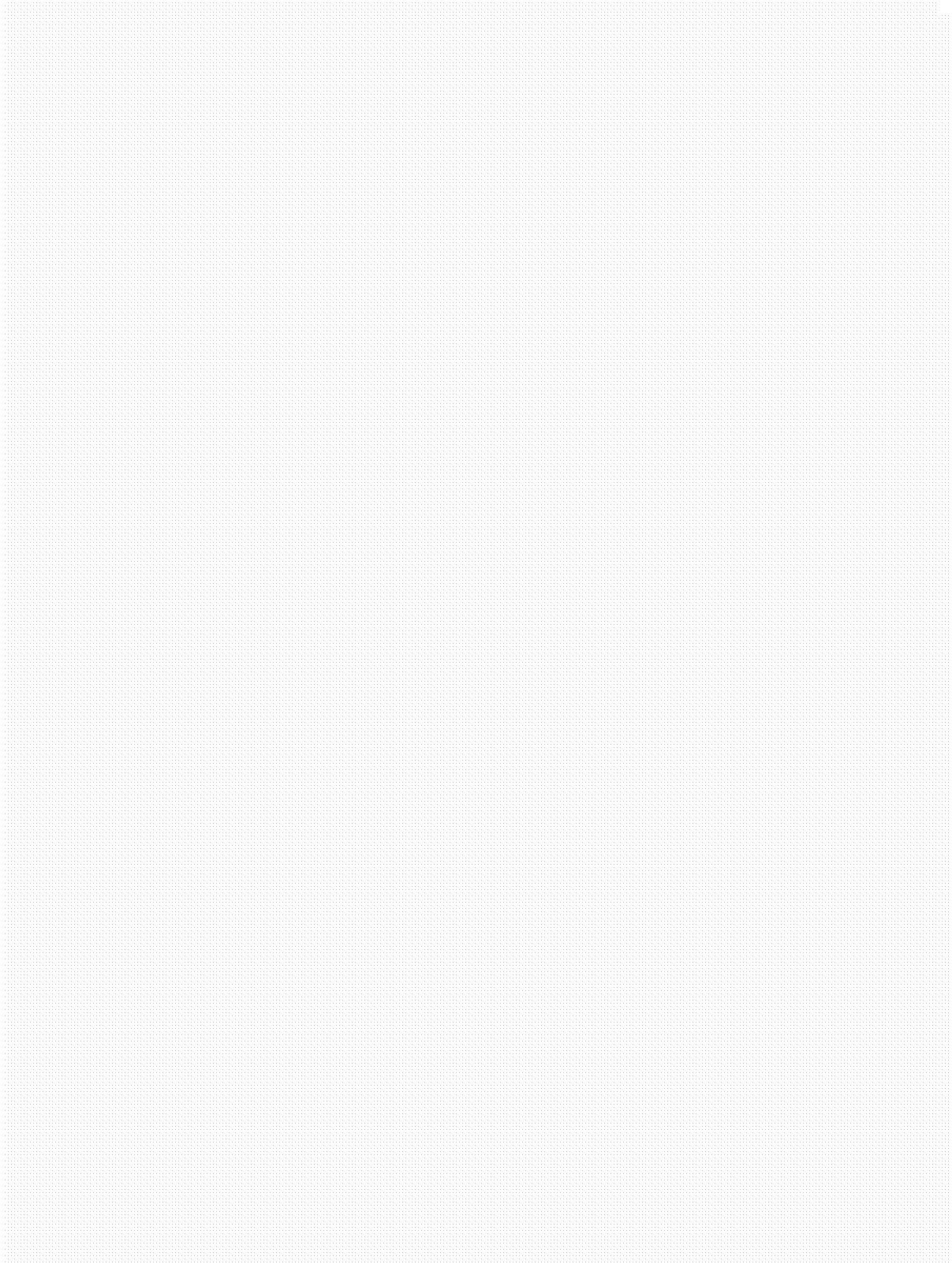
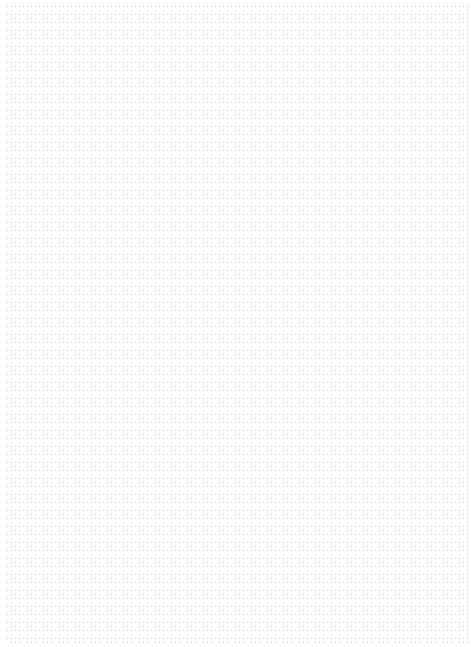


嵩天

www.python123.org

***WS01***



O4X -Tian

## Requests库的安装

Win

平台

:

“以管理员身份运行”

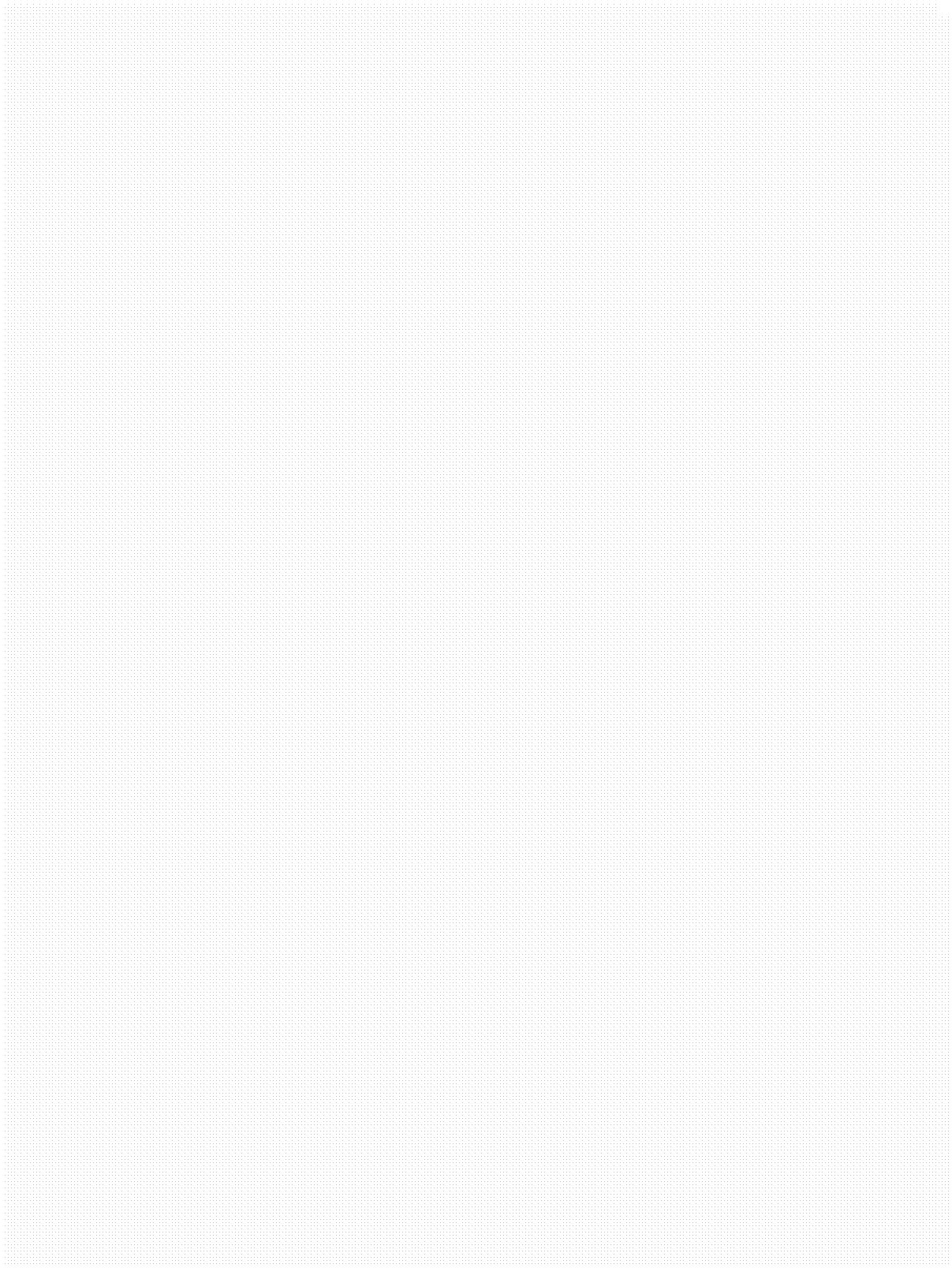
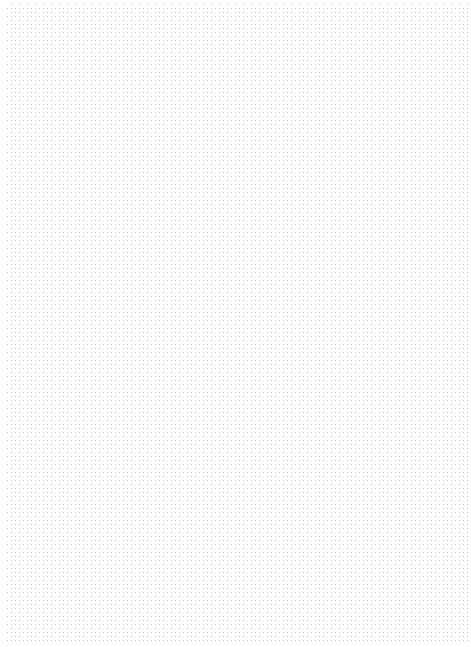
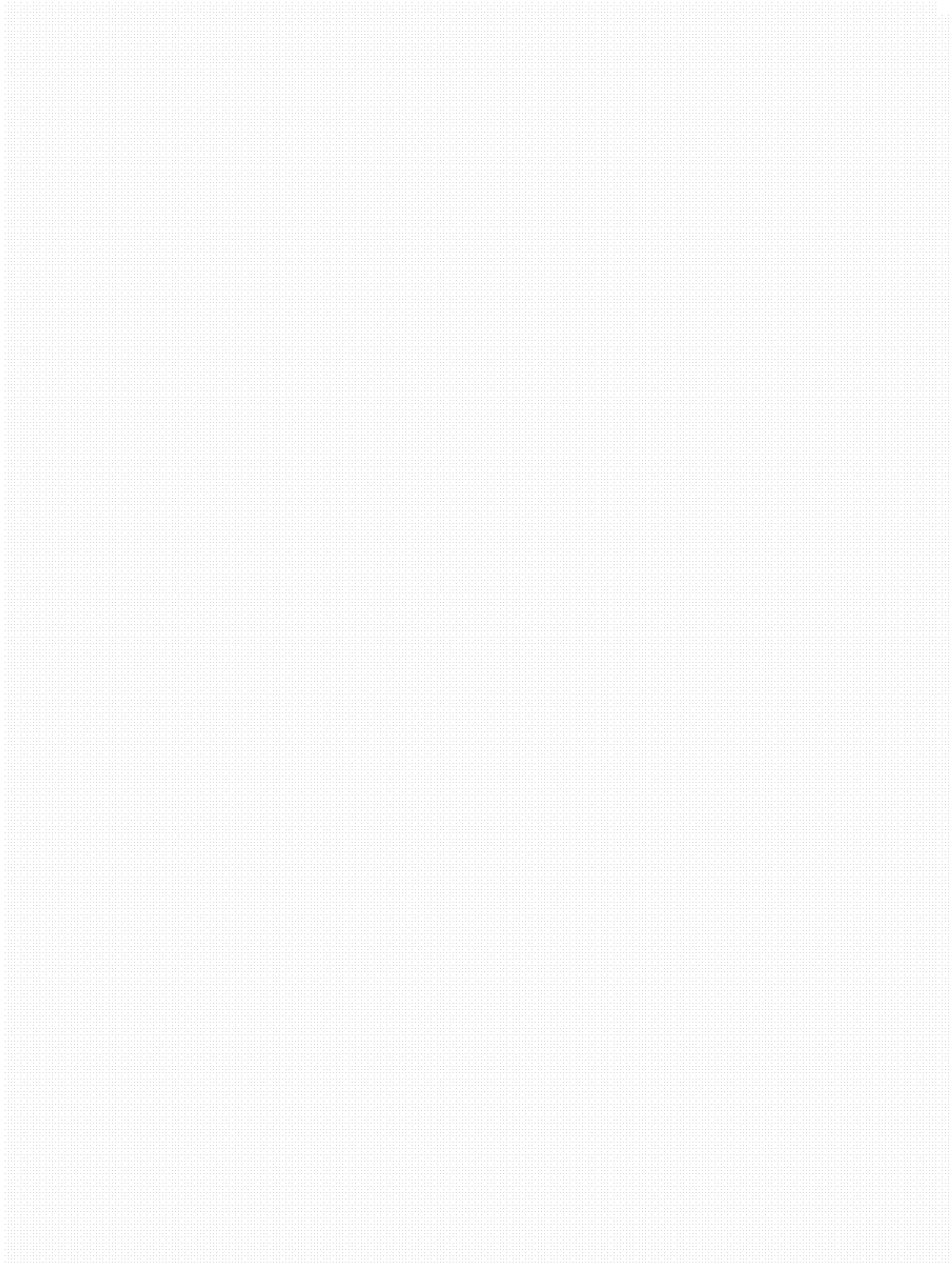
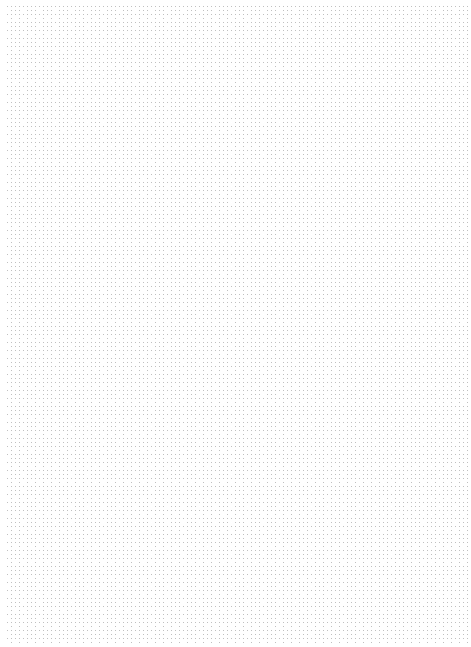
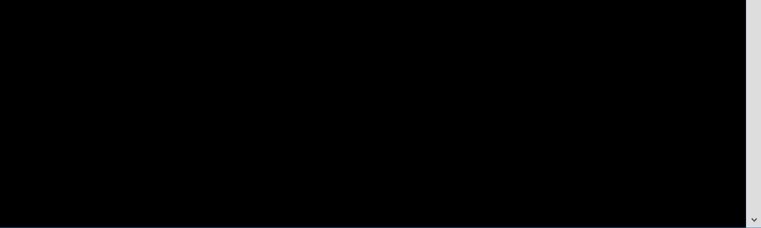
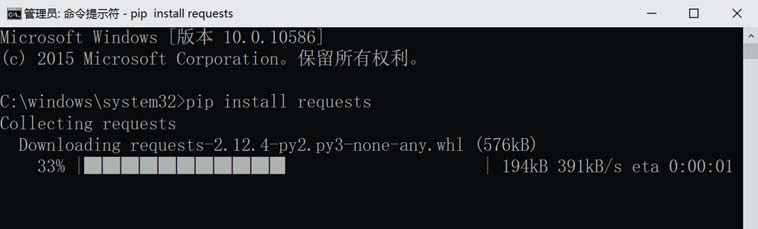
cmd

，执行

pip

install

requests



|  |  |
| --- | --- |
| 方法 | 说明 |
| requests.request() | 构造一个请求，支撑以下各方法的基础方法 |
| requests.get() | 获取HTML网页的主要方法，对应于HTTP的GET |
| requests.head() | 获取HTML网页头信息的方法，对应于HTTP的HEAD |
| requests.post() | 向HTML网页提交POST请求的方法，对应于HTTP的POST |
| requests.put() | 向HTML网页提交PUT请求的方法，对应于HTTP的PUT |
| requests.patch() | 向HTML网页提交局部修改请求，对应于HTTP的PATCH |
| requests.delete() | 向HTML页面提交删除请求，对应于HTTP的DELETE |

## Requsets库的七个方法

requests.get()

r

=

requests.get(url)

Request

构造一个向服务器请求

资源的

Request

对象

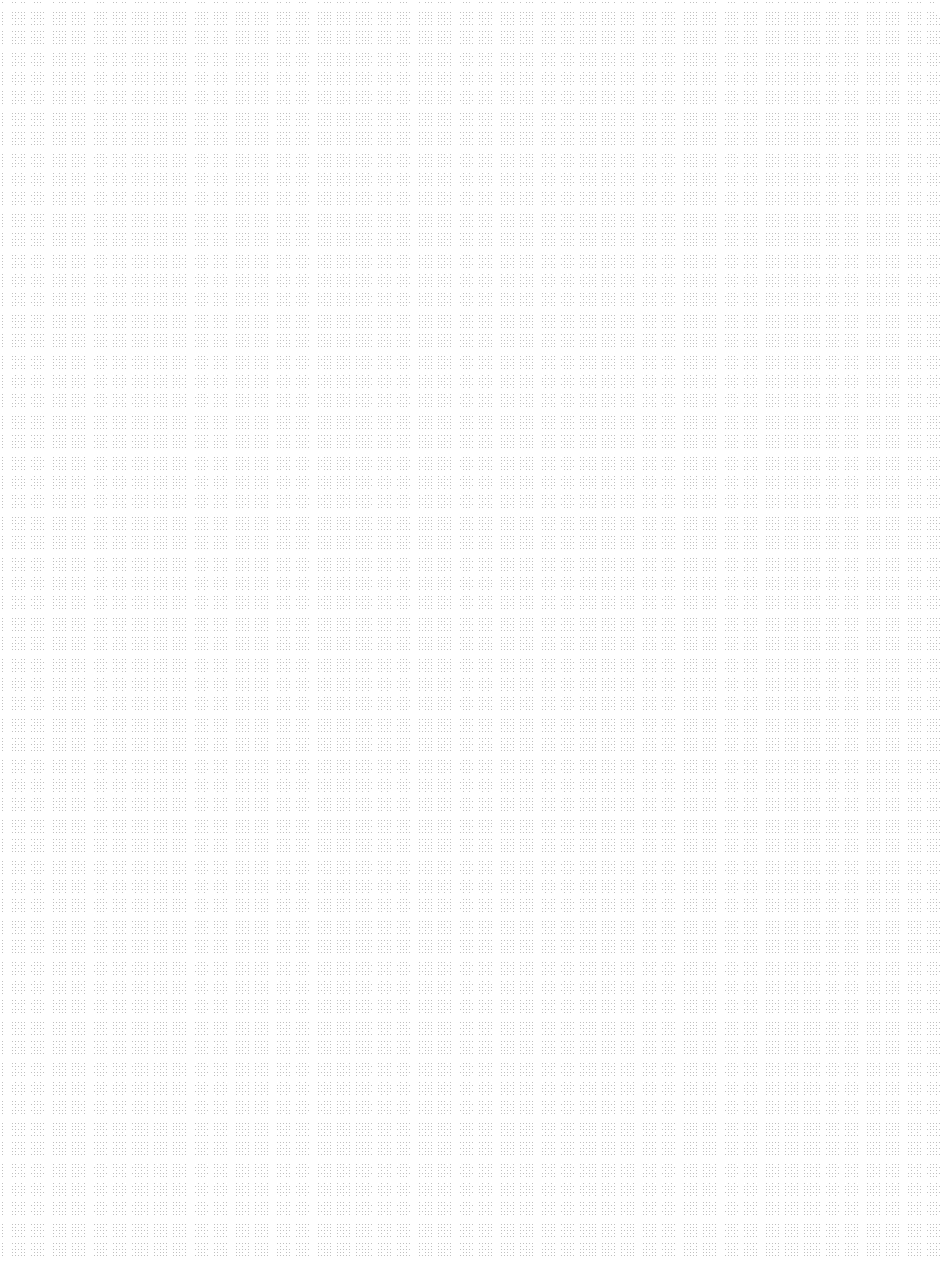
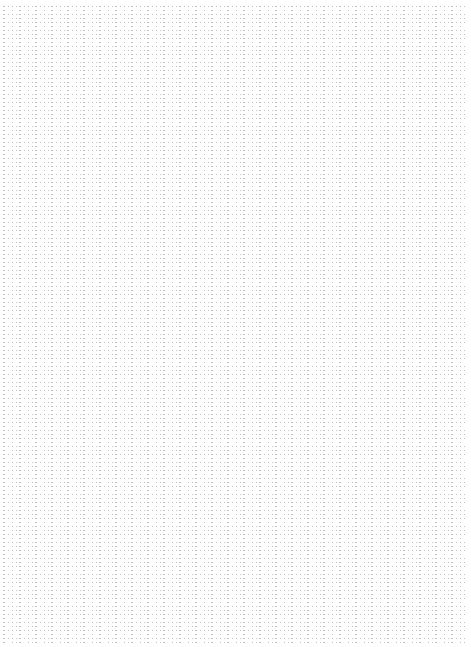
Response

