

èĹŚăĜžčžŚăđĪĲijŽ

```
[ '\n          Once upon a time there were three little sisters;
↳and their names were\n          ', <a class="sister" href=
↳"http://example.com/elsie" id="link1">
<span>Elsie</span>
</a>, '\n', <a class="sister" href="http://example.com/lacie" id=
↳"link2">Lacie</a>, ' \n          and\n          ', <a class=
↳"sister" href="http://example.com/tillie" id="link3">Tillie</a>,
↳'\n          and they lived at the bottom of a well.\n          ']
```

äŸŒcontentsæŮžæŸŤäŸ■ăŔŇčŽĐæŸŦchildrenæŸŦäŸŦäŸŦēŦ■ăžčăŽĲijŇčŦĲprintèĹŚăĜžčŽĐèŦĲij

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(soup.p.children)
for i, child in enumerate(soup.p.children):
    print(i, child)
```

èĹŚăĜžĲijŽ

```
<list_iterator object at 0x1064f7dd8>
0
          Once upon a time there were three little sisters; and
↳their names were

1 <a class="sister" href="http://example.com/elsie" id="link1">
<span>Elsie</span>
</a>
2

3 <a class="sister" href="http://example.com/lacie" id="link2">Lacie
↳</a>
4

          and

5 <a class="sister" href="http://example.com/tillie" id="link3">
↳Tillie</a>
6

          and they lived at the bottom of a well.
```

ēŦŸæĪĲŦäŸŦäŸŦēŦ■ăžčăŽĲijŇčŦĲäŸđæŦăĜĲijŇčŦŦăŽđčŽĐčžŚăđĪăžšæŸŦēŦ■ăžčăŽĲčžăđŇčŽĐăĂČčŦ

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(soup.p.descendants)
for i, child in enumerate(soup.p.descendants):
    print(i, child)
```

èĹŚăĜžčžŚăđĪæŸŦĲijŽ


```

<generator object descendants at 0x10650e678>
0
    Once upon a time there were three little sisters; and_
    ↳their names were

1 <a class="sister" href="http://example.com/elsie" id="link1">
<span>Elsie</span>
</a>
2

3 <span>Elsie</span>
4 Elsie
5

6

7 <a class="sister" href="http://example.com/lacie" id="link2">Lacie
    ↳</a>
8 Lacie
9

    and

10 <a class="sister" href="http://example.com/tillie" id="link3">
    ↳Tillie</a>
11 Tillie
12

    and they lived at the bottom of a well.

```

ăžŇăĹĹăĹŚăžňăŖĹăŸŕěŐăŖŪăăăĢčĹéĢňěĹćčŽĎăĚăőžĭĭĤňěĹŽéĢňăĹŚăžňăŖăžŐăăăĢčĹčŽĎă

8.3.7 çĹŮěĹĆčĆăŚŇčĹăĚĹěĹĆčĆă

```

html = ""
<html>
    <head>
        <title>The Dormouse's story</title>
    </head>
    <body>
        <p class="story">
            Once upon a time there were three little sisters; and_
            ↳their names were
                <a href="http://example.com/elsie" class="sister" id=
            ↳"link1">
                    <span>Elsie</span>
                </a>
                <a href="http://example.com/lacie" class="sister" id=
            ↳"link2">Lacie</a>
                and

```

(ăŸŇěăŧčžġčžĹ)

(čžăŸĹéąŧ)

```

        <a href="http://example.com/tillie" class="sister" id=
↪ "link3">Tillie</a>
        and they lived at the bottom of a well.
    </p>
    <p class="story">...</p>
"""
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(soup.a.parent) #ăăăĜčĹčŽĎčĹűèĹĆčĆž

```

èĹŚăĜžčžŚăđĹĵĵŽ

```

<p class="story">
    Once upon a time there were three little sisters; and_
↪ their names were
        <a class="sister" href="http://example.com/elsie" id=
↪ "link1">
<span>Elsie</span>
</a>
<a class="sister" href="http://example.com/lacie" id="link2">Lacie</
↪ a>

    and
        <a class="sister" href="http://example.com/tillie" id=
↪ "link3">Tillie</a>
    and they lived at the bottom of a well.
</p>

```

èĹŽăŸřèŬăŔŮčňăŸĂăŸĹăăĜčĹčŕřžăžŤčŽĎčĹűèĹĆčĆžăĒĒăžăĂĆ

ăŸŇéĹăŝŤčđ'žèŬăŔŮčĕŬăĚĹèĹĆčĆžĵĵŇčĹűèĹĆčĆžăžăŔĹăĚŭčĹűăžŝčŽĎčĹűăžŝèĹĆčĆžĵĵ

```

from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(list(enumerate(soup.a.parents)))

```

èĹŚăĜžčžŚăđĹĵĵŽ

```

[(0, <p class="story">
    Once upon a time there were three little sisters; and_
↪ their names were
        <a class="sister" href="http://example.com/elsie" id=
↪ "link1">
<span>Elsie</span>
</a>
<a class="sister" href="http://example.com/lacie" id="link2">Lacie</
↪ a>

    and
        <a class="sister" href="http://example.com/tillie" id=
↪ "link3">Tillie</a>
    and they lived at the bottom of a well.

```

(ăŸŇéăŧčžččĹ)

(çžăŸĹăŹ)

```

        </p>), (1, <body>
<p class="story">
    Once upon a time there were three little sisters; and_
    ↳their names were
        <a class="sister" href="http://example.com/elsie" id=
    ↳"link1">
<span>Elsie</span>
</a>
<a class="sister" href="http://example.com/lacie" id="link2">Lacie</
    ↳a>

        and
        <a class="sister" href="http://example.com/tillie" id=
    ↳"link3">Tillie</a>
        and they lived at the bottom of a well.
    </p>
<p class="story">...</p>
</body>), (2, <html>
<head>
<title>The Dormouse's story</title>
</head>
<body>
<p class="story">
    Once upon a time there were three little sisters; and_
    ↳their names were
        <a class="sister" href="http://example.com/elsie" id=
    ↳"link1">
<span>Elsie</span>
</a>
<a class="sister" href="http://example.com/lacie" id="link2">Lacie</
    ↳a>

        and
        <a class="sister" href="http://example.com/tillie" id=
    ↳"link3">Tillie</a>
        and they lived at the bottom of a well.
    </p>
<p class="story">...</p>
</body></html>), (3, <html>
<head>
<title>The Dormouse's story</title>
</head>
<body>
<p class="story">
    Once upon a time there were three little sisters; and_
    ↳their names were
        <a class="sister" href="http://example.com/elsie" id=
    ↳"link1">
<span>Elsie</span>
</a>
<a class="sister" href="http://example.com/lacie" id="link2">Lacie</
    ↳a>