返回一个包含服务器

资源的

Response

对象



**∙ url**

requests.get(

**url**

,

**params=None**

,

**\*\*kwargs**

)

:

拟获取页面的

url

链接

**∙**

**params**

:

url

中的额外参数，字典或字节流格式，可选

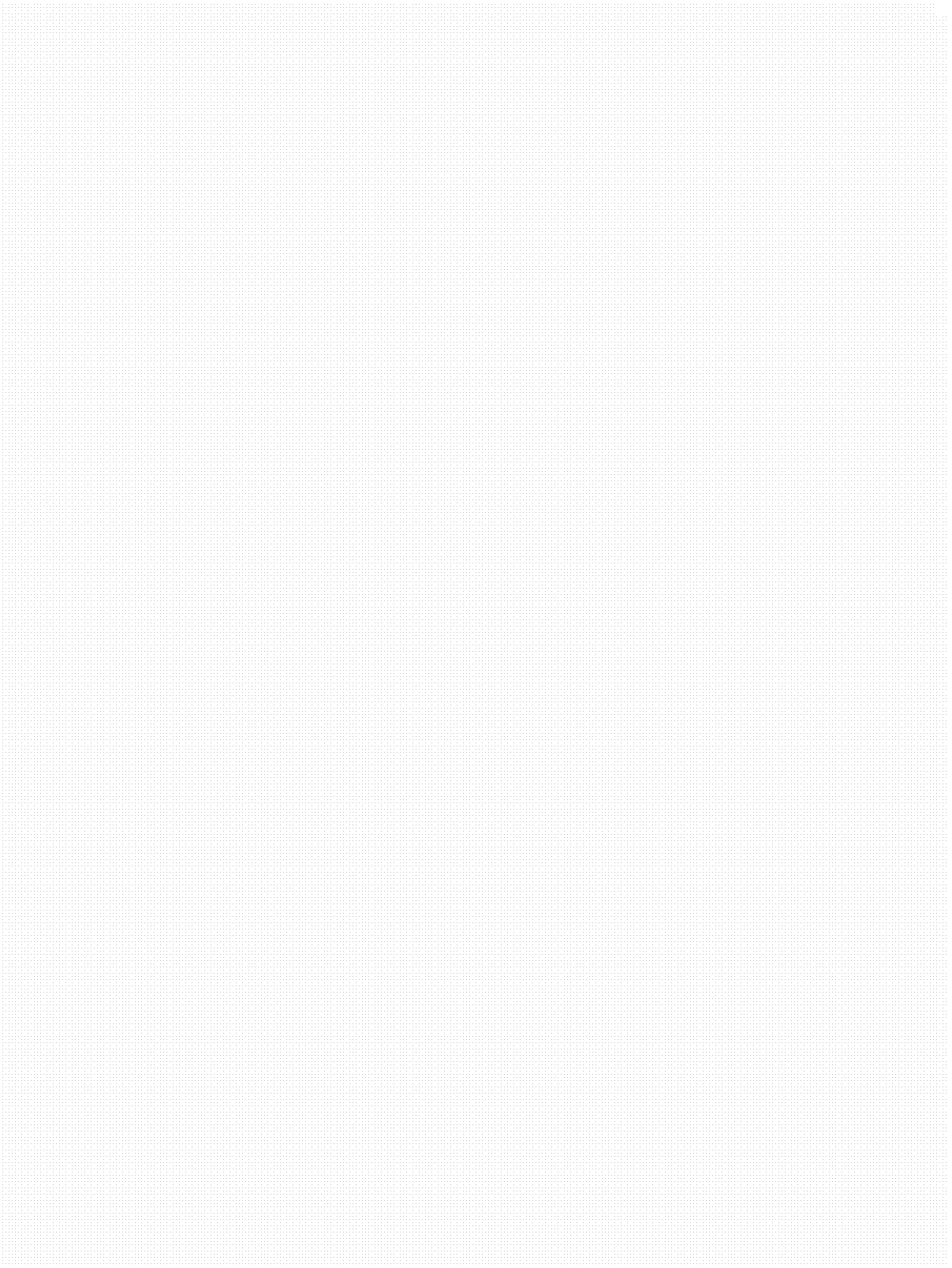
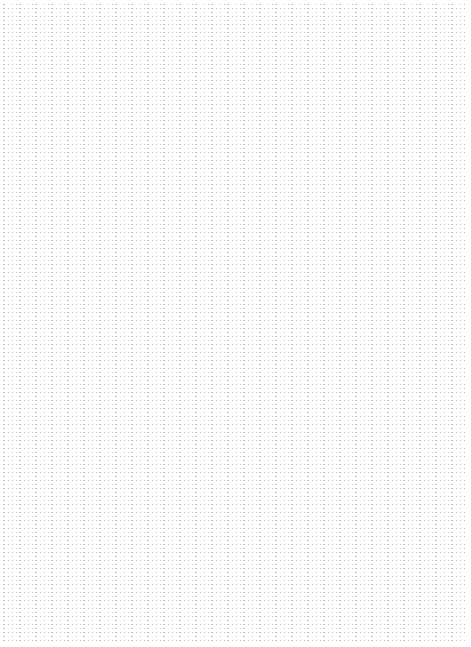
**∙**

**\*\*kwargs**

:

12

个控制访问的参数



requests.get(

**url**

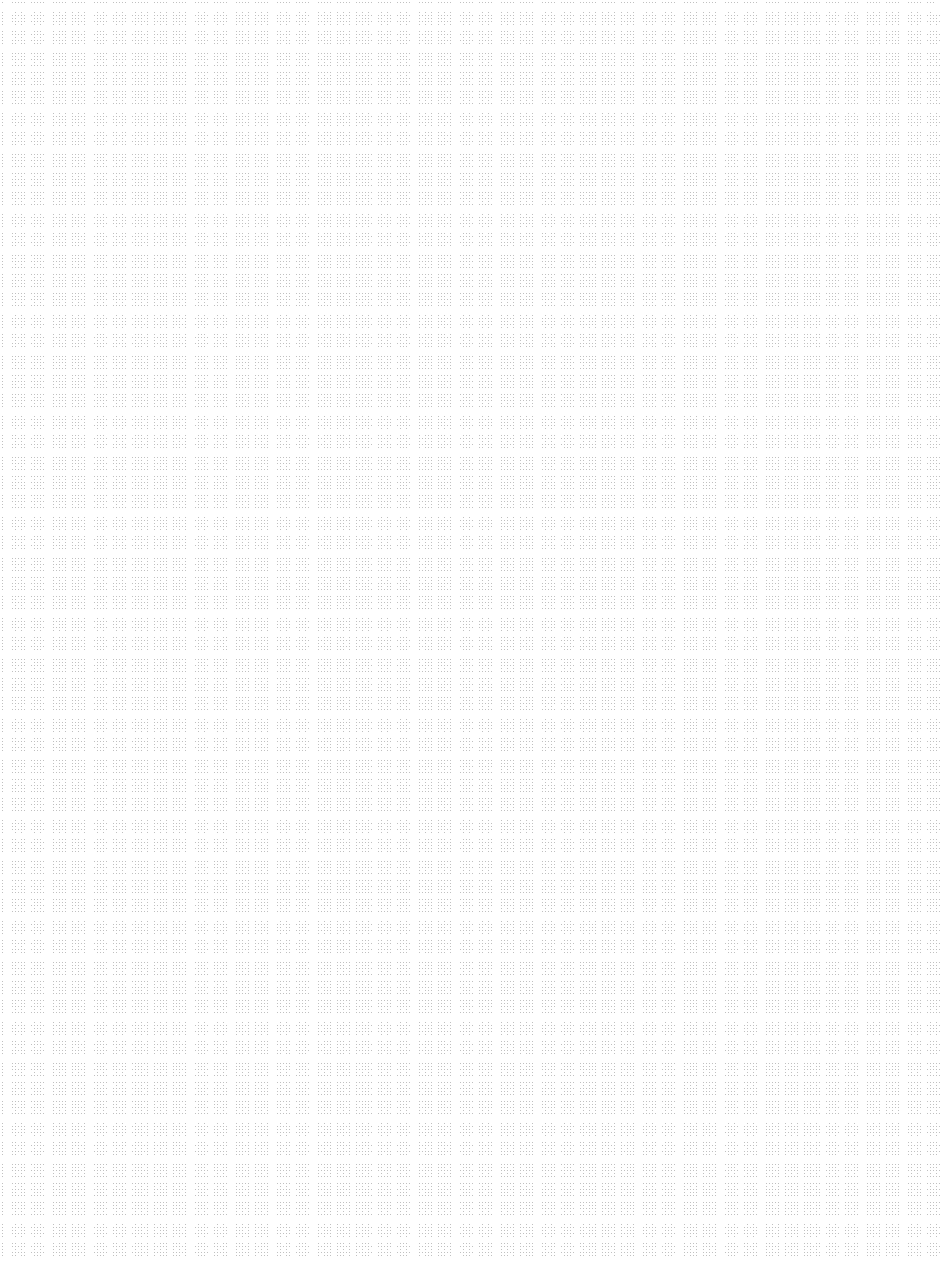
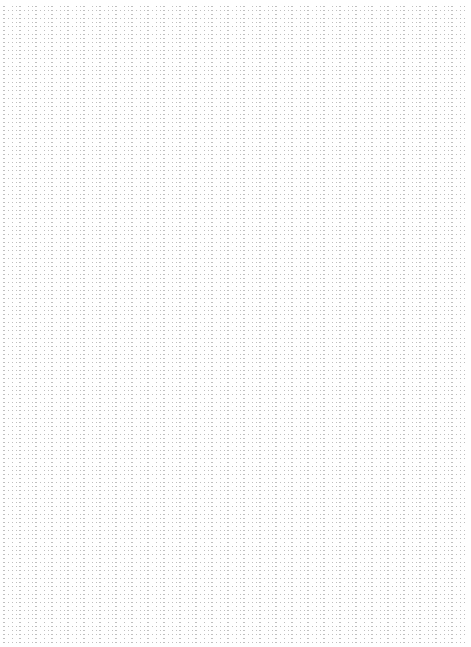
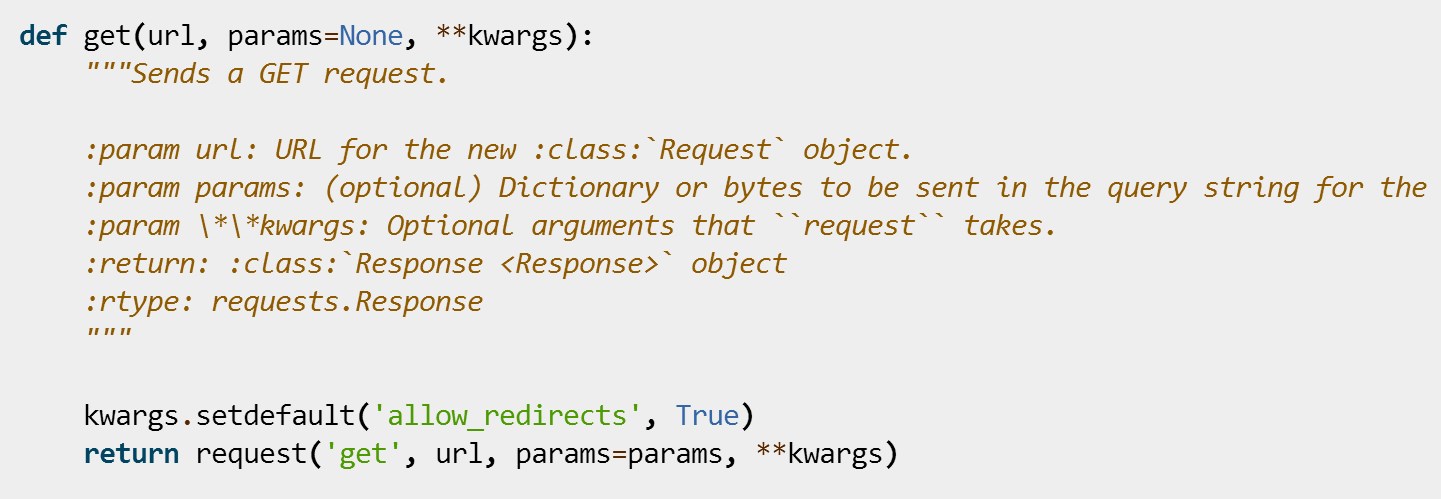
,

**params=None**

,

**\*\*kwargs**

)



r

=

requests.get(url)

Request

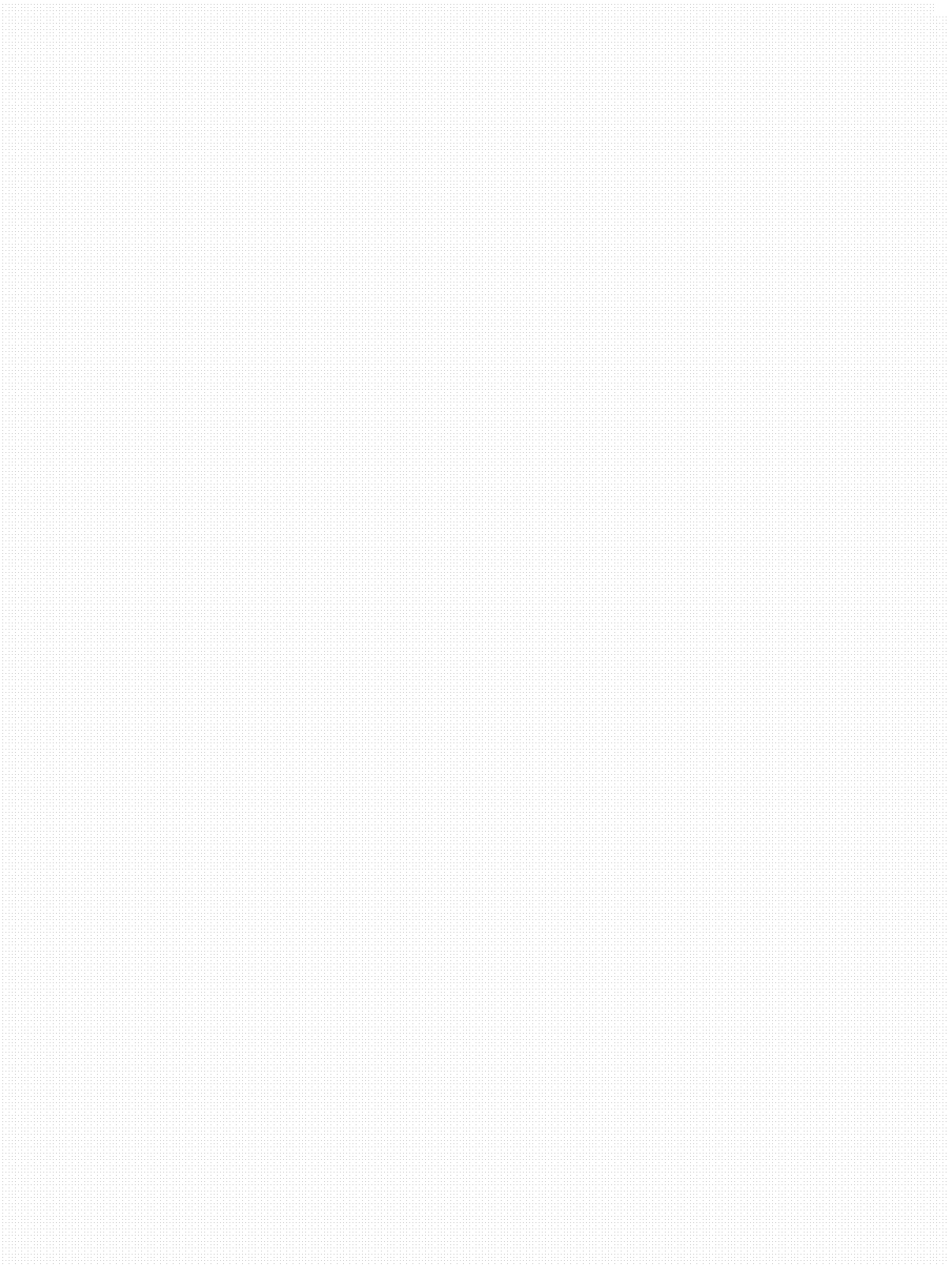
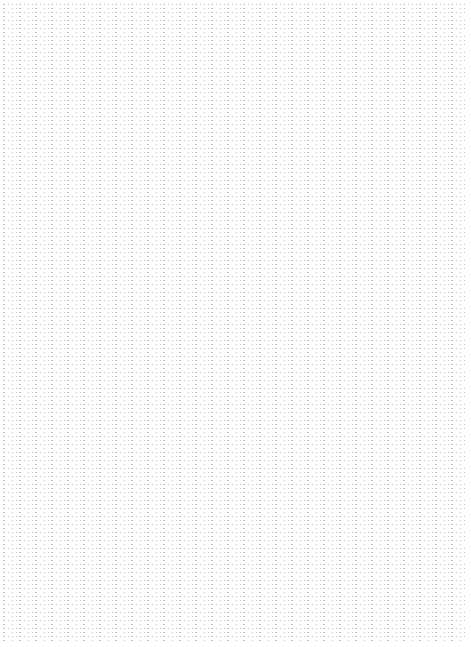
Response

Requests

2

Response

对象包含爬虫返回的内容



## Response对象

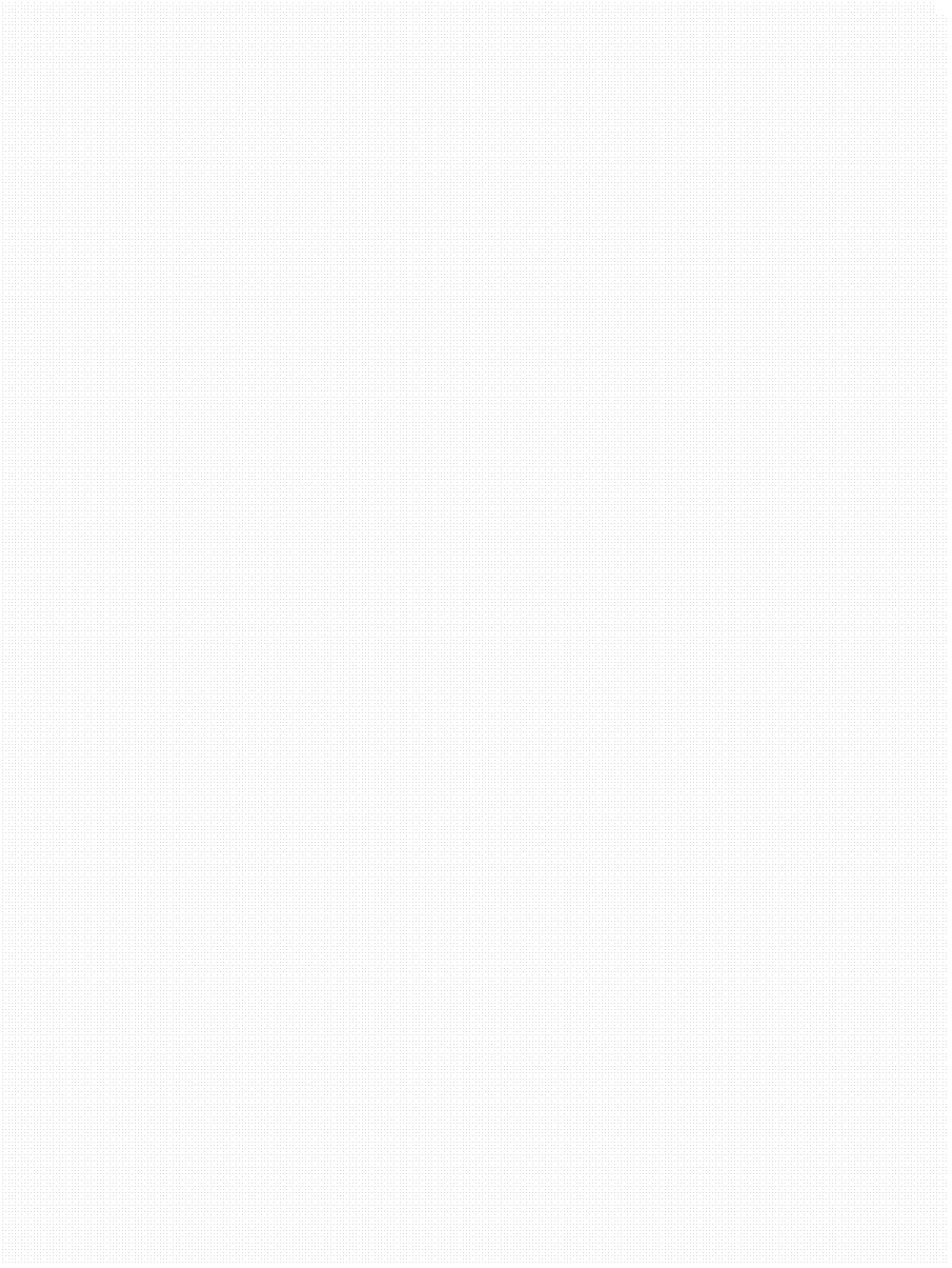
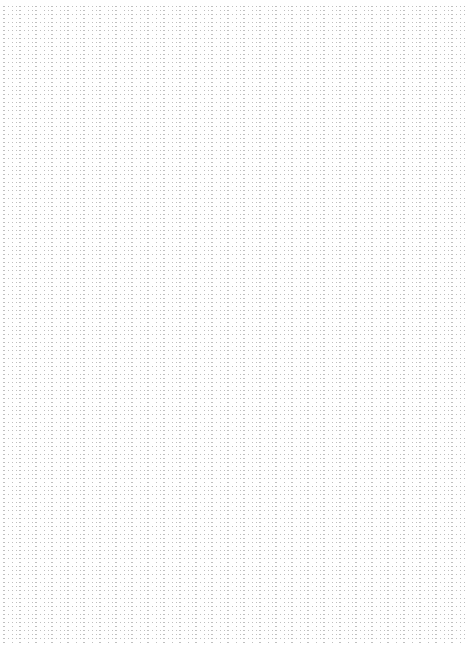
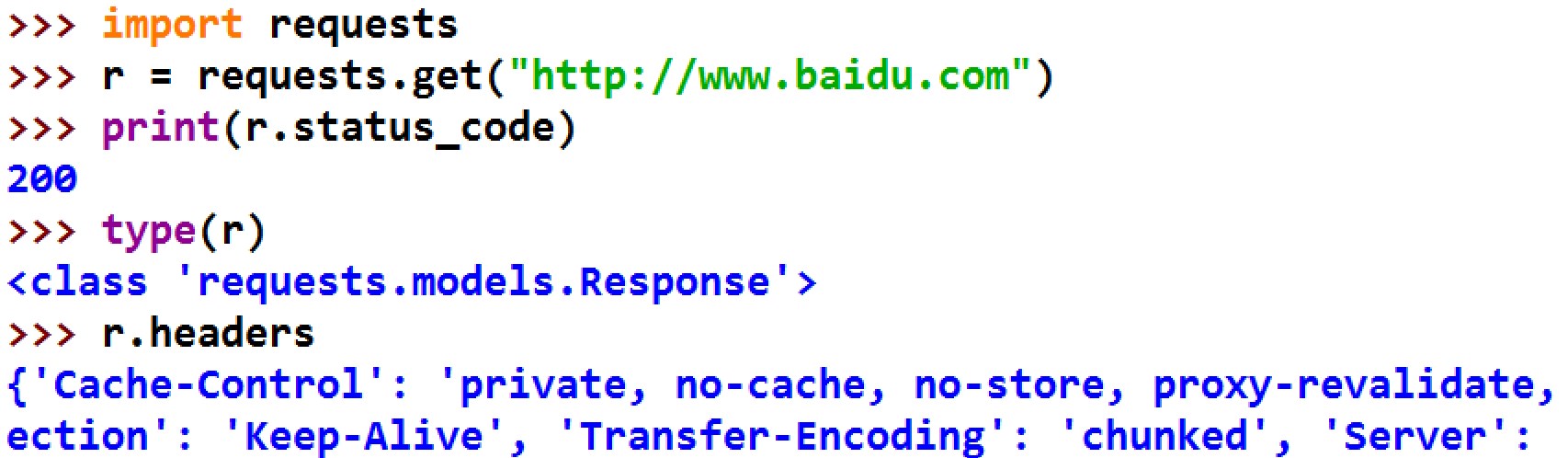
Response

Response

对象包含服务器返回的所有信息，也包含请求的

Request

信息



Response

(1)

属性

说明

r.status\_code

HTTP

请求的返回状态，

200

表示连接成功，

404

表示失败

r.text

HTTP

响应内容的字符串形式，即，

url

对应的页面内容

r.encoding

从

HTTP

header

中猜测的响应内容编码方式

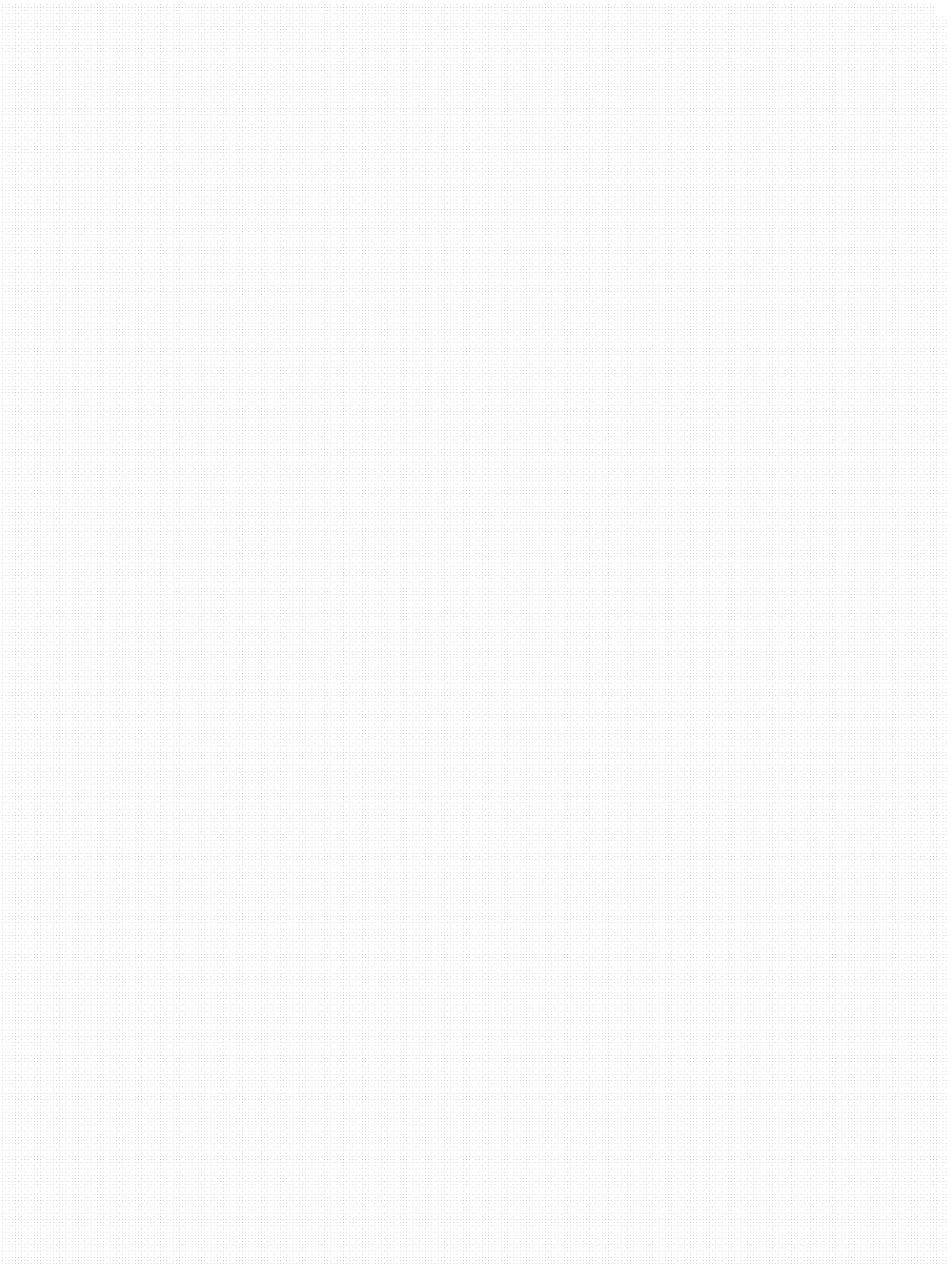
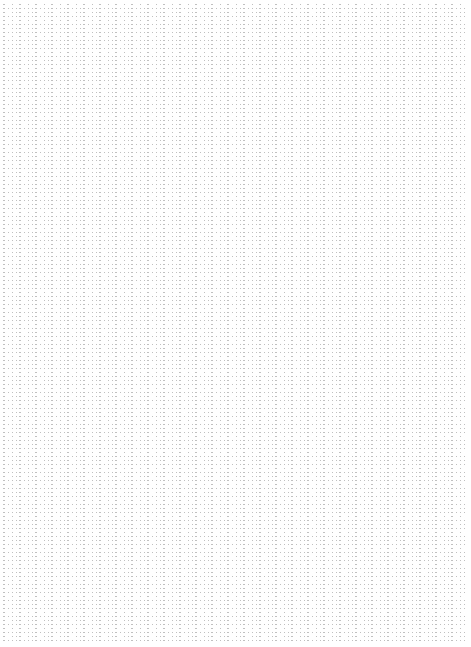
r.apparent\_encoding

从内容中分析出的响应内容编码方式（备选编码方式）

r.content

HTTP

响应内容的二进制形式



Response

r.status\_code

404

或其他

200

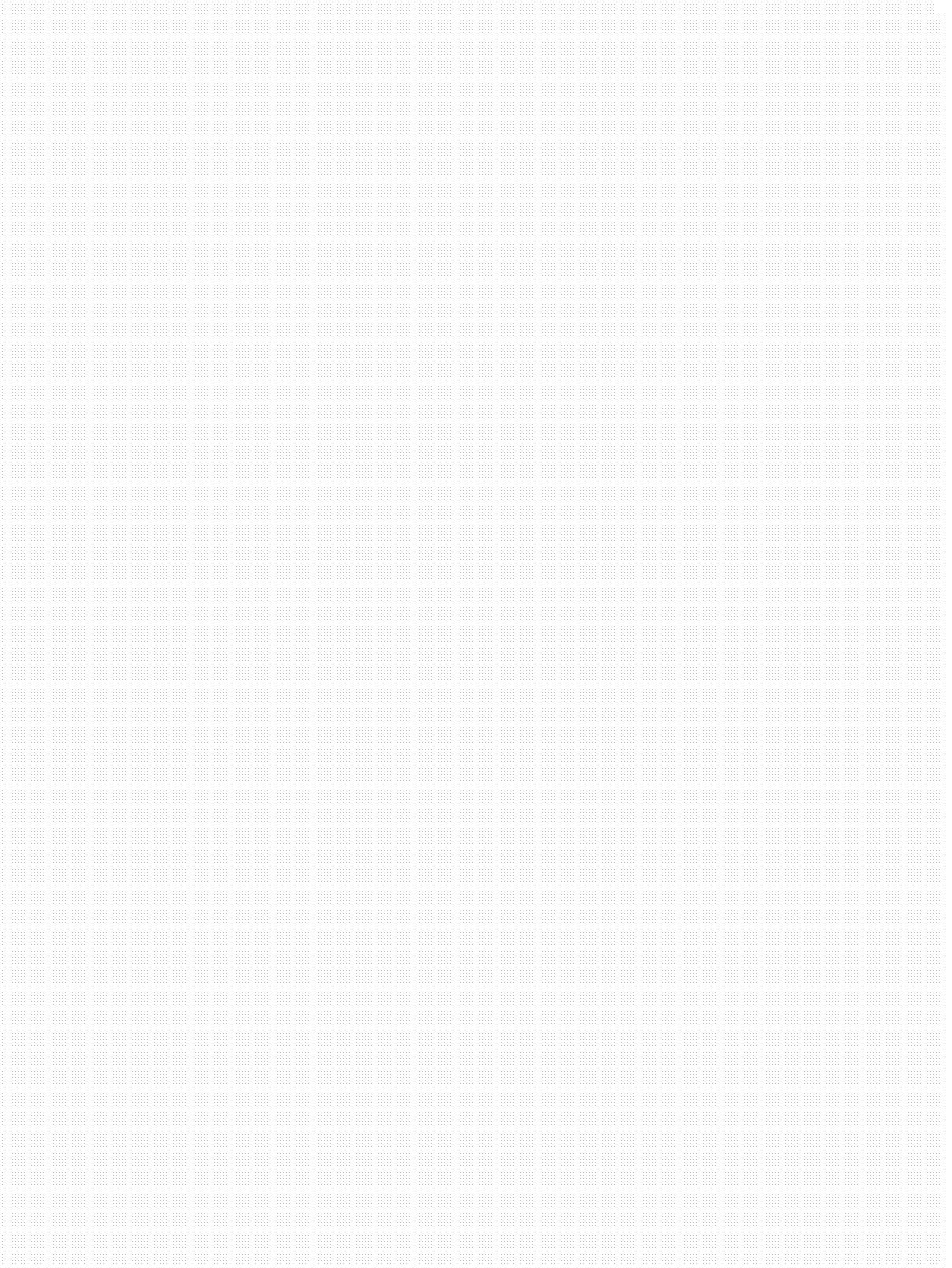
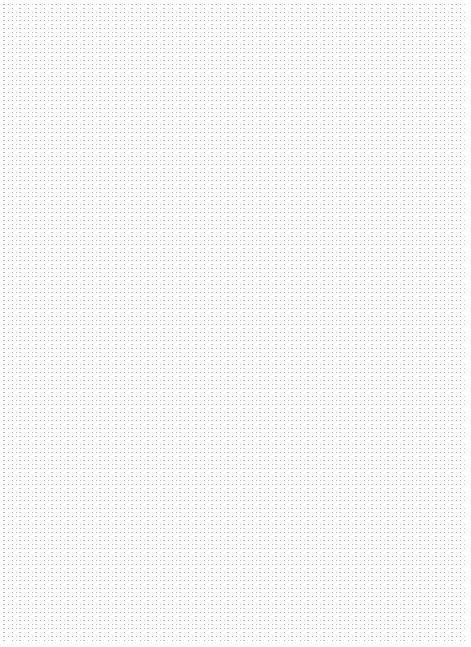
r.textr.encoding

r.apparent\_encoding

r.content

某些原因出错

将产生异常



Response

r.encoding

从

HTTP

header

中猜测的响应内容编码方式

r.apparent\_encoding

从内容中分析出的响应内容编码方式（备选编码方式）

r.encoding

：如果

header

中不存在

charset

，则认为编码为

ISO

‐

8859

‐

1

r.text

根据

r.encoding

显示网页内容

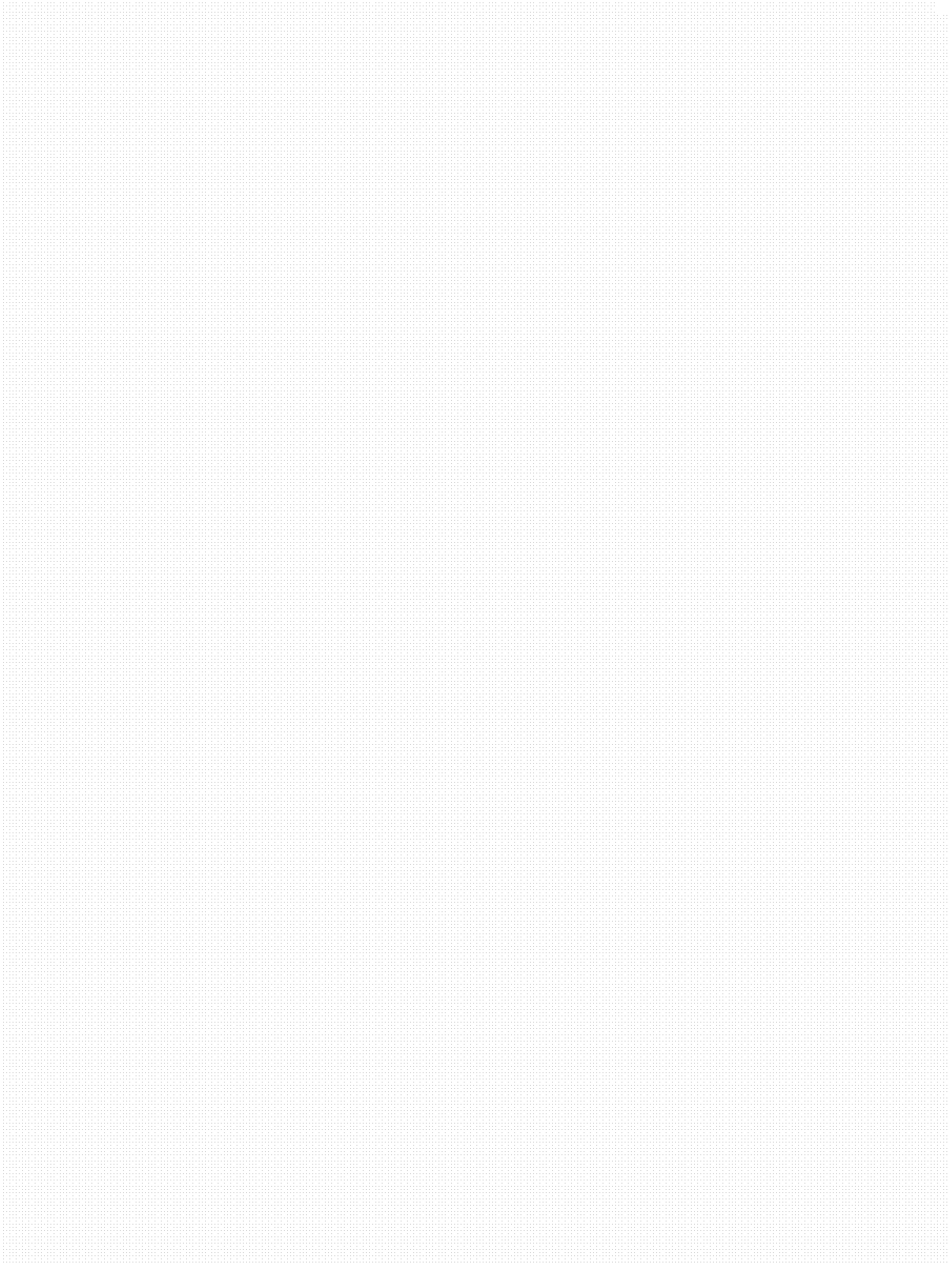
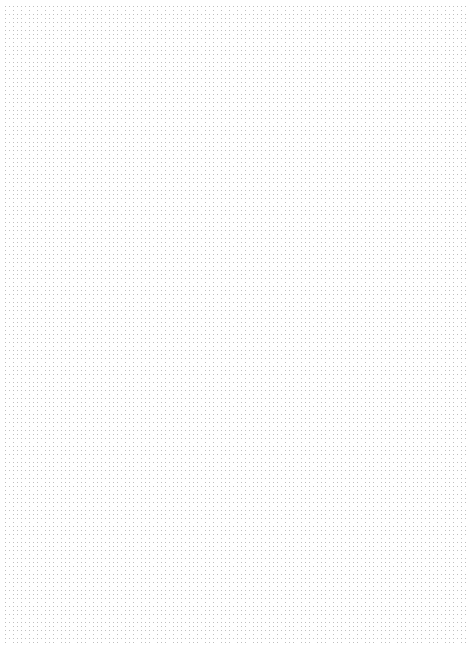
r.apparent\_encoding