```

(ăŸŖăŹăçžăçžă)

(çžăÿĹéą)

```

        and
        <a class="sister" href="http://example.com/tillie" id=
↪"link3">Tillie</a>
        and they lived at the bottom of a well.
    </p>
<p class="story">...</p>
</body></html>)]

```

8.3.8 âĖĎăĭşëĹĈçĈ

```

html = """
<html>
    <head>
        <title>The Dormouse's story</title>
    </head>
    <body>
        <p class="story">
            Once upon a time there were three little sisters; and_
↪their names were
            <a href="http://example.com/elsie" class="sister" id=
↪"link1">
                <span>Elsie</span>
            </a>
            <a href="http://example.com/lacie" class="sister" id=
↪"link2">Lacie</a>
            and
            <a href="http://example.com/tillie" class="sister" id=
↪"link3">Tillie</a>
            and they lived at the bottom of a well.
        </p>
        <p class="story">...</p>
"""
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(list(enumerate(soup.a.next_siblings)))
print(list(enumerate(soup.a.previous_siblings)))

```

èĹŞăĜžçžŞăđĬJ

```

[(0, '\n'), (1, <a class="sister" href="http://example.com/lacie"
↪id="link2">Lacie</a>), (2, ' \n                and\n                '),
↪(3, <a class="sister" href="http://example.com/tillie" id="link3">
↪Tillie</a>), (4, '\n                and they lived at the bottom of a_
↪well.\n                ')]
[(0, '\n                Once upon a time there were three little_
↪sisters; and their names were\n                ')]

```

8.4 æăĜăĜĚéĂĹæŇĹăŽĪ

8.4.1 find_all(name , attrs , recursive , text , **kwargs)

ăŘŕăăžăĕőăĜčĹăŘăăĀăśđăĀĝăĀăĔăőžăšăĹăŰăăç

name

```
html='''
<div class="panel">
  <div class="panel-heading">
    <h4>Hello</h4>
  </div>
  <div class="panel-body">
    <ul class="list" id="list-1">
      <li class="element">Foo</li>
      <li class="element">Bar</li>
      <li class="element">Jay</li>
    </ul>
    <ul class="list list-small" id="list-2">
      <li class="element">Foo</li>
      <li class="element">Bar</li>
    </ul>
  </div>
</div>
'''

from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(soup.find_all('ul'))
print(type(soup.find_all('ul')[0]))
```

ěĹŚăĜžčžŚăđĪĭjŽ

```
[<ul class="list" id="list-1">
<li class="element">Foo</li>
<li class="element">Bar</li>
<li class="element">Jay</li>
</ul>, <ul class="list list-small" id="list-2">
<li class="element">Foo</li>
<li class="element">Bar</li>
</ul>]
<class 'bs4.element.Tag'>
```

ăŘŕăžčĪŇăĹŕěĹăĜžčŽĐăŸŕăĹŰăăĹăăĭjŔăĂĆ

ěĹŸăŘŕăžěéĂžěĹĜăĀăŎĚčŽĐăŰžăĭjŔăĹăŕŔăŔŰăĔČčŕăĭjŽ

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
for ul in soup.find_all('ul'):
    print(ul.find_all('li'))
```

èĹŚăĜžijŽ

```
[<li class="element">Foo</li>, <li class="element">Bar</li>, <li_
→class="element">Jay</li>]
[<li class="element">Foo</li>, <li class="element">Bar</li>]
```

èĚŽæăüijŇæĪèĚŽèĀŇăśĆăśĆăĭŇăĉŮĉŽĎæšĕæĹĹăĂĆ

attrs

èĕAæśĆăijăèĹŚăŔĆæŤŕĉŽĎĉśăđŇăžŤerĕæŸŕăŸĂăŸĹă■ŮăĚŸăĭĉăijŔijŇæŇĹĉĚġĕŤôăĂijăŕĉĉŽĎġĎăĹ

```
html='''
<div class="panel">
  <div class="panel-heading">
    <h4>Hello</h4>
  </div>
  <div class="panel-body">
    <ul class="list" id="list-1" name="elements">
      <li class="element">Foo</li>
      <li class="element">Bar</li>
      <li class="element">Jay</li>
    </ul>
    <ul class="list list-small" id="list-2">
      <li class="element">Foo</li>
      <li class="element">Bar</li>
    </ul>
  </div>
</div>
'''

from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(soup.find_all(attrs={'id': 'list-1'}))
print(soup.find_all(attrs={'name': 'elements'}))
```

èĹŚăĜžĉžŚăđĪijŽ

```
[<ul class="list" id="list-1" name="elements">
<li class="element">Foo</li>
<li class="element">Bar</li>
<li class="element">Jay</li>
</ul>]
[<ul class="list" id="list-1" name="elements">
<li class="element">Foo</li>
```

(ăŸŇéăĉĉžġĉž■)

(çžăŸĹéą)

```
<li class="element">Bar</li>
<li class="element">Jay</li>
</ul>]
```

ăđđçŖăŖŇăăüçŽĐăĹşèĈ;èĹŸăIJĹăŸĂăŸĹăŽt çđĂă■ŤçŽĐăĹđăşŤiijŇăŸăéIJĂèĹĂă;ĹçŤĹatrsiijŽ

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(soup.find_all(id='list-1'))
print(soup.find_all(class_='element'))
```

çžŞăđIJăŸŕiijŽ

```
[<ul class="list" id="list-1">
<li class="element">Foo</li>
<li class="element">Bar</li>
<li class="element">Jay</li>
</ul>]
[<li class="element">Foo</li>, <li class="element">Bar</li>, <li_
↪class="element">Jay</li>, <li class="element">Foo</li>, <li class=
↪"element">Bar</li>]
```