：根据网页内容分析出的编码方式

可以看作是

r.encoding

的备选



## Requests库异常

Requests

异常

说明

requests.ConnectionError

网络连接错误异常，如

DNSS

查询失败、拒绝连接等

requests.HTTPError

HTTP

错误异常

requests.URLRequired

URL

缺失异常

requests.TooManyRedirects

超过最大重定向次数，产生重定向异常

requests.ConnectTimeout

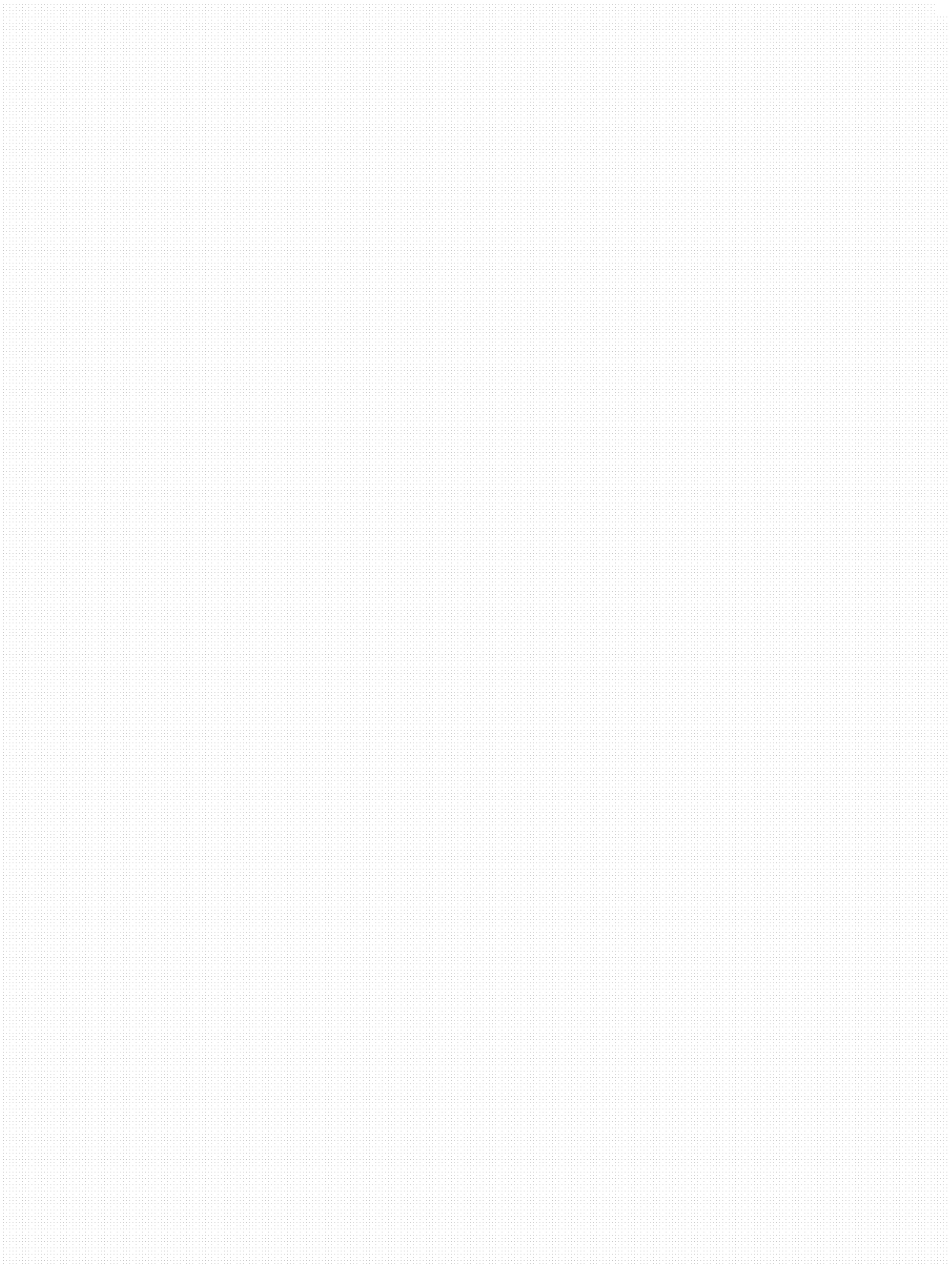
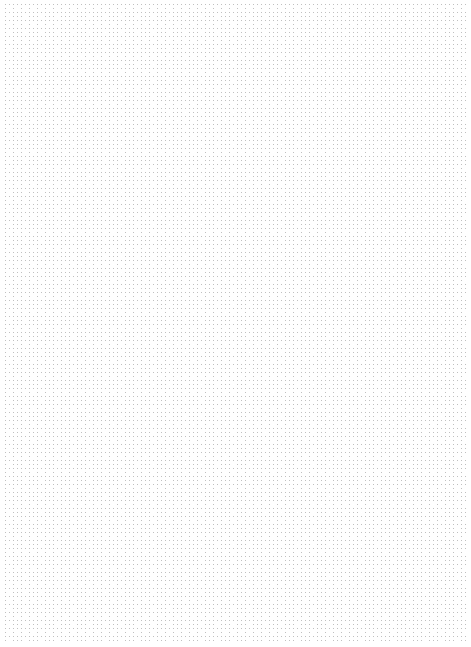
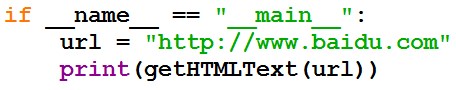
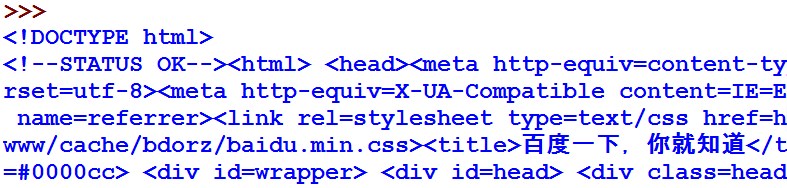
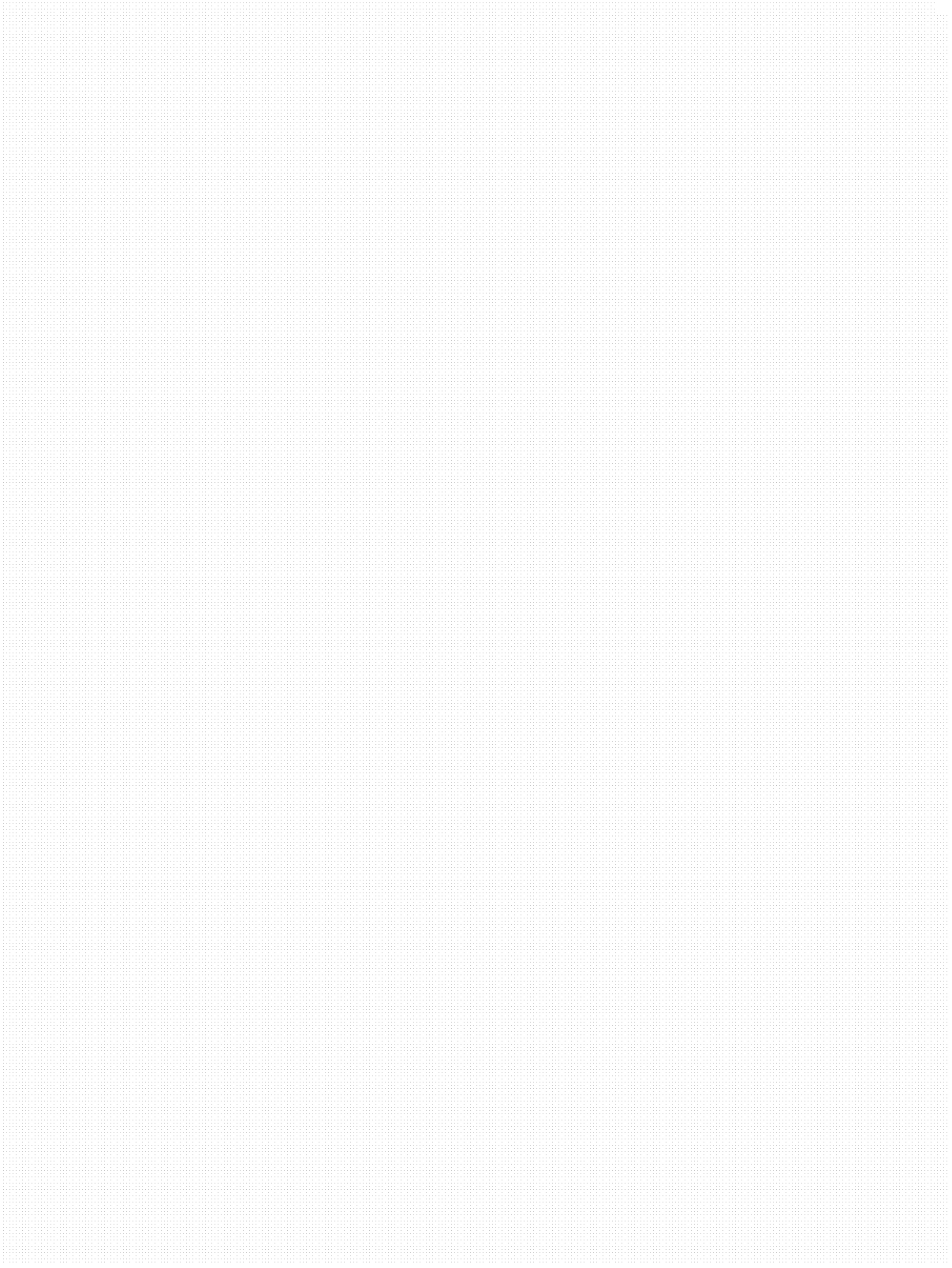
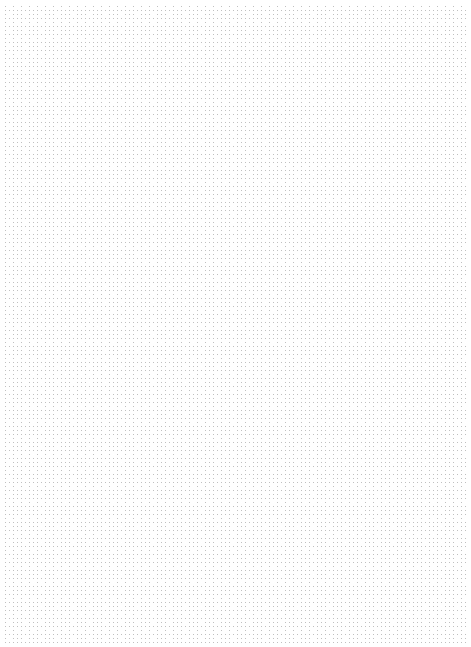
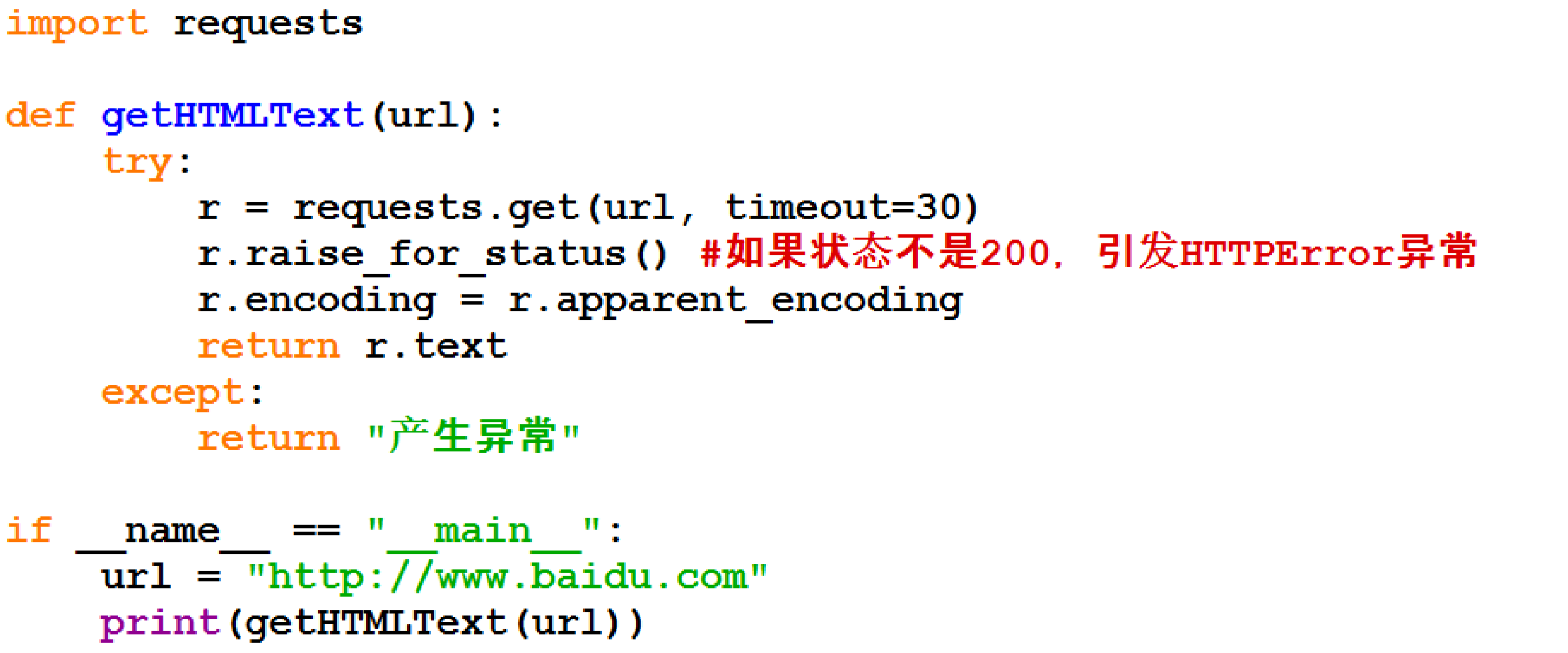
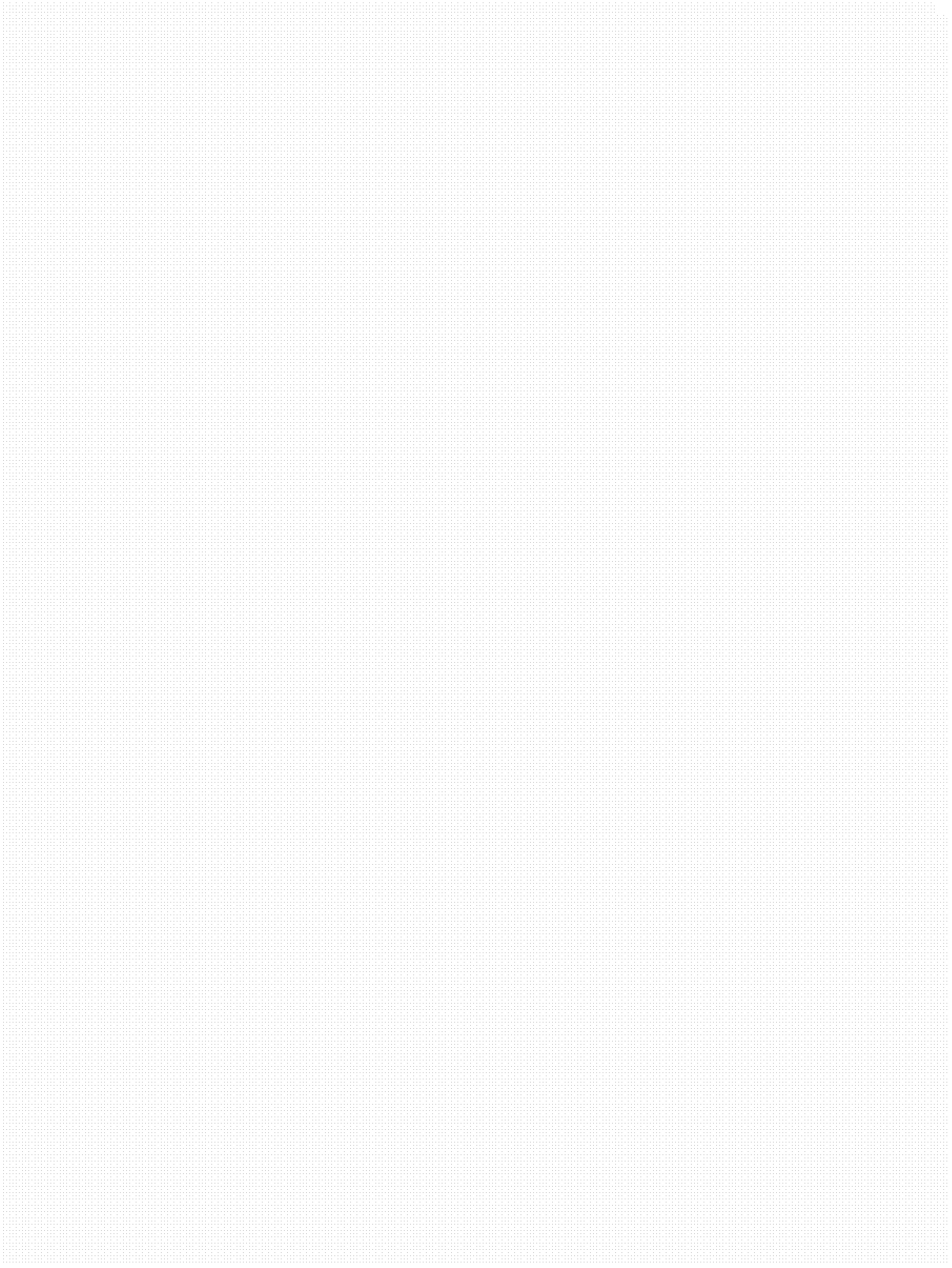
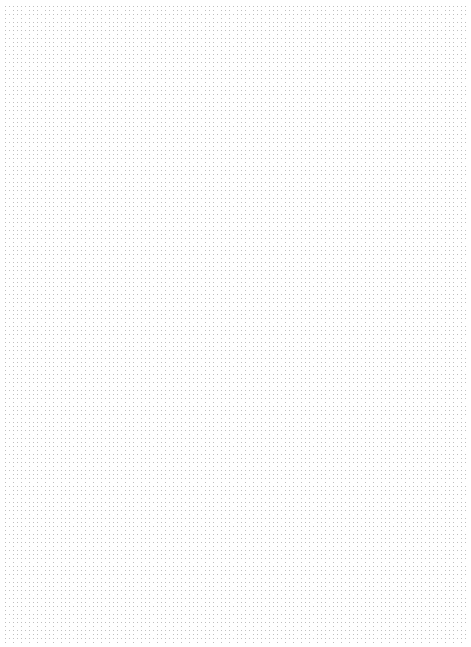
连接远程服务器超时异常

requests.Timeout

请求

URL

超时，产生超时异常



## HTTP协议

HTTP

**HTTP**

，

Hypertext

Transfer

Protocol

，超文本传输协议

HTTP

是一个基于“请求与响应”模式的、无状态的应用层协议

HTTP

协议采用

URL

作为定位网络资源的标识，

URL

格式如下：

http://host[:port][path]

host

:

合法的

Internet

主机域名或

IP

地址

port

:

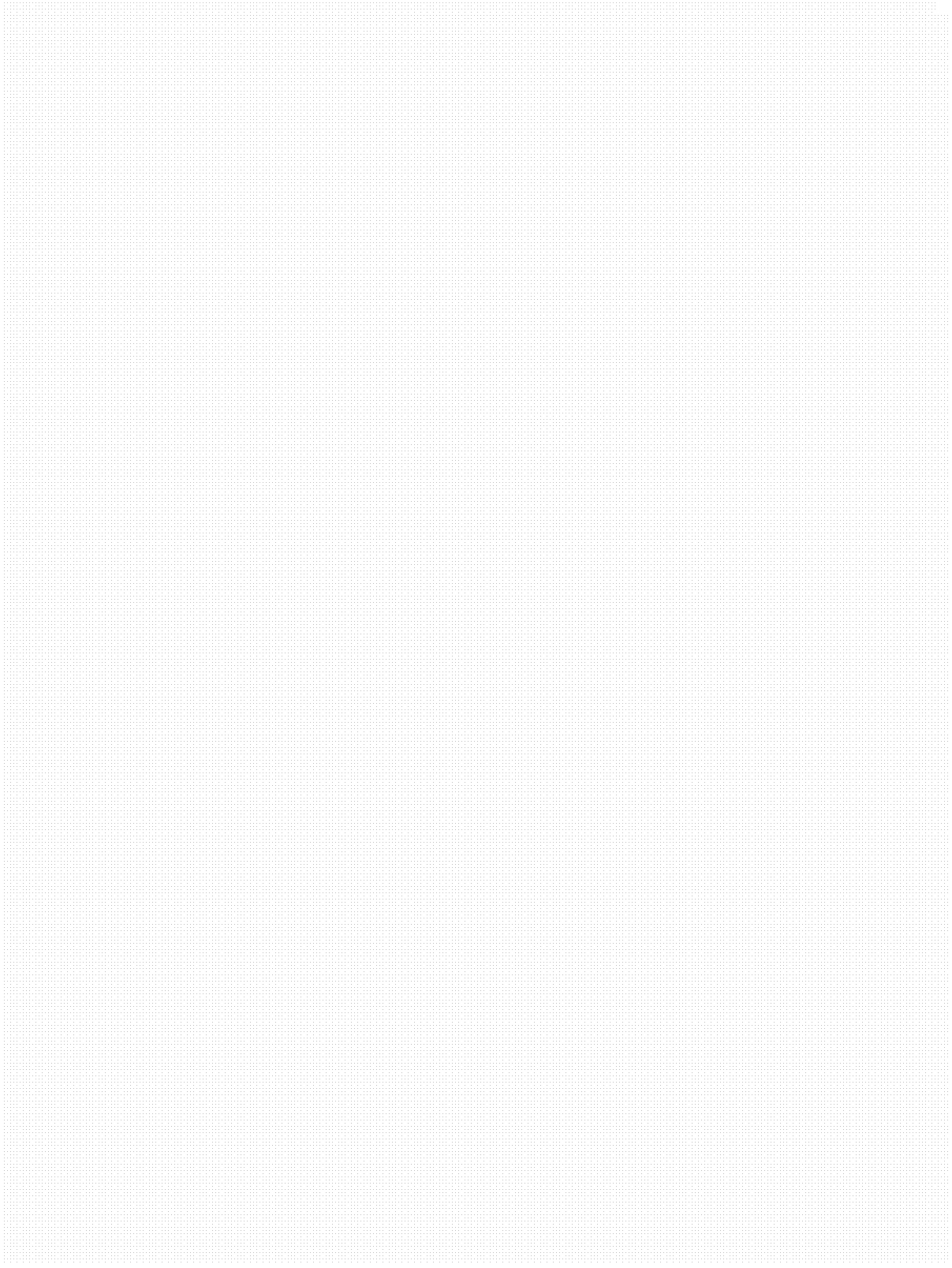
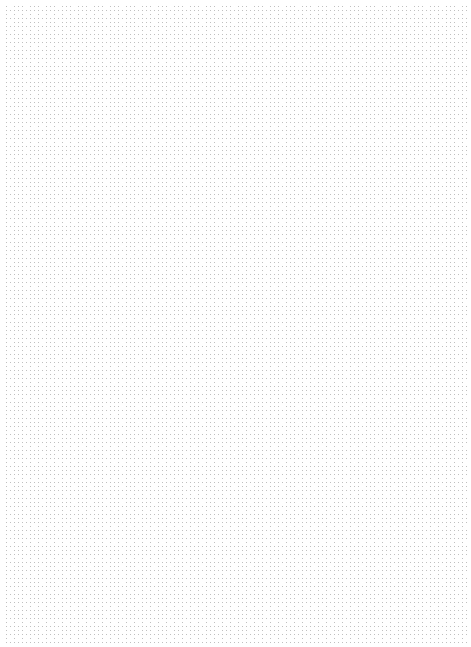
端口号，缺省端口为

80

path

:

请求资源的路径



HTTP

**HTTP**

，

Hypertext

Transfer

Protocol

，超文本传输协议

HTTP

URL

实例：

http://www.bit.edu.cn

http://220.181.111.188/duty

HTTP

URL

的理解：

URL

是通过

HTTP

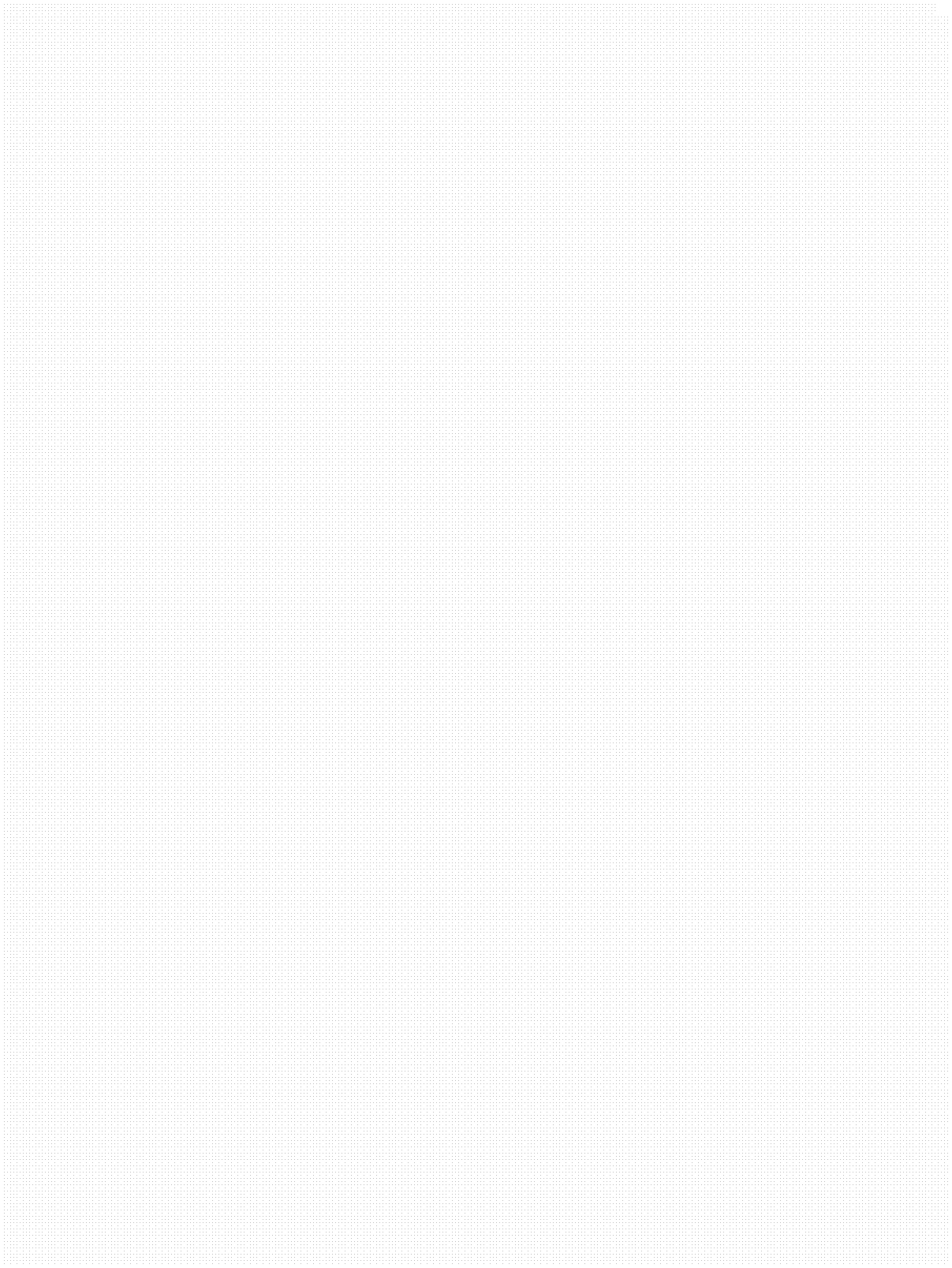
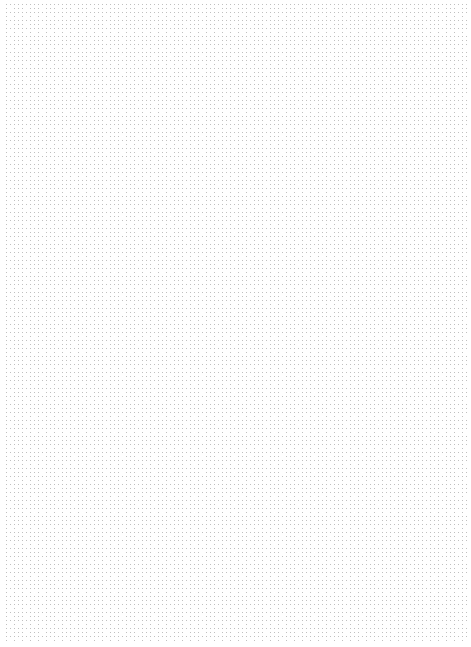
协议存取资源的

Internet

路径，一个

URL

对应一个数据资源



HTTP

方法

说明

GET

请求获取

URL

位置的资源

HEAD

请求获取

URL

位置资源的响应消息报告，即获得该资源的头部信息

POST

请求向

URL

位置的资源后附加新的数据

PUT

请求向

URL

位置存储一个资源，覆盖原

URL

位置的资源

PATCH

请求局部更新

URL

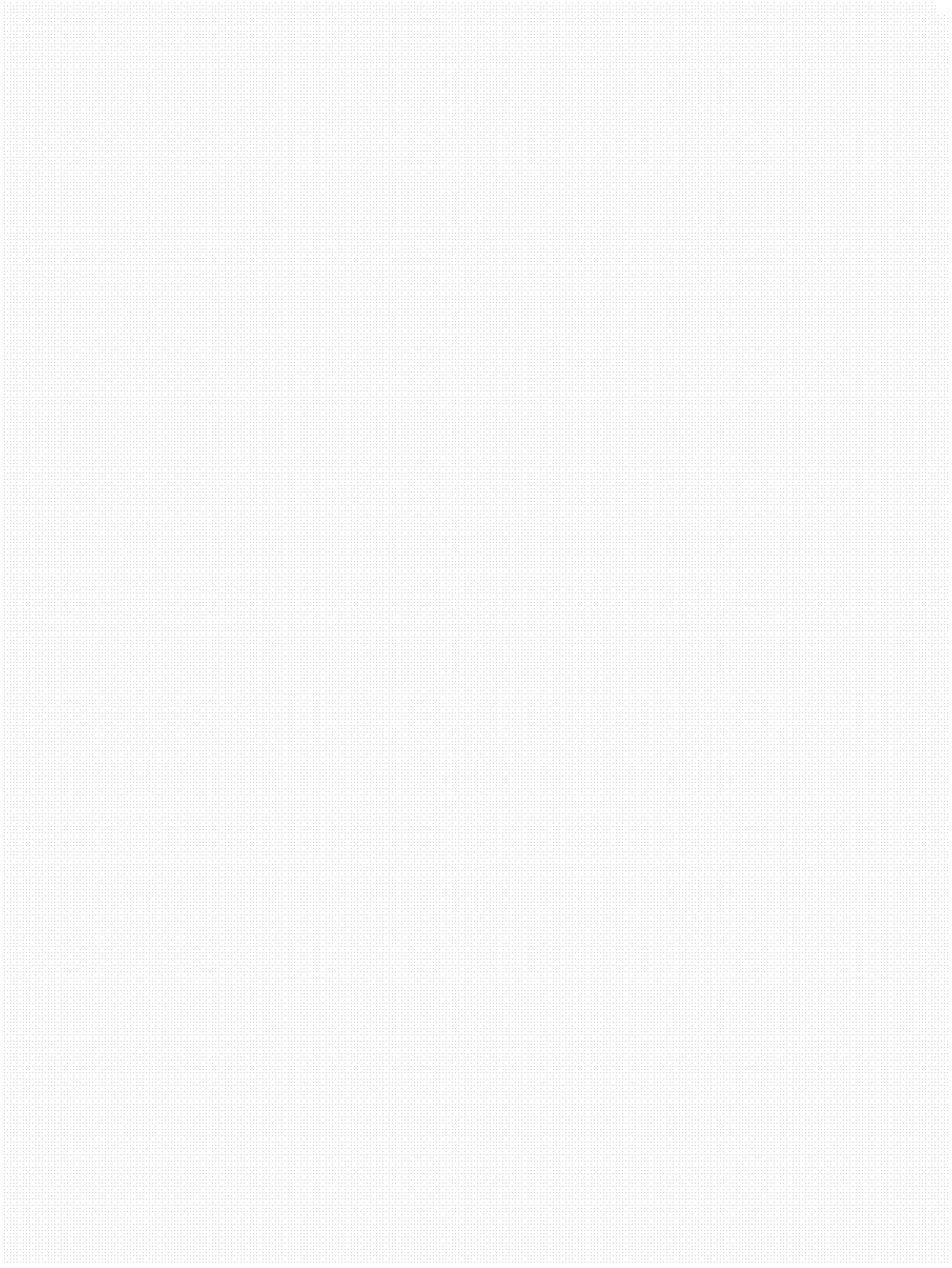
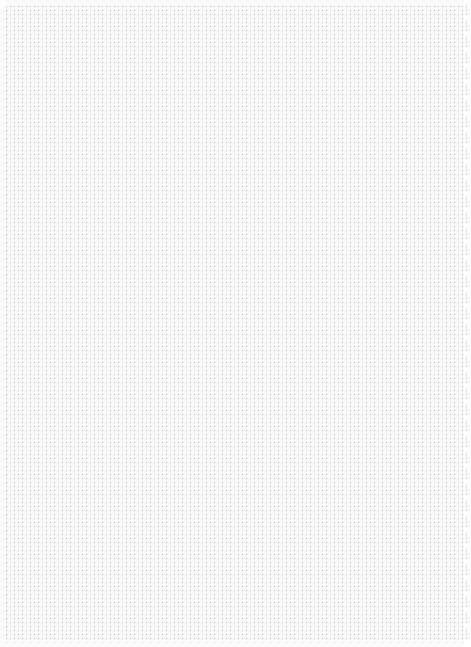
位置的资源，即改变该处资源的部分内容

DELETE

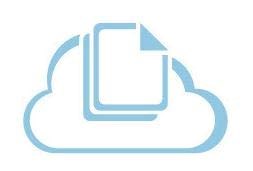
请求删除

URL

位置存储的资源



HTTP



GET

HEAD

PUT

POST

PATCH

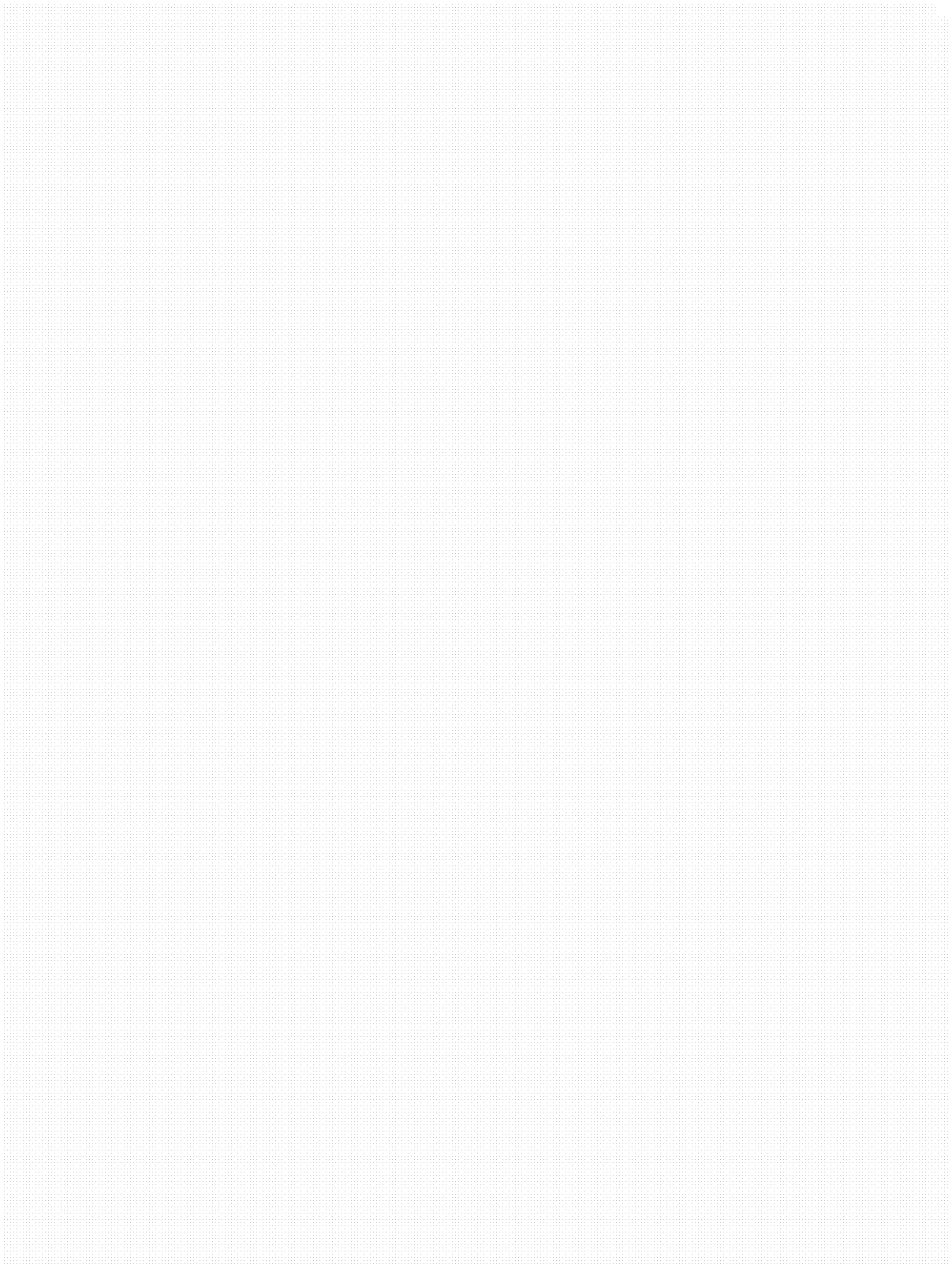
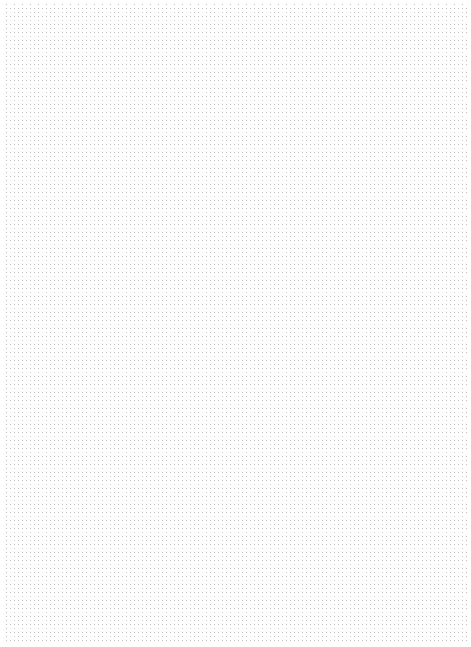
DELETE