èĹŽéĜŇiijŇăIJĹăŸĂăŸĹéIJĂèĹĂăşĹăĎŖçŽĐăIJŖăŰăŸăŰclassăŸŕpythonéĜŇéĹççŽĐăŸĂăŸĹăĖşéŤ

text

textăŖşăŸŖăăžă■ŖăŰĜăIJăăĖĖăăđžèĹŽăąŇéĂĹăŇŖăĂĈéIJĂèĹĂăşĹăĎŖçŽĐăŸŕiijŇèĹŽéĜŇçŽĐăş

```
html='''
<div class="panel">
  <div class="panel-heading">
    <h4>Hello</h4>
  </div>
  <div class="panel-body">
    <ul class="list" id="list-1">
      <li class="element">Foo</li>
      <li class="element">Bar</li>
      <li class="element">Jay</li>
    </ul>
    <ul class="list list-small" id="list-2">
      <li class="element">Foo</li>
      <li class="element">Bar</li>
    </ul>
  </div>
</div>
'''
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(soup.find_all(text='Foo'))
```

èĹŚăĜžçzŚæđIJæŸřijŽ[âĂŸFooâĂŽ, âĂŸFooâĂŽ]

8.4.2 find(name , attrs , recursive , text , **kwargs)

findèĚTăŽđă■TăŸlăĚČt'ăüijŇfind_allèĚTăŽđăL'ĂæIJL'ăĚČt'ă

findèĚTăŽđăLřçŽĐæŸřăŇzéĚ■çŽĐçňňăŸĂăŸlăĂijăĂĆăĚüăžŮçŤlăşŤăŇfind_allæŸřăŋăŇĂĚlăŸĂæăŮç

8.4.3 find_parents() find_parent()

find_parents()èĚTăŽđăL'ĂæIJL'çĚŮăĚĹèĹĆçĆžüijŇfind_parent()èĚTăŽđçŽt'æŎĚĹüèĹĆçĆž

8.4.4 find_next_siblings() find_next_sibling()

find_next_siblings()èĚTăŽđăŔŎĚlăL'ĂæIJL'ăĚĐăijşèĹĆçĆžüijŇfind_next_sibling()èĚTăŽđăŔŎĚ

8.4.5 find_previous_siblings() find_previous_sibling()

find_previous_siblings()èĚTăŽđăL'■ĚlăL'ĂæIJL'ăĚĐăijşèĹĆçĆžüijŇfind_previous_sibling()èĚTă

8.4.6 find_all_next() find_next()

find_all_next()èĚTăŽđèĹĆçĆžăŔŎæL'ĂæIJL'çņăŔĹæĹăžžüçŽĐèĹĆçĆž,
find_next()èĚTăŽđçňňăŸĂăŸlçņăŔĹæĹăžžüçŽĐèĹĆçĆž

8.4.7 find_all_previous() áŇ find_previous()

find_all_previous()èĚTăŽđèĹĆçĆžăŔŎæL'ĂæIJL'çņăŔĹæĹăžžüçŽĐèĹĆçĆž,
find_previous()èĚTăŽđçňňăŸĂăŸlçņăŔĹæĹăžžüçŽĐèĹĆçĆž

8.5 CSSéĹ'æŇl'ăŽl

CSSæŸřăŝĆăŔăæăüăijŔèăĹ(èŇŝăŮĜăĚĹçğřijŽCascading Style
Sheets)üijŇæŸřăŸĂçğ■çŤlăĹĚèăĹçŎřHTMLüijĹæăĜăĜĚéĂŽçŤlăăĜĚŋřĚ■ĚĹçŽĐăŸĂăŸlăžŤçŤlřijL'æĹŮXM
éĂŽèĚĜselect()çŽt'æŎĚăijăăĚĚCSSéĹ'æŇl'ăŽlă■şăŔřăŋăŇæĹŔĚĹ'æŇl'ăĂĆ


```
html='''
<div class="panel">
  <div class="panel-heading">
    <h4>Hello</h4>
  </div>
  <div class="panel-body">
    <ul class="list" id="list-1">
      <li class="element">Foo</li>
      <li class="element">Bar</li>
      <li class="element">Jay</li>
    </ul>
    <ul class="list list-small" id="list-2">
      <li class="element">Foo</li>
      <li class="element">Bar</li>
    </ul>
  </div>
</div>
'''

from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
print(soup.select('.panel .panel-heading'))
print(soup.select('ul li'))
print(soup.select('#list-2 .element'))
print(type(soup.select('ul')[0]))
```

èĹŚăĜžčžŚæđĬæŸřijŽ

```
[<div class="panel-heading">
<h4>Hello</h4>
</div>]
[<li class="element">Foo</li>, <li class="element">Bar</li>, <li_
→class="element">Jay</li>, <li class="element">Foo</li>, <li class=
→"element">Bar</li>]
[<li class="element">Foo</li>, <li class="element">Bar</li>]
<class 'bs4.element.Tag'>
```

èĹŽéĜŇăžŇčž■ăÿĂăÿŇăĜăçĝ■selectčŽĎéĂĹ'æŇĹ'æŮžăijŘăĂĆ

čňňăÿĂčĝ■ijŽéĂĹ'æŇĹ'paneléĜŇéĹččŽĎpanel-headingæăĜč■ĹăĂĆéĂŽèĹĜčņăŘû.
 âŚŇčĹ'žæăijčŽĎæŮžăijŘèĹŽèăŇéĂĹ'æŇĹ'ăĂĆ

čňňăžŇčĝ■ijŽéĂĹ'æŇĹ'æĹ'ĂæĬĹ'ulæăĜč■ĹéĜŇéĹččŽĎliæăĜč■ĹăĂĆ

čňňăÿĹ'čĝ■ijŽéĂŽèĹĜidèĹŽèăŇéĂĹ'æŇĹ'ijŇčŇňæĬĹ'čŽĎæăĜăĹŮæŸř#ăĂĆăžŇăŘŎăĹăçĹ'žæăijăŚŇ
 âŚĆăŚĆèĹ■ăžččŽĎæŮžăijŘèĹŸăŘřăžèèĹŽæăuăĹŽřijŽ

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
for ul in soup.select('ul'):
    print(ul.select('li'))
```

èĽŚăĜžčžŚăđĬăŸřijŽ

```
[<li class="element">Foo</li>, <li class="element">Bar</li>, <li_
↪class="element">Jay</li>]
[<li class="element">Foo</li>, <li class="element">Bar</li>]
```

8.5.1 èŌŭăŔŮăśđăĖĖ

```
html='''
<div class="panel">
  <div class="panel-heading">
    <h4>Hello</h4>
  </div>
  <div class="panel-body">
    <ul class="list" id="list-1">
      <li class="element">Foo</li>
      <li class="element">Bar</li>
      <li class="element">Jay</li>
    </ul>
    <ul class="list list-small" id="list-2">
      <li class="element">Foo</li>
      <li class="element">Bar</li>
    </ul>
  </div>
</div>
'''

from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
for ul in soup.select('ul'):
    print(ul['id'])      #ăŸď'čĝ■èŌŭăŔŮăśđăĖĖĖčŽĐăŮžăşŤăŸřăŸĂăăŭčŽĐ
    print(ul.attrs['id'])
```

čžŚăđĬăŸřijŽ

```
list-1
list-1
list-2
list-2
```

8.5.2 èŌŭăŔŮăĖĖăőž

èŌŭăŔŮăăĜč■éĜňĚĬččŽĐăŮĜăĬňăĖĖăőžăĂČ

```
html='''
<div class="panel">
  <div class="panel-heading">
    <h4>Hello</h4>
```

(ăŸŇéăŧčžĝč■)

(čžăŸĹéął)

```

</div>
<div class="panel-body">
  <ul class="list" id="list-1">
    <li class="element">Foo</li>
    <li class="element">Bar</li>
    <li class="element">Jay</li>
  </ul>
  <ul class="list list-small" id="list-2">
    <li class="element">Foo</li>
    <li class="element">Bar</li>
  </ul>
</div>
</div>
'''
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'lxml')
for li in soup.select('li'):
    print(li.get_text())

```

èĹŚăĜžčžŚăđĪăŸřĭjŽ

Foo
 Bar
 Jay
 Foo
 Bar

8.6 æĂžčžŚ

- æŌĺë■ŘăĵčŤĺlxmlèğčăđŘăžŚĭĭjŇěğčăEşăŸ■ăžEĭĭjŇăĹŮèĂĚăăĜč■ăŸăăžăŸăŮŮčŤĺhtml.parser
- æăĜč■ăĚĂĹăŇĬč■ŽéĂĹăĹşèČĭăĭjsăĭEăŸréĂşăžăăĤń
- âžžèőăĭčŤĺfind()ăĂăfind_all() æşèěrcăŇzéĚ■ă■ŤăŸĭčžŚăđĪăĹŮèĂĚăđŹăŸĭčžŚăđĪ
- âĕČăđĪăřžCSSéĂĹăŇĬăŽĬčEşăĈĹăžžèőăĭčŤĺselect()
- èőřăĭŘăŸŸčŤĬčŽĐëŮăŔŮăśđăĂğăŚŇăŮĜăĪŇăĂĭčŽĐăŮžăşŤ

CHAPTER 9

PyQuery

âijžād' ġāŔĹçAŧæt' žčŽĎç; SéąŧèġčæđŘāžŠiijŇăęĆæđIJèġL' â; Ūæ■čāĹZèąĹè; âijŔâEŽèŧûæĹčād'Ĺad'■æĪĆ
ăŏL'èčĚiijŽpip3 install pyquery

9.1 âĹĹăġŇăŇŪ

9.1.1 â■ŪçņęäÿšăĹĹăġŇăŇŪ

```
html = '''
<div>
  <ul>
    <li class="item-0">first item</li>
    <li class="item-1"><a href="link2.html">second item</a></
→li>
    <li class="item-0 active"><a href="link3.html"><span class=
→"bold">third item</span></a></li>
    <li class="item-1 active"><a href="link4.html">fourth item
→</a></li>
    <li class="item-0"><a href="link5.html">fifth item</a></li>
  </ul>
</div>
'''
from pyquery import PyQuery as pq
doc = pq(html)
print(doc('li'))
```

è;ŠăĠžăÿžriijŽ

```
<li class="item-0">first item</li>
    <li class="item-1"><a href="link2.html">second item</a></li>
    <li class="item-0 active"><a href="link3.html"><span class="bold">third item</span></a></li>
    <li class="item-1 active"><a href="link4.html">fourth item</a></li>
    <li class="item-0"><a href="link5.html">fifth item</a></li>
```

èĚŽăŸĹéĀĹ'æŇĹ'ăŽĹæŸřăŸăŸĹCSSéĀĹ'æŇĹ'ăŽĹřijŇăôđčŎřæĹĹăđIJăŠŇBeautiful
SoupéĜŇéĹčŽĎCSSéĀĹ'æŇĹ'ăŽĹæĹĹăđIJæŸřăŸăæăŸčŽĎăĂĆéĀĹ'idčŽĎĹřĹăĹ'■éĹăĹă#ĹijŇăĕĆăđIJéĀĹ'
ĹijŇéĀĹ'æăĜč■ĹčŽĎĹřijŇăĹ'■éĹăžĂăžĹăžšăŸ■ăĹăăĂĆ

9.1.2 URLăĹĹăĝŇăŇŮ

ăĕĆăđIJăĹăăĒĕăŸĂăŸĹéŚĹ;æŎĕčŽĎĹřijŇăĹijŽĕĜĹăĹĹĕřăăśĆéŚĹ;æŎĕĹijŇăžăŸăĹĹéŚĹ;æŎĕăĹ;ŮăĹřčŽĎHT

```
from pyquery import PyQuery as pq
doc = pq(url='http://www.baidu.com')
print(doc('head'))
```

9.1.3 æŮĜăžăŮăĹĹăĝŇăŇŮ

ăĹřăžĕăŇĜăŏŽfilenameĹijŇĕřăăŖŮæŮĜăžăŮĕŮăĹ;ĎčŽĎHTMLăĂĆ

```
from pyquery import PyQuery as pq
doc = pq(filename='demo.html')
print(doc('li'))
```

9.2 áŹŹăĹJŇCSSéĀĹ'æŇĹ'ăŽĹ