URL

通过

URL

和命令管理资源，操作独立无状态，网络通道及服务器成为了黑盒子



PATCH

PUT

假设

URL

位置有一组数据

UserInfo

，包括

UserID

、

UserName

等

20

个字段

需求：用户修改了

UserName

，其他不变

•

采用

PATCH

，仅向

URL

提交

UserName

的局部更新请求

•

采用

PUT

，必须将所有

20

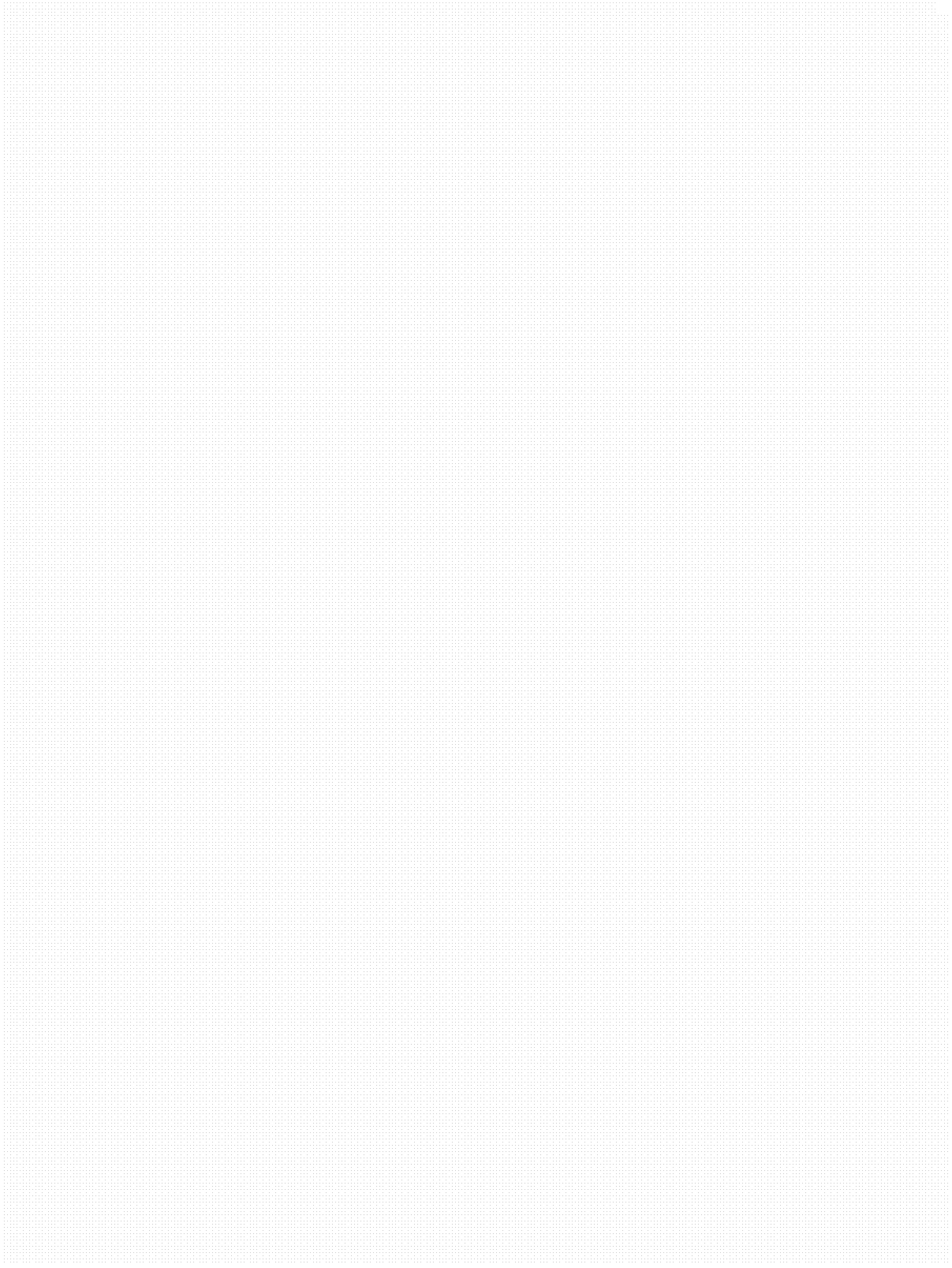
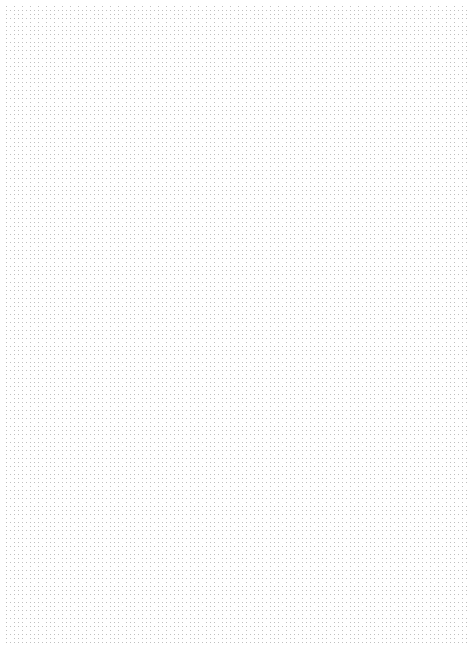
个字段一并提交到

URL

，未提交字段被删除

PATCH

的最主要好处：节省网络带宽



HTTP

Requests

HTTP协议方法

Requests库方法

功能一致性

GET

requests.get()

一致

HEAD

requests.head()

一致

POST

requests.post()

一致

PUT

requests.put()

一致

PATCH

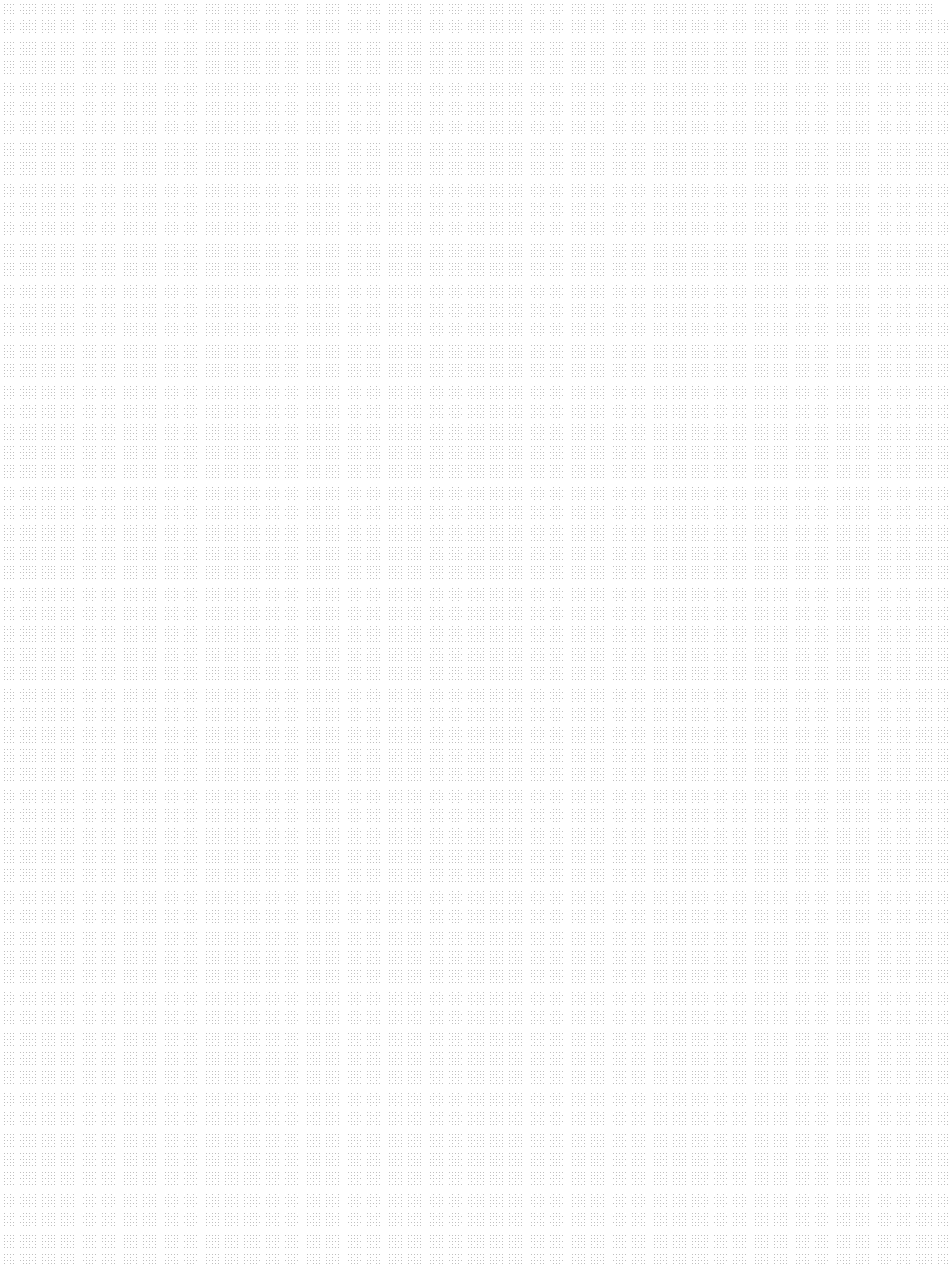
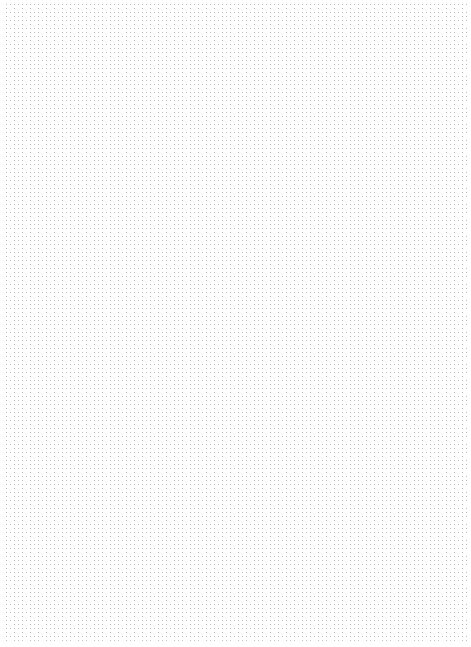
requests.patch()

一致

DELETE

requests.delete()

一致



## 七个方法详解

|  |
| --- |
| >>> r = requests.head**(**'http://httpbin.org/get'**)** >>> r.headers  {'Content‐Length': '238', 'Access‐Control‐Allow‐Origin': '\*', 'AccessControl‐Allow‐Credentials': 'true', 'Content‐Type':  'application/json', 'Server': 'nginx', 'Connection': 'keep‐alive',  'Date': 'Sat, 18 Feb 2017 12:07:44 GMT'} >>> r.text  '' |

Requests

post()

>>>

payload

=

{

'key1'

:

'value1'

,

'key2'

:

'value2'

}

>>>

r

=

requests

.

post

**(**

'http://httpbin.org/post'

, data

=

payload

**)**

>>>

print(r.text)

{

...

"form":

{

"key2":

"value2",

"key1":

"value1"

}

,

}

向

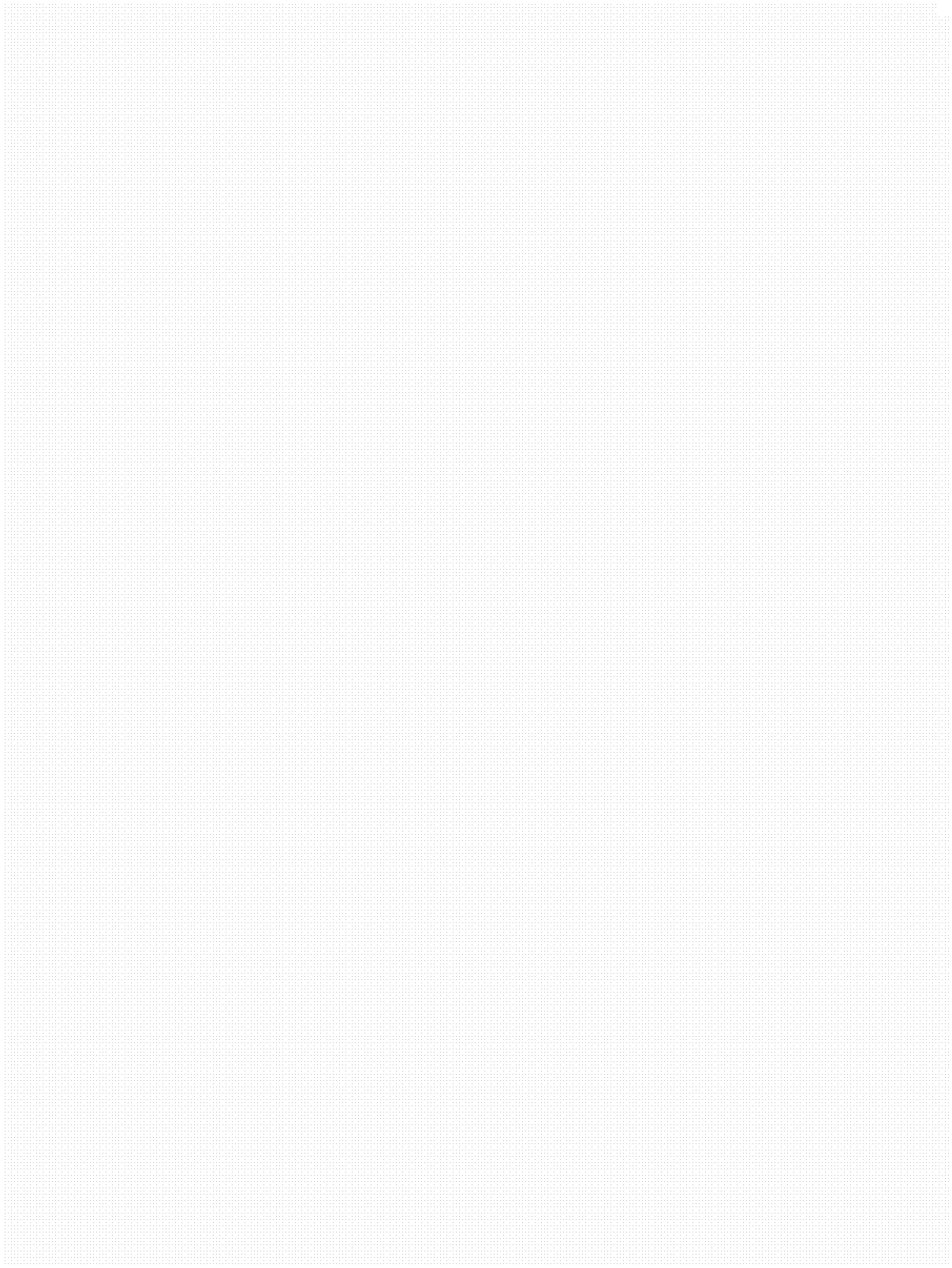
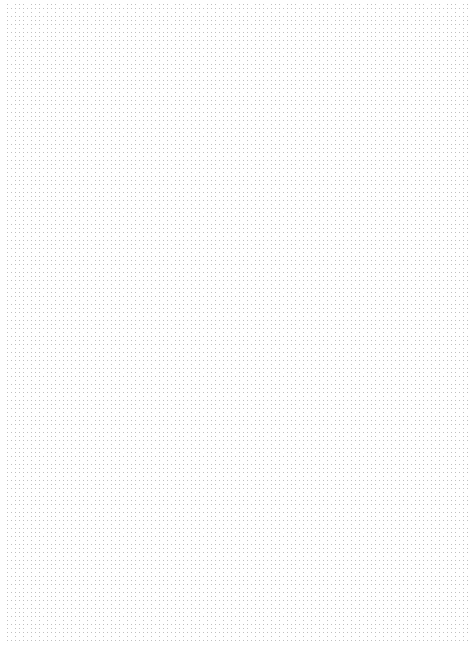
URL POST

一个字典

自动编码为

form

（表单）



Requests

post()

>>>

r

=

requests

.

post

**(**

'http://httpbin.org/post'

, data

=

'ABC'

**)**

>>>

print(r.text)

{

...

"data":

"ABC"

"form":

,

{}

}

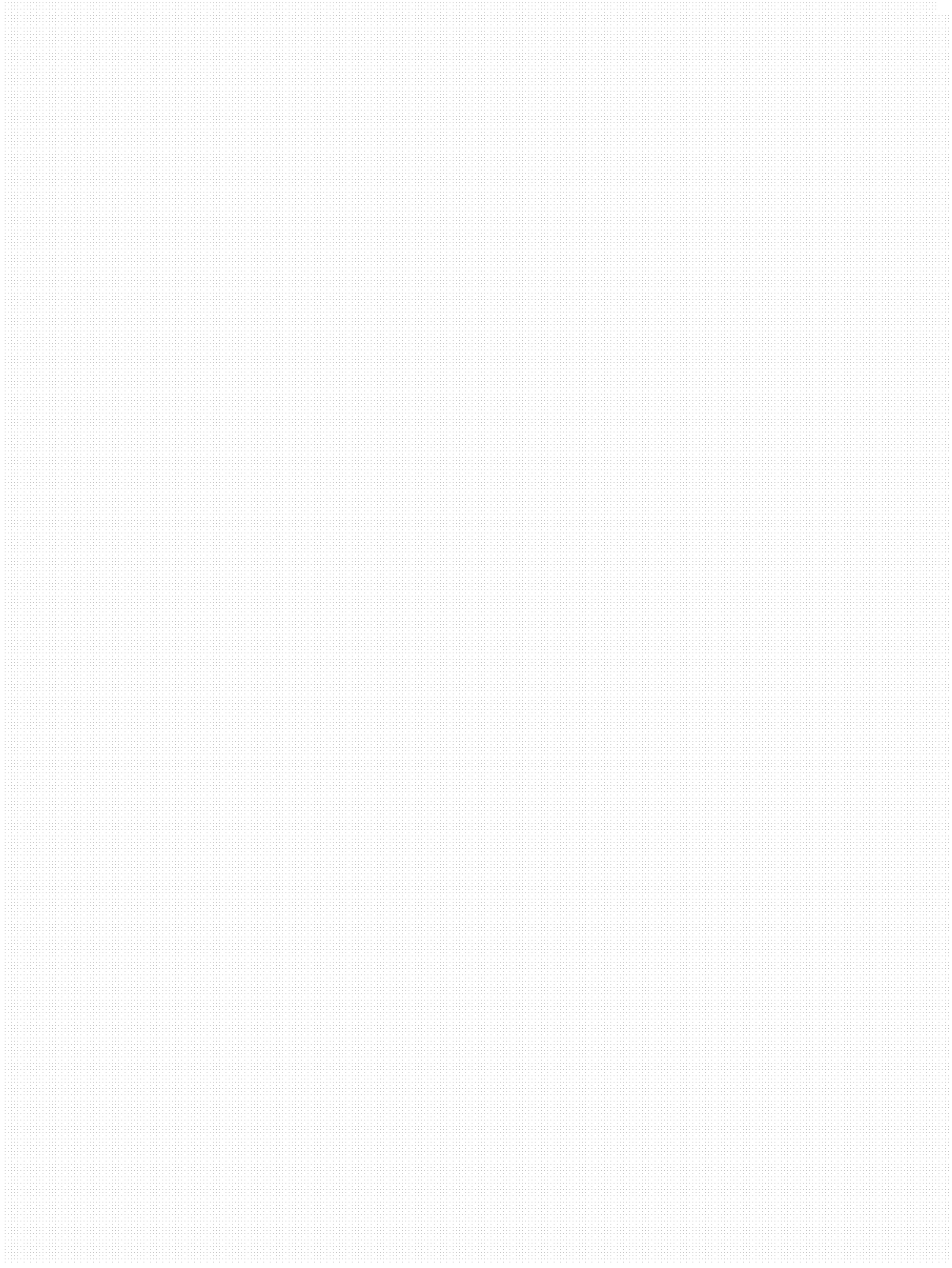
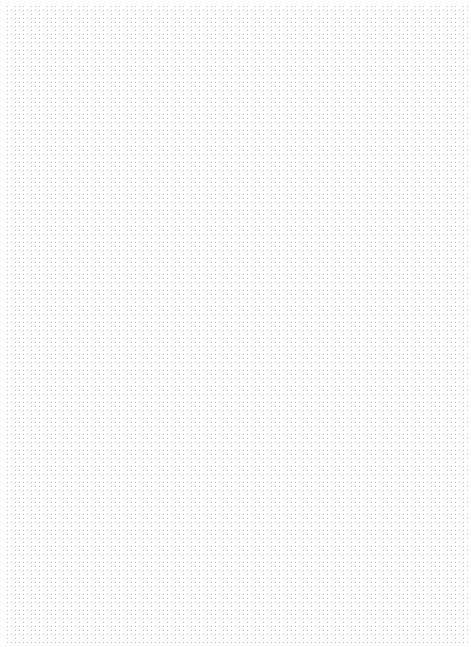
向

URL POST

一个字符串

自动编码为

data



Requests

put()

>>>

payload

=

{

'key1'

:

'value1'

,

'key2'

:

'value2'

}

>>>

r

=

requests

.

put

**(**

'http://httpbin.org/put'

, data

=

payload

**)**

>>>

print(r.text)

{

...