```
html = '''
<div id="container">
    <ul class="list">
        <li class="item-0">first item</li>
        <li class="item-1"><a href="link2.html">second item</a></li>
        <li class="item-0 active"><a href="link3.html"><span class="bold">third item</span></a></li>
        <li class="item-1 active"><a href="link4.html">fourth item</a></li>
        <li class="item-0"><a href="link5.html">fifth item</a></li>
    </ul>
</div>
```

(ăŸŇéăĹčžĝčž■)

(çžăŸĹéął)

```
'''
from pyquery import PyQuery as pq
doc = pq(html)
print(doc('#container .list li'))
```

èĹŚăĜžçžŚăđĪæŸřijŽ

```
<li class="item-0">first item</li>
    <li class="item-1"><a href="link2.html">second item</a></
↪li>
    <li class="item-0 active"><a href="link3.html"><span class=
↪"bold">third item</span></a></li>
    <li class="item-1 active"><a href="link4.html">fourth item
↪</a></li>
    <li class="item-0"><a href="link5.html">fifth item</a></li>
```

èĹŽăŸłéĀĹæŃĹæŮžăijŘăsĆăsĆæŌłèĹŽçŽĎæŮžăijŘèĹŽèąŃéĀĹæŃĹăĂĈèĹŽéĜŃæŸřæšæĹĹidăŸžco

9.3 æšěæĹĹăĚČťă

9.3.1 âĹŘăĚČťă

findăijŽæšěæĹĹăĹŚăĹăĚČťăăĜŃéĹčæžăèŸŸCSSéĀĹæŃĹăŽĹčŽĎæĹĂæĪĹăĚČťăăĂĈ

```
html = '''
<div id="container">
    <ul class="list">
        <li class="item-0">first item</li>
        <li class="item-1"><a href="link2.html">second item</a></
↪li>
        <li class="item-0 active"><a href="link3.html"><span class=
↪"bold">third item</span></a></li>
        <li class="item-1 active"><a href="link4.html">fourth item
↪</a></li>
        <li class="item-0"><a href="link5.html">fifth item</a></li>
    </ul>
</div>
'''
from pyquery import PyQuery as pq
doc = pq(html)
items = doc('.list')
print(type(items))
print(items)
lis = items.find('li')
print(type(lis))
print(lis)
```

```
<class 'pyquery.pyquery.PyQuery'>
<ul class="list">
    <li class="item-0">first item</li>
    <li class="item-1"><a href="link2.html">second item</a></li>
    <li class="item-0 active"><a href="link3.html"><span class="bold">third item</span></a></li>
    <li class="item-1 active"><a href="link4.html">fourth item</a></li>
    <li class="item-0"><a href="link5.html">fifth item</a></li>
</ul>

<class 'pyquery.pyquery.PyQuery'>
<li class="item-0">first item</li>
    <li class="item-1"><a href="link2.html">second item</a></li>
    <li class="item-0 active"><a href="link3.html"><span class="bold">third item</span></a></li>
    <li class="item-1 active"><a href="link4.html">fourth item</a></li>
    <li class="item-0"><a href="link5.html">fifth item</a></li>
```

```
lis = items.children()
print(type(lis))
print(lis)
```

```
lis = items.children('.active')
print(lis)
```

æfRäyÄäyIæaGč■;äd' ŪeIcáRlāRrēČ;æIJL'äyÄäyIçLúāĒČçt'äijNāRrāzēēÄŽēfGparent

(äyNéatçzğçz■)

```

<li class="item-1 active"><a href="link4.html">fourth item
↪</a></li>
    <li class="item-0"><a href="link5.html">fifth item</a></li>
</ul>
</div>
'''

from pyquery import PyQuery as pq
doc = pq(html)
items = doc('.list')
container = items.parent()
print(type(container))
print(container)

```

```
<class 'pyquery.pyquery.PyQuery'>
<div id="container">
  <ul class="list">
    <li class="item-0">first item</li>
    <li class="item-1"><a href="link2.html">second item</a></li>
    <li class="item-0 active"><a href="link3.html"><span class="bold">third item</span></a></li>
    <li class="item-1 active"><a href="link4.html">fourth item</a></li>
    <li class="item-0"><a href="link5.html">fifth item</a></li>
  </ul>
</div>
```

```
from pyquery import PyQuery as pq
doc = pq(html)
items = doc('.list')
parents = items.parents()
print(type(parents))
print(parents)
```

```
parent = items.parents('.wrap')
print(parent)
```

```
html = '''
<div class="wrap">
```

52

(çžăŸŁéąŧ)

```
</div>
'''
from pyquery import PyQuery as pq
doc = pq(html)
lis = doc('li').items()
print(type(lis))
for li in lis:
    print(li)
```

ěŁŖăĜžčžŖăđŁŦijŽ

```
<class 'generator'>
<li class="item-0">first item</li>

<li class="item-1"><a href="link2.html">second item</a></li>

<li class="item-0 active"><a href="link3.html"><span class="bold">
→third item</span></a></li>

<li class="item-1 active"><a href="link4.html">fourth item</a></li>

<li class="item-0"><a href="link5.html">fifth item</a></li>
```

9.5 ăŦŭăŔŰăŁăæĀŖ

9.5.1 ăŦŭăŔŰăŖăđăĜ

```
html = '''
<div class="wrap">
    <div id="container">
        <ul class="list">
            <li class="item-0">first item</li>
            <li class="item-1"><a href="link2.html">second item</a>
→</li>
            <li class="item-0 active"><a href="link3.html"><span_
→class="bold">third item</span></a></li>
            <li class="item-1 active"><a href="link4.html">fourth_
→item</a></li>
            <li class="item-0"><a href="link5.html">fifth item</a>
→</li>
        </ul>
    </div>
</div>
'''
from pyquery import PyQuery as pq
doc = pq(html)
```

(ăŸŦéąŧçžčžŖăđ)

(çž■äŸĹéął)

```
a = doc('.item-0.active a')
→ #čĹ'žăăĭjèàĬčđ'žéĂĹ'ăŇĹ'éĜŇéĬćčŽĐăĚĚăăőžĭĭjŇăřśăŸŕăéĜŇéĬćčŽĐăĚĚăăőž
print(a)
print(a.attr('href')) #èŌŭăŔŮăśđăĂġ
print(a.attr.href)
→ #èĚŽăŸđ'çġăŮžăşŦăŸŕăŸĂăăŭçŽĐĭĭjŇéČ;ăŔŕăžžěçŦĹăĬěèŌŭăŔŮăśđăĂġăĚăăŦăŕ
```

èĹŚăĜžăŸžĭĭjŽ

```
<a href="link3.html"><span class="bold">third item</span></a>
link3.html
link3.html
```

9.5.2 èŌŭăŔŮăśđăĂġĹň

```
html = '''
<div class="wrap">
  <div id="container">
    <ul class="list">
      <li class="item-0">first item</li>
      <li class="item-1"><a href="link2.html">second item</a>
→ </li>
      <li class="item-0 active"><a href="link3.html"><span_
→ class="bold">third item</span></a></li>
      <li class="item-1 active"><a href="link4.html">fourth_
→ item</a></li>
      <li class="item-0"><a href="link5.html">fifth item</a>
→ </li>
    </ul>
  </div>
</div>
'''
from pyquery import PyQuery as pq
doc = pq(html)
a = doc('.item-0.active a')
print(a)
print(a.text())
```

èĹŚăĜžăŸžĭĭjŽ

```
<a href="link3.html"><span class="bold">third item</span></a>
third item
```

9.5.3 ěŮăăŮHTML

```
html = '''
<div class="wrap">
  <div id="container">
    <ul class="list">
      <li class="item-0">first item</li>
      <li class="item-1"><a href="link2.html">second item</a>
↪</li>
      <li class="item-0 active"><a href="link3.html"><span_
↪class="bold">third item</span></a></li>
      <li class="item-1 active"><a href="link4.html">fourth_
↪item</a></li>
      <li class="item-0"><a href="link5.html">fifth item</a>
↪</li>
    </ul>
  </div>
</div>
'''
from pyquery import PyQuery as pq
doc = pq(html)
li = doc('.item-0.active')
print(li)
print(li.html())
```

ěŚăĹăžăŸijŽ

```
<li class="item-0 active"><a href="link3.html"><span class="bold">
↪third item</span></a></li>

<a href="link3.html"><span class="bold">third item</span></a>
```

9.6 DOMăŚăĹJ

9.6.1 addClassăĀremoveClass

```
html = '''
<div class="wrap">
  <div id="container">
    <ul class="list">
      <li class="item-0">first item</li>
      <li class="item-1"><a href="link2.html">second item</a>
↪</li>
      <li class="item-0 active"><a href="link3.html"><span_
↪class="bold">third item</span></a></li>
      <li class="item-1 active"><a href="link4.html">fourth_
↪item</a></li>
```

(ăŸŹăĹăžăŸijŽ)

(çžăŸŁéął)

```

        <li class="item-0"><a href="link5.html">fifth item</a>
↪</li>
        </ul>
    </div>
</div>
'''
from pyquery import PyQuery as pq
doc = pq(html)
li = doc('.item-0.active')
print(li)
li.removeClass('active')
print(li)
li.addClass('active')
print(li)

```

èŁŖăĜžăŸžijŽ

```

<li class="item-0 active"><a href="link3.html"><span class="bold">
↪third item</span></a></li>

<li class="item-0"><a href="link3.html"><span class="bold">third_
↪item</span></a></li>

<li class="item-0 active"><a href="link3.html"><span class="bold">
↪third item</span></a></li>

```

9.6.2 äŔŖăĜžăŸžijŽ

```

html = '''
<div class="wrap">
    <div id="container">
        <ul class="list">
            <li class="item-0">first item</li>
            <li class="item-1"><a href="link2.html">second item</a>
↪</li>
            <li class="item-0 active"><a href="link3.html"><span_
↪class="bold">third item</span></a></li>
            <li class="item-1 active"><a href="link4.html">fourth_
↪item</a></li>
            <li class="item-0"><a href="link5.html">fifth item</a>
↪</li>
        </ul>
    </div>
</div>
'''
from pyquery import PyQuery as pq
doc = pq(html)

```

(ăŸŔéąçžğçž)

(çäýŁęą)

```
li = doc('.item-0.active')
print(li)
li.attr('name', 'link')
#åĊđIJæšæIJL' nameåśđæĀġiijŃäijŽæŭžåŁăname=linkiijŃåĊđIJæIJŃæĬěæIJL' nameåśđæĀġiijŃäijŽæŭžåŁă
print(li)
li.css('font-size', '14px')
print(li)
```

èŁŞåĠžäýžijŽ

```
<li class="item-0 active"><a href="link3.html"><span class="bold">
#third item</span></a></li>

<li class="item-0 active" name="link"><a href="link3.html"><span_
#class="bold">third item</span></a></li>

<li class="item-0 active" name="link" style="font-size: 14px"><a_
#href="link3.html"><span class="bold">third item</span></a></li>
```

9.6.3 remove

```
html = '''
<div class="wrap">
  Hello, World
  <p>This is a paragraph.</p>
</div>
'''
from pyquery import PyQuery as pq
doc = pq(html)
wrap = doc('.wrap')
print(wrap.text())
wrap.find('p').remove()
#åĬŁĉŤĬĬfindæŮžæşŤæL' ħåĬŕpæăĠĉĬiijŃăžŃăŔŌĉŤĬremoveçġžéŽd' pæăĠĉĬ
print(wrap.text())
```

èŁŞåĠžäýžijŽ

```
Hello, World This is a paragraph.
Hello, World
```

9.6.4 åĬŮăžŮDOMæŮžæşŤ

<http://pyquery.readthedocs.io/en/latest/api.html>

9.7 äijĹčŝžéĀĹæŇĖăŽĪ

âĖŖăžžčĹĲCSS3čŽĎăŸĂăžŽäijĹčŝžéĀĹéąžăŽĲijŇéĀĹæŇĖăŸĂăžŽčĹŹăŏŽčŽĎăĖĈčĹăăĀĈ