"form":

{

"key2":

"value2",

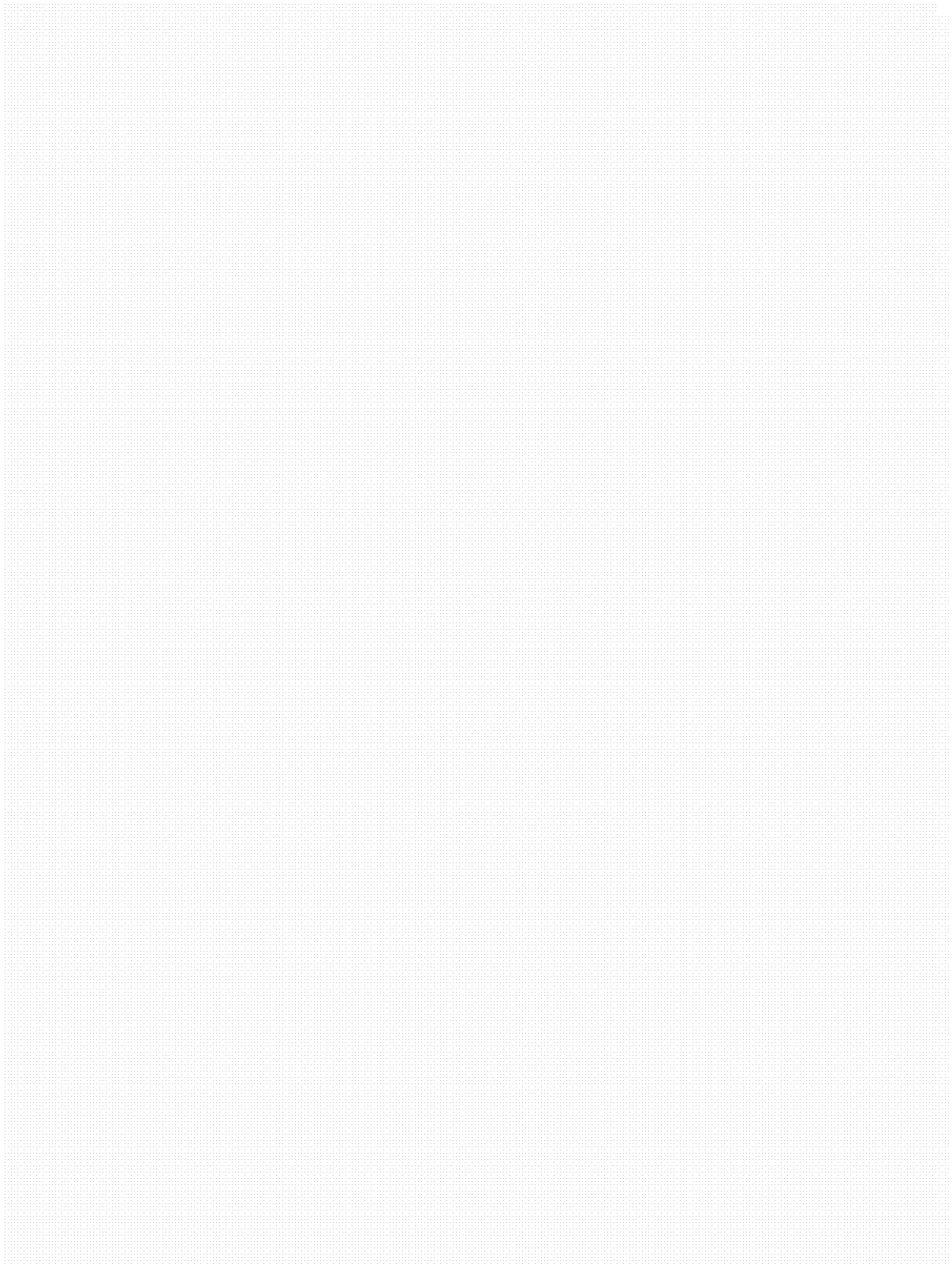
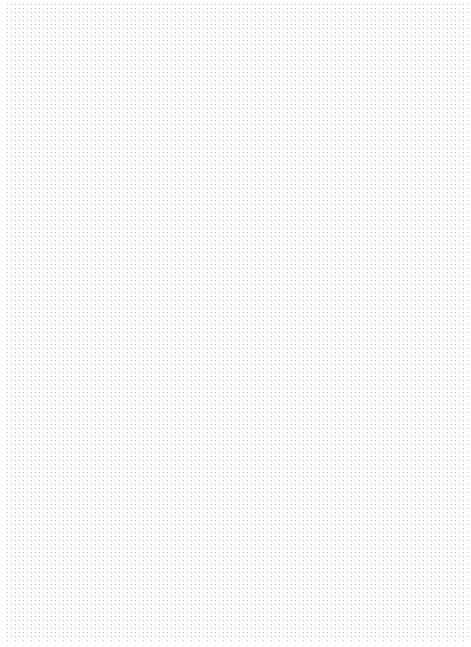
"key1":

"value1"

}

,

}



**∙ url** requests.request(**method**, **url**, **\*\*kwargs**)

requests.request(

**method**

,

**url**

,

**\*\*kwargs**

)

**∙**

**method**

:

请求方式，对应

get/put/post

等

7

种

:

拟获取页面的

url

链接

**∙**

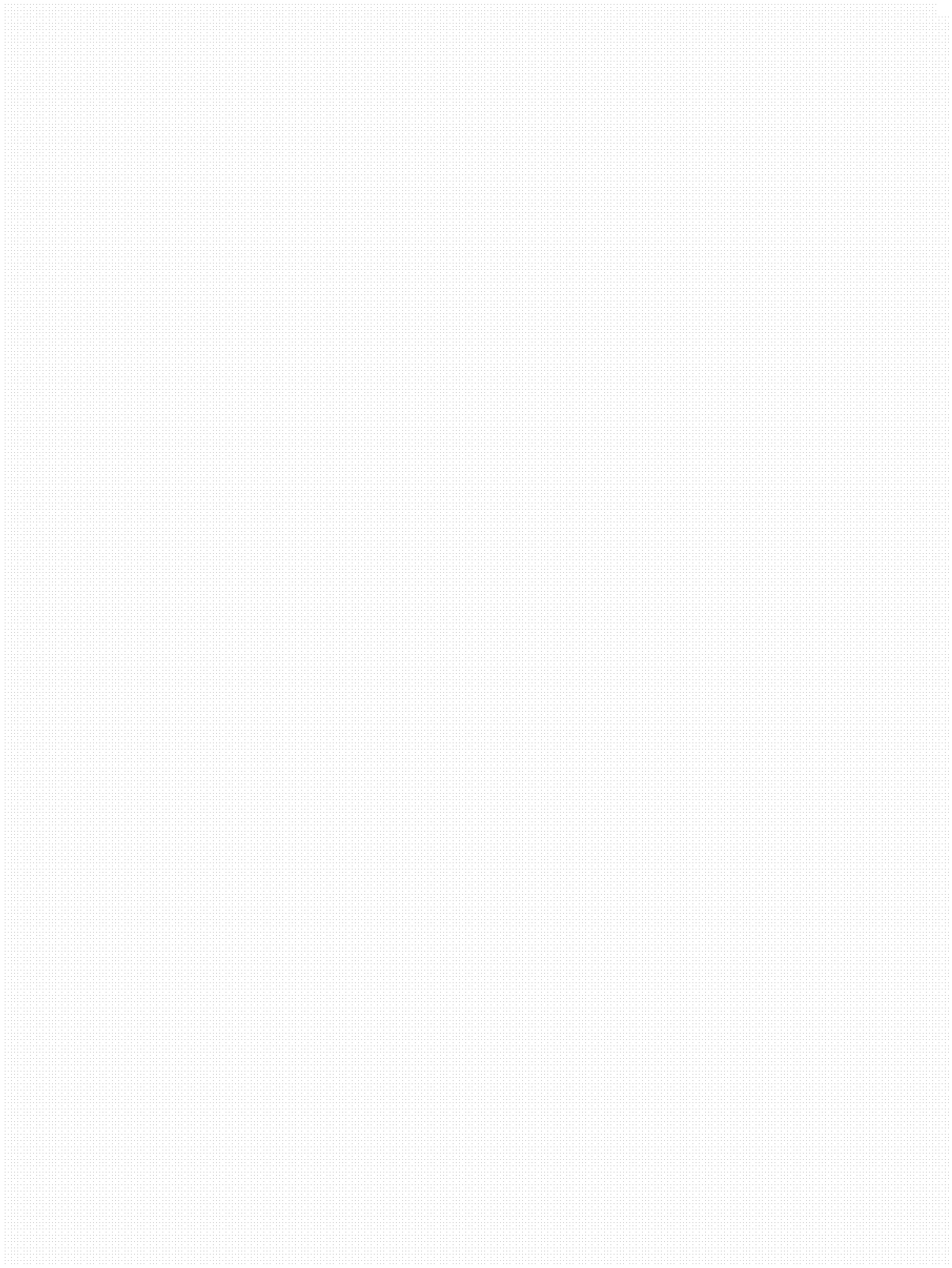
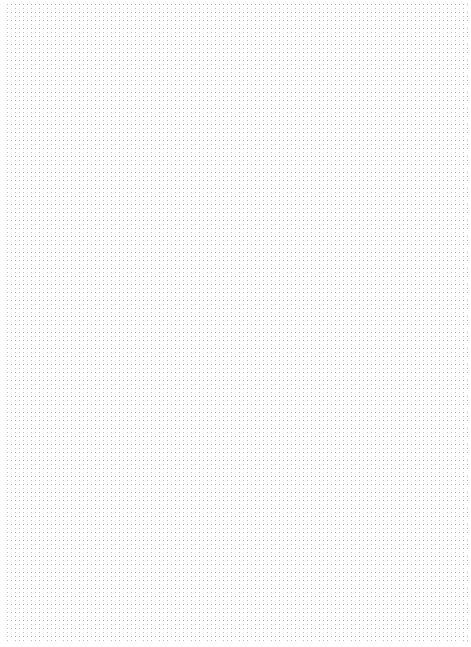
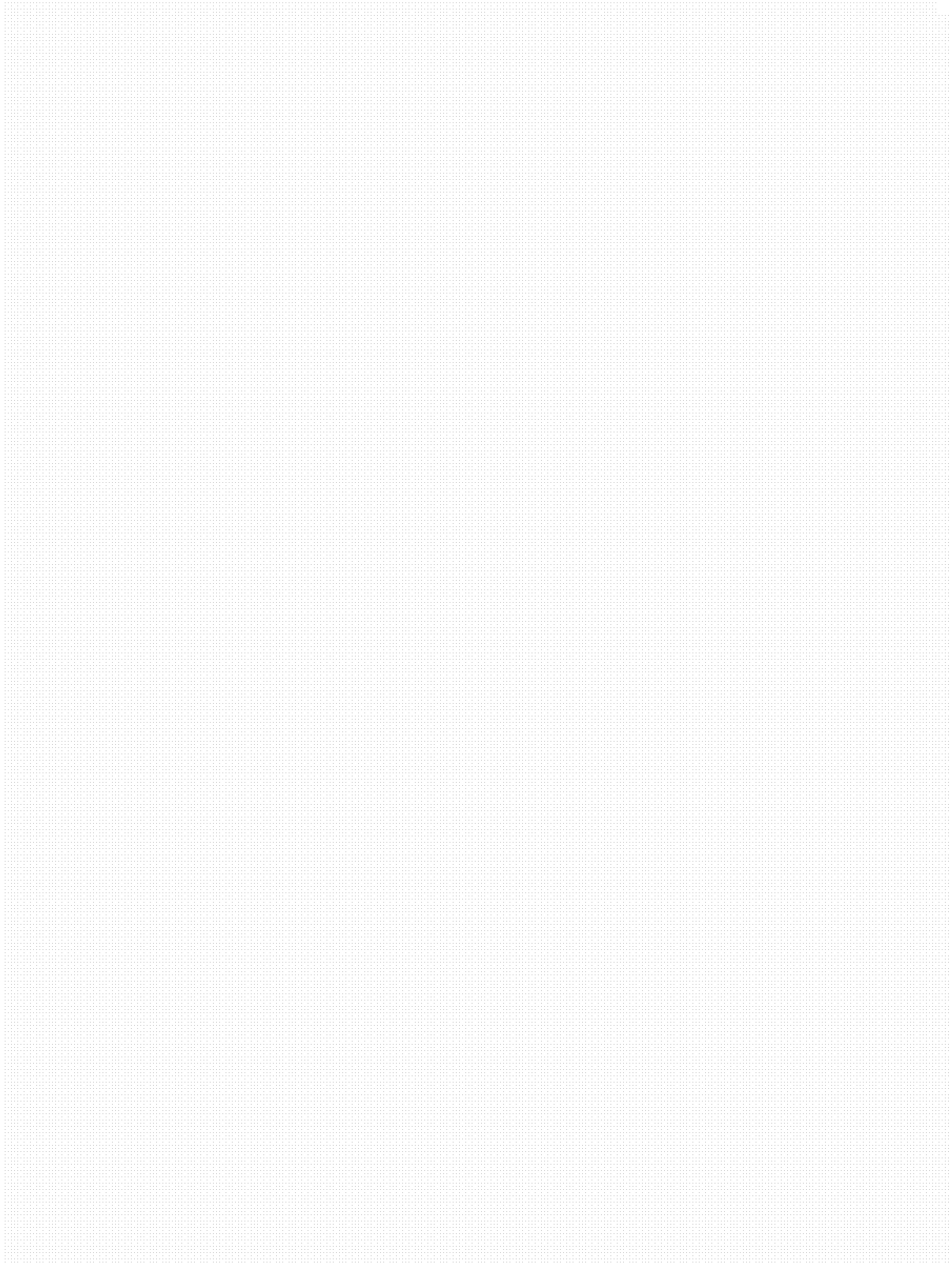
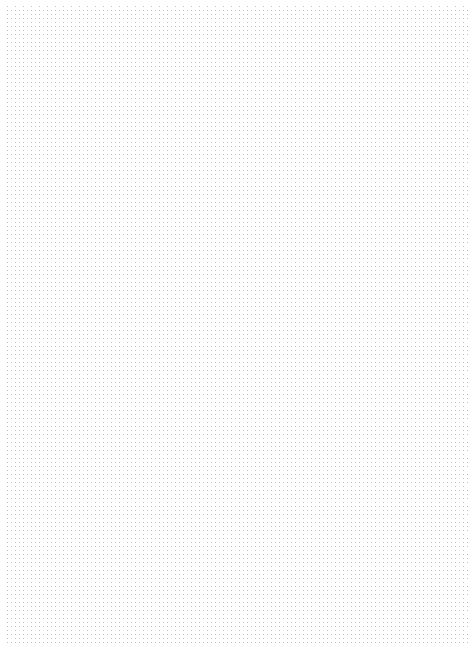
**\*\*kwargs**

:

控制访问的参数，共

13

个



**∙ method** : 请求方式

r = requests.request(**'GET'**, **url**, **\*\*kwargs**) r = requests.request(**'HEAD'**, **url**, **\*\*kwargs**) r = requests.request(**'POST'**, **url**, **\*\*kwargs**) r = requests.request(**'PUT'**, **url**, **\*\*kwargs**) r = requests.request(**'PATCH'**, **url**, **\*\*kwargs**) r = requests.request(**'delete'**, **url**, **\*\*kwargs**) r = requests.request(**'OPTIONS'**, **url**, **\*\*kwargs**)

requests.request(

**method**

,

**url**

,

**\*\*kwargs**

)

**∙**

**\*\*kwargs**

:

控制访问的参数，均为可选项

params:

字典或字节序列，作为参数增加到

url

中

>>>

kv

=

**{**

'key1'

**:**

'value1'

**,**

'key2'

**:**

'value2'

**}**

>>>

r

=

requests

.

request

**(**

'GET'

**,**

'http://python123.io/ws'

**,**

params

=

kv

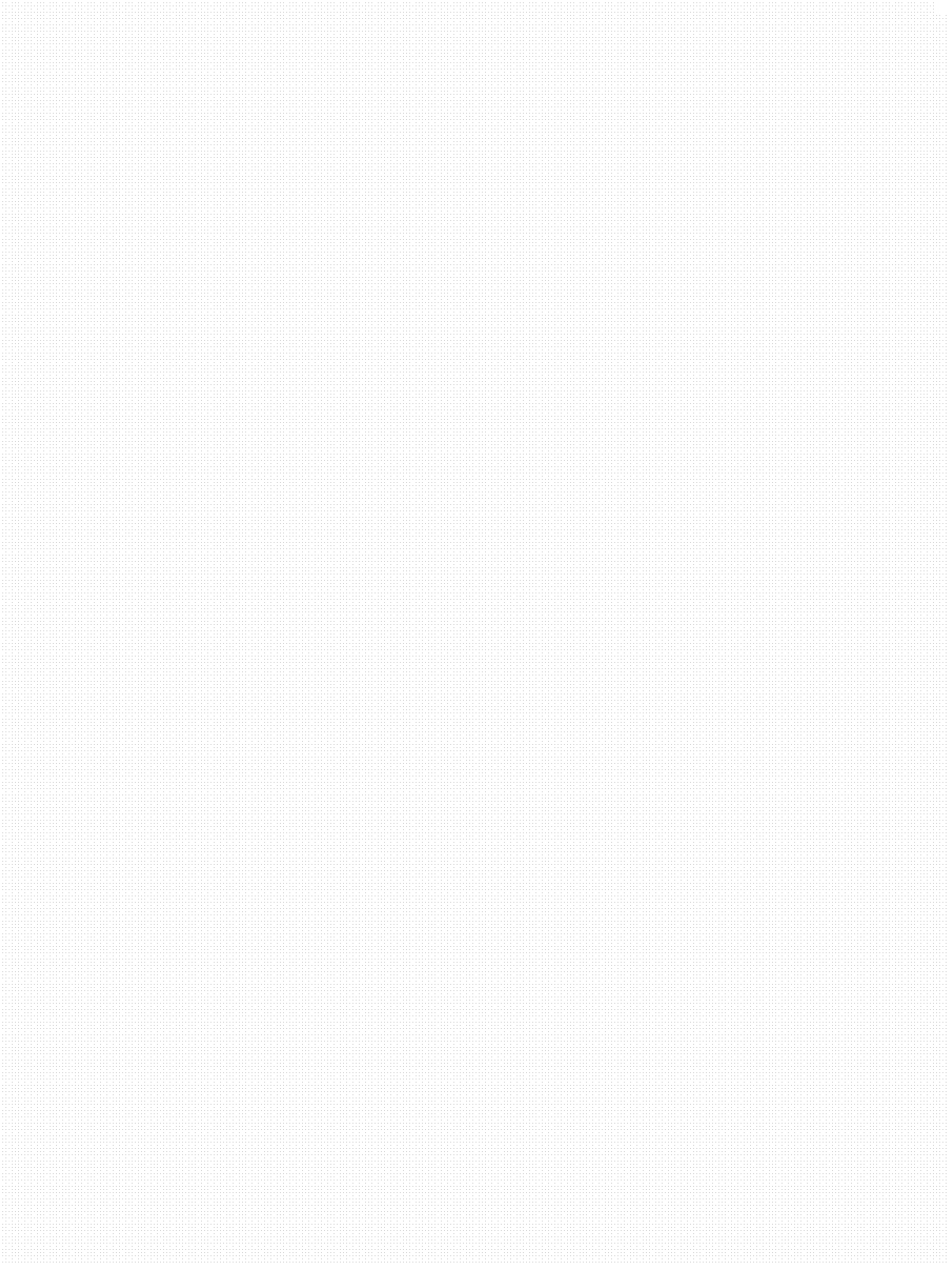
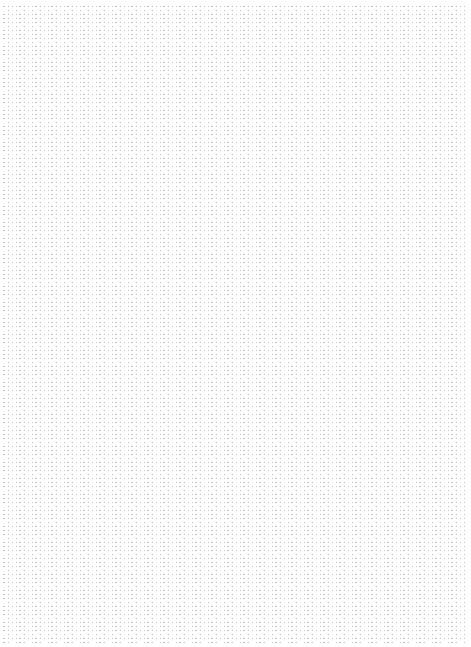
**)**