```
html = '''
<div class="wrap">
  <div id="container">
    <ul class="list">
      <li class="item-0">first item</li>
      <li class="item-1"><a href="link2.html">second item</a>
↪</li>
      <li class="item-0 active"><a href="link3.html"><span_
↪class="bold">third item</span></a></li>
      <li class="item-1 active"><a href="link4.html">fourth_
↪item</a></li>
      <li class="item-0"><a href="link5.html">fifth item</a>
↪</li>
    </ul>
  </div>
</div>
'''

from pyquery import PyQuery as pq
doc = pq(html)
li = doc('li:first-child')
print(li)
li = doc('li:last-child')
print(li)
li = doc('li:nth-child(2)')
↪#ăŇĠăŏŽčĹŹăĲijŤéąžăžŖĲijŇéŐŭăŖŪčŇăžŇăŸĲăĖĈčĹăăĠč■ăăĀĈ
print(li)
li = doc('li:gt(2)') #greater_
↪thančŽĎĲijĲăĖŽĲijŇéŐŭăŖŪăžŖăŖŭăŖŤ2ăĎŹčŽĎăĠč■ă
print(li)
li = doc('li:nth-child(2n)')
↪#éŐŭăŖŪăĀŭăŤŖăăĠč■ăĲijŇăĲčăĎIJăŸŖăăĠč■ăŤŖăŖăžžăăĖŽăĲŖ2n+1
print(li)
li = doc('li:contains(second)') #éŐŭăŖŪăŇĠăŏŽăăĖĖăŏŽčŽĎăăĠč■ă
print(li)
```

èĲŖăĠžăŸžĲijŽ

```
<li class="item-0">first item</li>

<li class="item-0"><a href="link5.html">fifth item</a></li>

<li class="item-1"><a href="link2.html">second item</a></li>

<li class="item-1 active"><a href="link4.html">fourth item</a></li>
    <li class="item-0"><a href="link5.html">fifth item</a>
↪</li>
```

(ăŸŇéăĲčžĲ■ă)

(çžăŸŁéął)

```
<li class="item-1"><a href="link2.html">second item</a></li>
    <li class="item-1 active"><a href="link4.html">fourth_
↪item</a></li>

<li class="item-1"><a href="link2.html">second item</a></li>
```

æŽt'ád'ŽCSSéĀL'æŃl'ăŽlăŔfăžěæšěçIJŃ <http://www.w3school.com.cn/css/index.asp>

9.8 áŏŸæŮžæŮĜæąč

<http://pyquery.readthedocs.io/>

CHAPTER 10

çd'žăĭ NĭijŽăĹĭçTĭçĹñèŽñèŎũăRŪăd'ĲæřTăĖæAř

ăĹĲçTĭçĹñèŽñèŎũăRŪăd'ĲæřTăĖæAřĭijNăžũéĂŽèĖĖĆŏăžũæřRăd'ĲăŏŽæŬũăRŚéĂĂăĹçĹ'žăŏŽéĆ
çĹñăRŪçŽĎçĭŚçñŽĭijŽăy■ăŽĭăd'ĲæřTçĭŚ <www.weather.com.cn/weather/101270101.shtmlĂă>

10.1 äzzăĹăĹĖæđŘ

æĹŚăžnăRřăžěăřĖăžzăĹăĹĖăyžăžăyNăyĹăyĲéĆĲăĹĖĭijŽ

1. çĹñăRŪăd'ĲæřTăĖæAř
2. ăRŚéĂĂéĆŏăžũ
3. ăŏŽæŬũèĖŽèăNăžzăĹă

10.2 çĲNăžRçijŬăĖŽ

10.2.1 çĹñăRŪăd'ĲæřTăĖæAř

æĹŚăĭjĂăy■ăŽĭăd'ĲæřTçĭŚæĹŘéĆĭăyĆçŽĎăd'ĲæřTéćĎæĹĭijNăRşéTŏèĖŽăĖĖăĲæçĂæşĖăĂĲijNèĖĖ
èĖŽăĖĖăĲĖĲăĲĭijNéĂĲæŲăŭăyĲèğŚçŽĎăĲĲSelect an element in the page to
inspect ităĂĲăŲĲéŚŏĭijNăĹĭăĹăĹăĖşéTŏăĖĖçĲ'ăăĂĆ

ăĖşéTŏăĖĖçĲ'ăéĂĲăRŪ

ăĹĲçTĲBeautifulSoup æĹŚăRŪæŬæĲşăĂĂăd'ĲæřTăŲNăyĲăžăyĲăyĲăĖæAřĭijNăřĖăĖĲăTřăřĂĖĖĖ

```
import requests
from bs4 import BeautifulSoup

def get_weather():

    url = 'http://www.weather.com.cn/weather/101270101.shtml'
    headers = {
        'user-agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64)
        ↳ AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131
        ↳ Safari/537.36'
    }

    res = requests.get(url, headers=headers)
    res.encoding='utf-8'

    soup = BeautifulSoup(res.text, 'html.parser')

    date = soup.find(class_='t').find('h1')
    wea = soup.find(class_='wea')
    tem = soup.find(class_='tem')
    return date, wea, tem
```

10.2.2 äŔŠěÄĚČőäžŮ

äŏĹäĚĹetűëğÄřijŇčĹŇăžŔæĹŠăžñěĹĹÄëĚÄĚĹăŸëçŞăĚĚăŔŠăĹăăžžěČőčőśăŤŇăŕĚçăÄřijŇăŤŭăžŭăžžě
čĹŇăžŔăĹçŤĹqqéČőčőśĹijŇăžŭăĹĹĚĚČőčőśăŸPOP3/SMTPæĹĹăĹăřijŇăŸĹăŏŽëŏŕăĹŔæŐĹæĹČçăĹ

```
import smtplib
from email.mime.text import MIMEText
from email.header import Header

from_addr = input('ĚŕűëçŞăĚĚăŔŠăĹăăžžěČőčőśĹijŽ')
print('\n')
password = input('ĚŕűëçŞăĚĚăŕĚçăĹ')
print('\n')
to_addr = input('ĚŕűëçŞăĚĚăŤŭăĹăăžžěČőčőś')
print('\n')

def send_email(date, wea, tem):
    global from_addr, password, to_addr
    smtp_server = 'smtp.qq.com'

    # äŮçñĚăŸstripæŮžæşŤĹijŽ
    # Strip() æŮžæşŤçŤĹăžŐăĹăéžd'ăĹĹăăğŇăĹŮçžŞăŕççŽďăŮçñĚăĹČ
    # lstrip() | rstirp() äĹĚăĹŇăžŐăŭĚăŔşæĹ'ğĚăŇăĹăéžd'æŞăĹĹĹăĹČ
    #
    ↳ éžŸëŏď'æČĚăĚĹăŸŇăĹijŽăĹăéžd'çĹ'žçŽ;æĹŮĚăĚăçĚăŇçñĚĹijŇăžŞăŔŕăžžăŇĠăŏŽăĚŭăžŭăŮç
```

(äŸŇĚăççžğç)

(čžăŸĹéą)

```

text = 'æĹŘéČ;'+date.text+'\năd' l' æřŤiijž'+wea.text+
↳ '\næřŤæŸl' iijž'+tem.text.strip()

msg = MIMEText(text,'plain','utf-8')
msg['From'] = Header(from_addr)
msg['TO'] = Header(to_addr)
msg['Subject'] = Header('æĹŘéČ;æřŘæŮěăd' l' æřŤ')

server = smtplib.SMTP_SSL(smtp_server)
server.connect(smtp_server,465)
server.login(from_addr,password)

try:
    server.sendmail(from_addr, to_addr, msg.as_string())
    print('éĆőăžűăŘSéĂĀæĹŘăĹS')
except:
    print('éĆőăžűăŘSéĂĀăd' sét' ě')

server.quit()

```

10.2.3 èĠăĹăĹăŘSéĂĀéĆőăžű

æĹSăžňă;čŸŤĹ schedule âNĚæĹěăŮNæĹŘèĠăĹăĹăŘSéĂĀéĆőăžűčŽDăžžăĹăăĀĆ

```

import schedule
import time

def job():
    date,wea,tem = get_weather()
    # çĹňăŮžăd' l' æřŤăĹăăĀř
    send_email(date,wea,tem)
    # âŘSéĂĀéĆőăžű
    print('âŘSéĂĀăŮNæĹŘ')

schedule.every().day.at("7:00").do(job)
# èŮçăŮžæřŘăd' l' 7:00âŘSéĂĀéĆőăžűiijNăŘřăžžěăĹŮæŤžæŮúéŮt' éŮt' éžŤ
while True:
    schedule.run_pending()
    time.sleep(1)

```

10.3 æĹJĂçžĹčĹNăžŘ

čĹNăžŘéĹJĂèçĀăĹĹăĹăĹæĹJžŤŤèDŸăŸĹæNăçžăĹŘăăNæĹăĹč;æřŘăd' l' âŘřăžžăŘSéĂĀéĆőăžűiijNăž

```

import requests
from bs4 import BeautifulSoup

import smtplib
from email.mime.text import MIMEText
from email.header import Header

import schedule
import time

# -----çĹňăŘŮăđ'ł'æřřăĹæAř-----
↳-----
def get_weather():

    url = 'http://www.weather.com.cn/weather/101270101.shtml'
    headers = {
        'user-agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64)
↳AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131
↳Safari/537.36'
    }

    res = requests.get(url, headers=headers)
    res.encoding='utf-8'

    soup = BeautifulSoup(res.text, 'html.parser')

    date = soup.find(class_='t').find('h1')
    wea = soup.find(class_='wea')
    tem = soup.find(class_='tem')
    return date, wea, tem

# -----ăŘŚéĂĂéĆőăžű-----
def send_email(date, wea, tem):
    global from_addr, password, to_addr
    smtp_server = 'smtp.qq.com'

    # ăŮŋĕăŸšstripæŮžæşřiiĵž
    # Strip()æŮžæşřřčřĹăžŎăĹăéžđ'ăijĂăğNăĹŮçzşăř;çžĐăŮŋĕăĂĆ
    # lstrip()|rstrip()ăĹĒăĹŋăžŎăŮĕăŘşăĹ'ğĕăŇăĹăéžđ'æşă;IJăĂĆ
    #
↳éžŸĕŏđ'æČĚăĒăŸŇăijžăĹăéžđ'çĹ'žçž;ăĹŮĕĂĒăĹăçĕăŇċĕĕiiĵŇăžŚăŘřăžĕăŇĞăŎžăĒŮăžŮăŮŮç
    text = 'ăĹŘĕČ;'+date.text+'Ĺăđ'ł'æřřiiĵž'+wea.text+
↳'ĹăřřăŸł'iiĵž'+tem.text.strip()

    msg = MIMEText(text, 'plain', 'utf-8')
    msg['From'] = Header(from_addr)
    msg['TO'] = Header(to_addr)
    msg['Subject'] = Header('ăĹŘĕČ;ăřŘăŮĕăđ'ł'æřř')

    server = smtplib.SMTP_SSL(smtp_server)

```

(ăŸŇĕăçžğçžă)

(çz■äyŁéął)

```
server.connect(smtp_server,465)
server.login(from_addr,password)

try:
    server.sendmail(from_addr, to_addr, msg.as_string())
    print('éĆőăžűâŔŚéĂĀæĹŖăĽ$')
except:
    print('éĆőăžűâŔŚéĂĀăd' set'ě')

server.quit()

from_addr = input('èŕûè;ŞăĚëârŜăfăăžžéĆőÇősiijŽ')
print('\n')
password = input('èŕûè;ŞăĚëârĒçăĀ')
print('\n')
to_addr = input('èŕûè;ŞăĚëăŦűăfăăžžéĆőÇős')
print('\n')

# -----ăōŽæŮú-----
def job():
    date,wea,tem = get_weather()
    # çĹňăŎžăd'l'æřťăfăæĀř
    send_email(date,wea,tem)
    # âŔŚéĂĀéĆőăžű
    print('âŔŚéĂĀăōŇæĹŖ')

schedule.every().day.at("7:00").do(job)
# èőçăôžæŕăd'l'7:00âŔŚéĂĀéĆőăžűii jŇăŔŕăžěăfăœŦzæŮűéŮt'éŮt'éŽŦ
while True:
    schedule.run_pending()
    time.sleep(1)
```