>>>

print(r.url)

http://python123.io/ws?ke

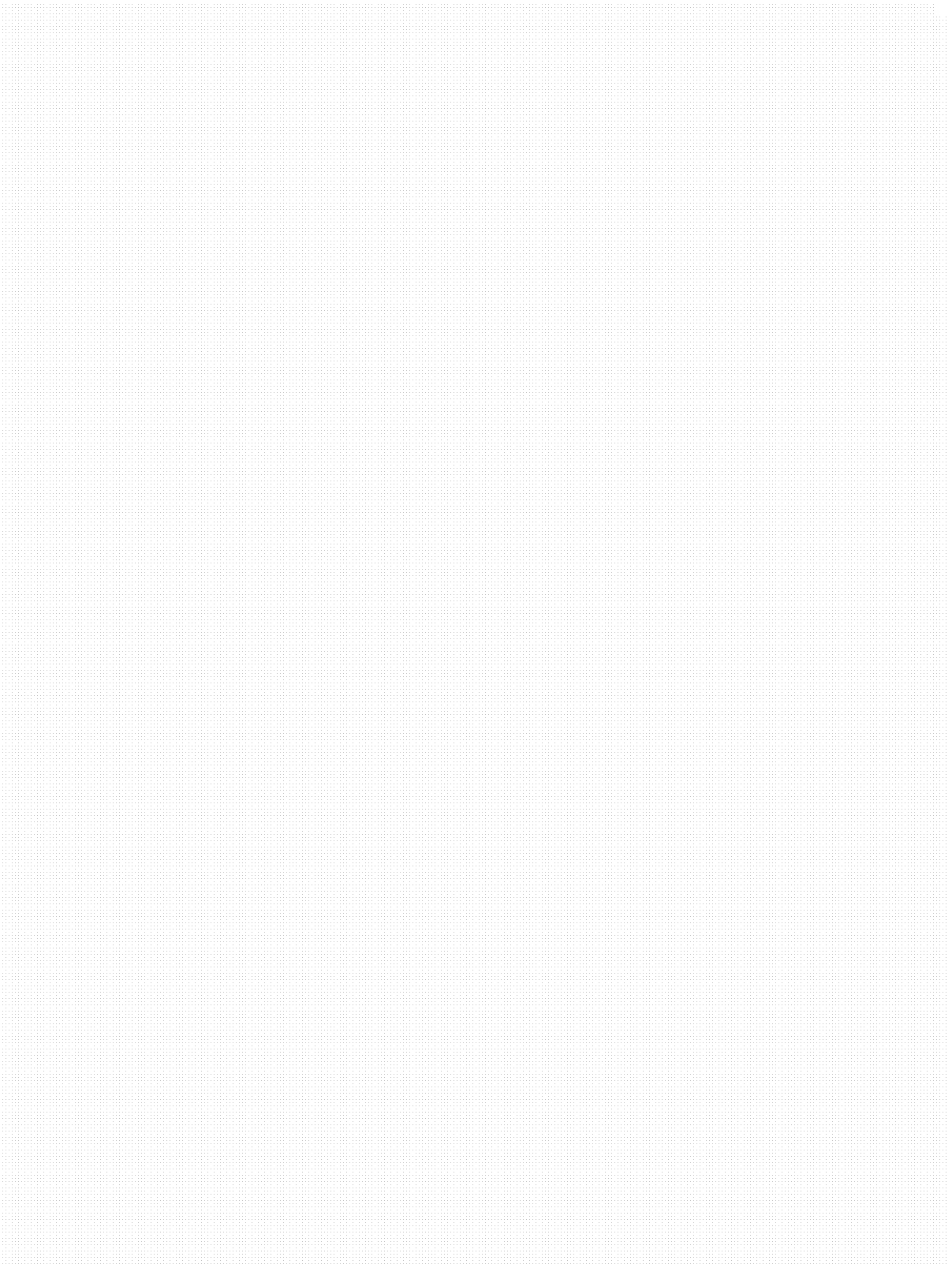
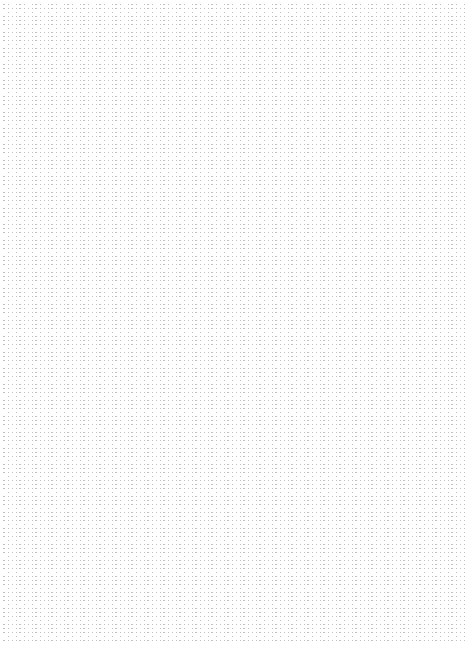
y1=value1&key2=value2



requests.request(**method**, **url**, **\*\*kwargs**)

# **∙ \*\*kwargs**: 控制访问的参数，均为可选项

params : 字典或字节序列，作为参数增加到url中 data : 字典、字节序列或文件对象，作为Request的内容

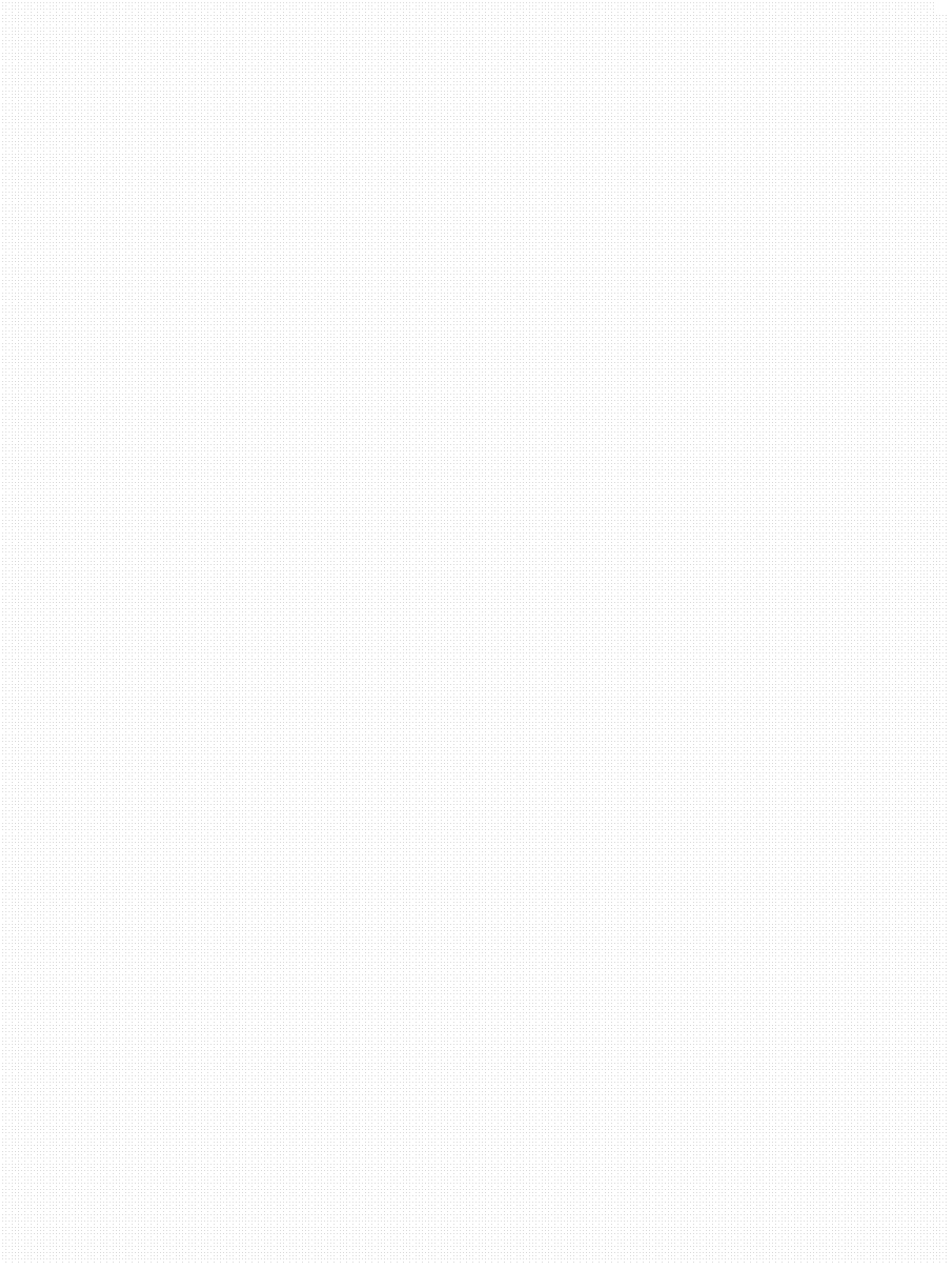
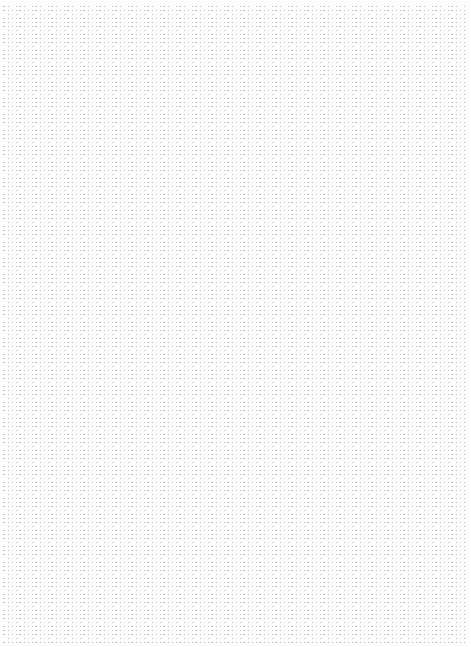


|  |
| --- |
| >>> kv = **{**'key1'**:** 'value1'**,** 'key2'**:** 'value2'**}**  >>> r = requests.request**(**'POST'**,** 'http://python123.io/ws'**,** data=kv**)** >>> body = '主体内容'  >>> r = requests.request**(**'POST'**,** 'http://python123.io/ws'**,** data=body**)** |

requests.request(**method**, **url**, **\*\*kwargs**)

# **∙ \*\*kwargs**: 控制访问的参数，均为可选项

params : 字典或字节序列，作为参数增加到url中 data : 字典、字节序列或文件对象，作为Request的内容 json : JSON格式的数据，作为Request的内容

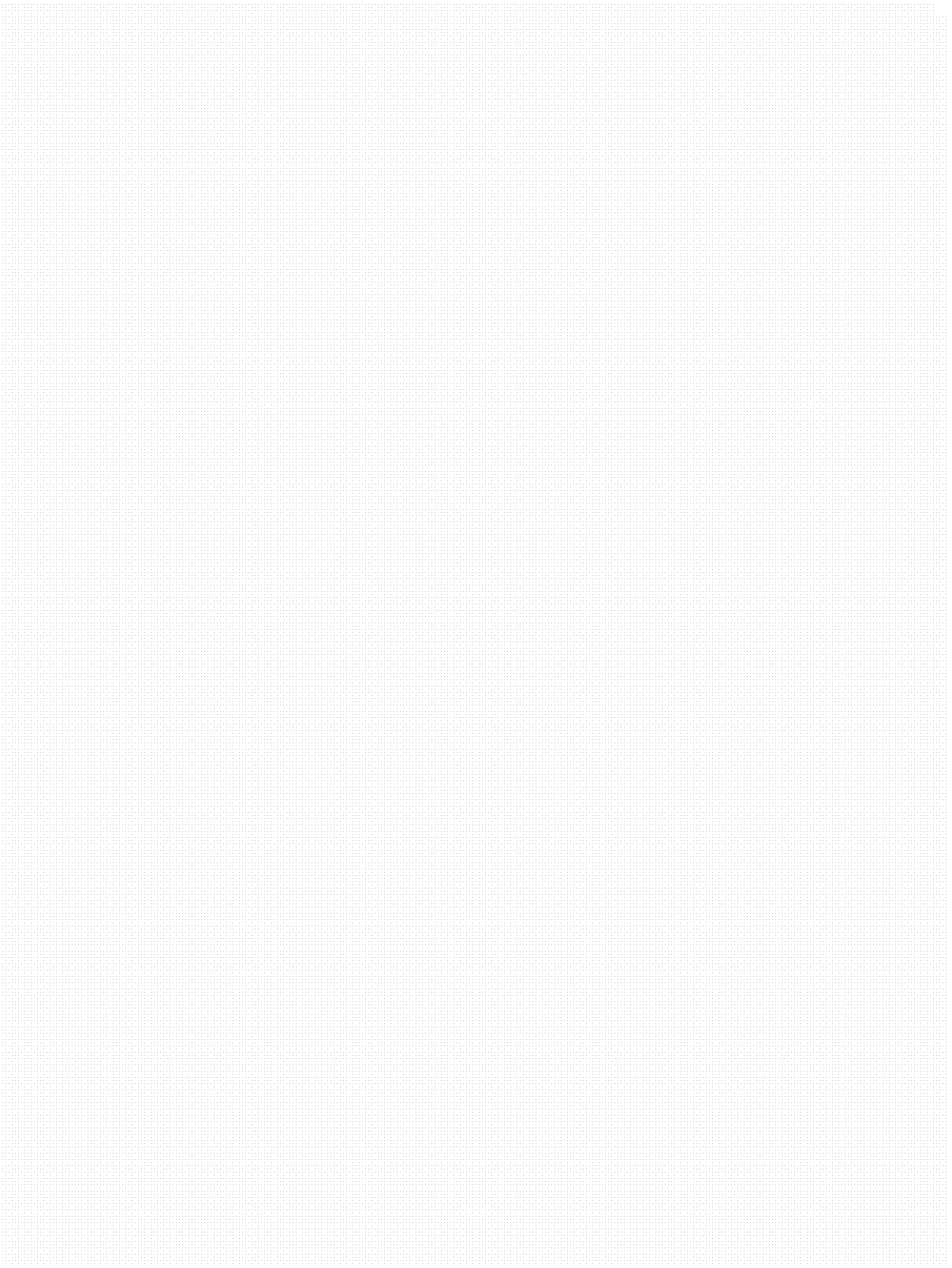


|  |
| --- |
| >>> kv = **{**'key1'**:** 'value1'**}**    >>> r = requests.request**(**'POST'**,** 'http://python123.io/ws'**,** json=kv**)** |

requests.request(**method**, **url**, **\*\*kwargs**)

# **∙ \*\*kwargs**: 控制访问的参数，均为可选项

params : 字典或字节序列，作为参数增加到url中 data : 字典、字节序列或文件对象，作为Request的内容 json : JSON格式的数据，作为Request的内容



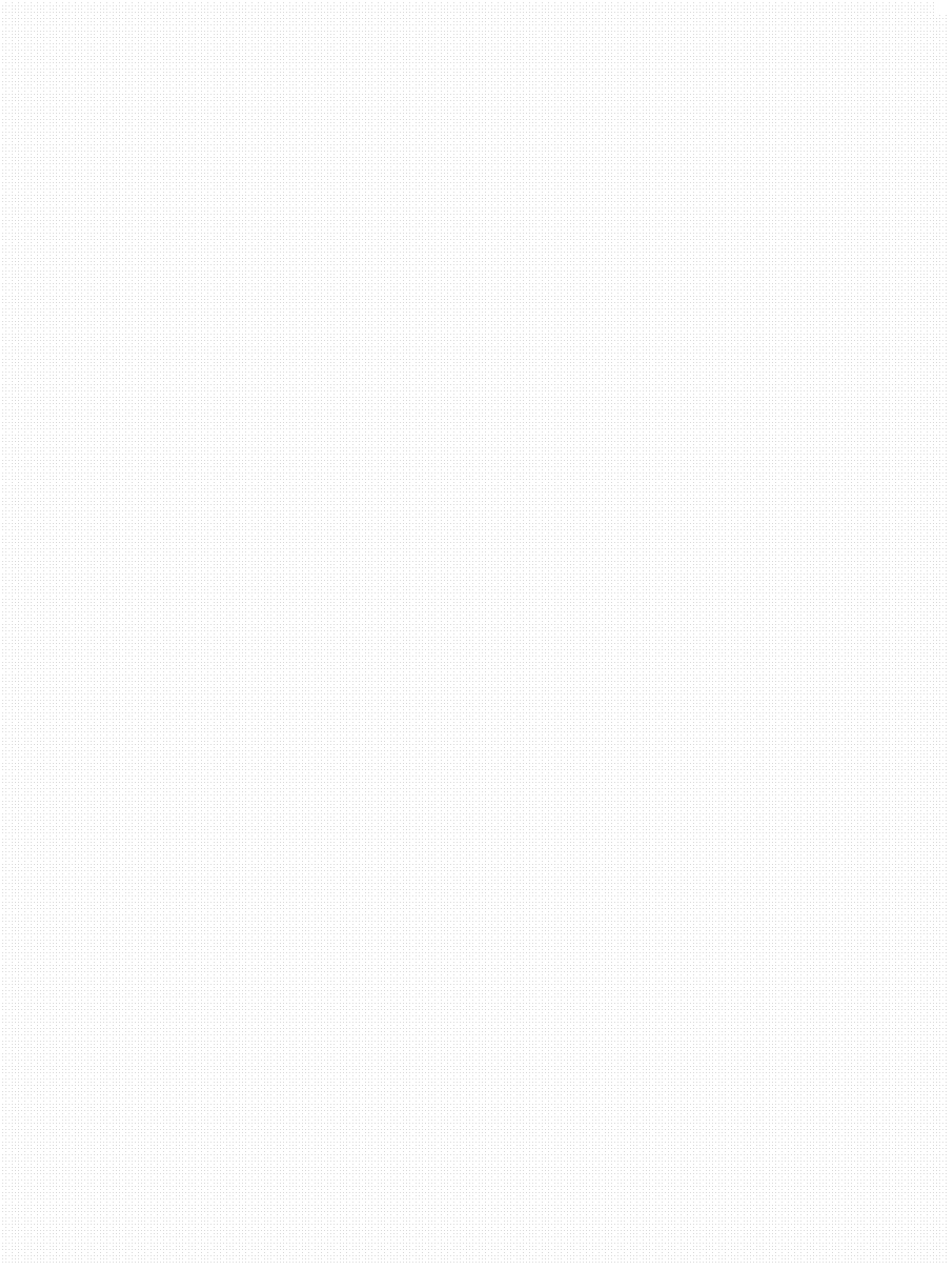
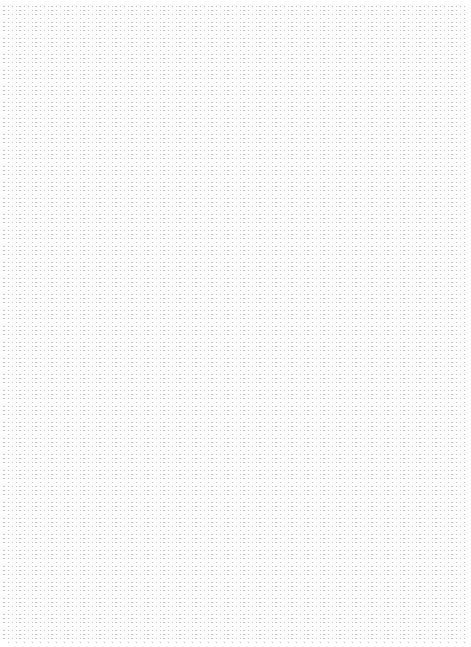
headers : 字典，HTTP定制头

|  |
| --- |
| >>> hd = **{**'user‐agent'**:** 'Chrome/10'**}**  >>> r = requests.request**(**'POST'**,** 'http://python123.io/ws'**,** headers=hd**)** |

requests.request(**method**, **url**, **\*\*kwargs**)

# **∙ \*\*kwargs**: 控制访问的参数，均为可选项

params : 字典或字节序列，作为参数增加到url中 data : 字典、字节序列或文件对象，作为Request的内容 json : JSON格式的数据，作为Request的内容 headers : 字典，HTTP定制头



cookies : 字典或CookieJar，Request中的cookie

auth : 元组，支持HTTP认证功能

requests.request(

**method**

,

**url**

,

**\*\*kwargs**

)

**∙**

**\*\*kwargs**

:

控制访问的参数（续）

files

:

字典类型，传输文件

>>>

fs

=

**{**

'file'

**:**

open(

'data.xls'

**,**

'rb'

**)}**

>>>

r

=

requests

.

request

**(**

'POST'

**,**

'http://python123.io/ws'

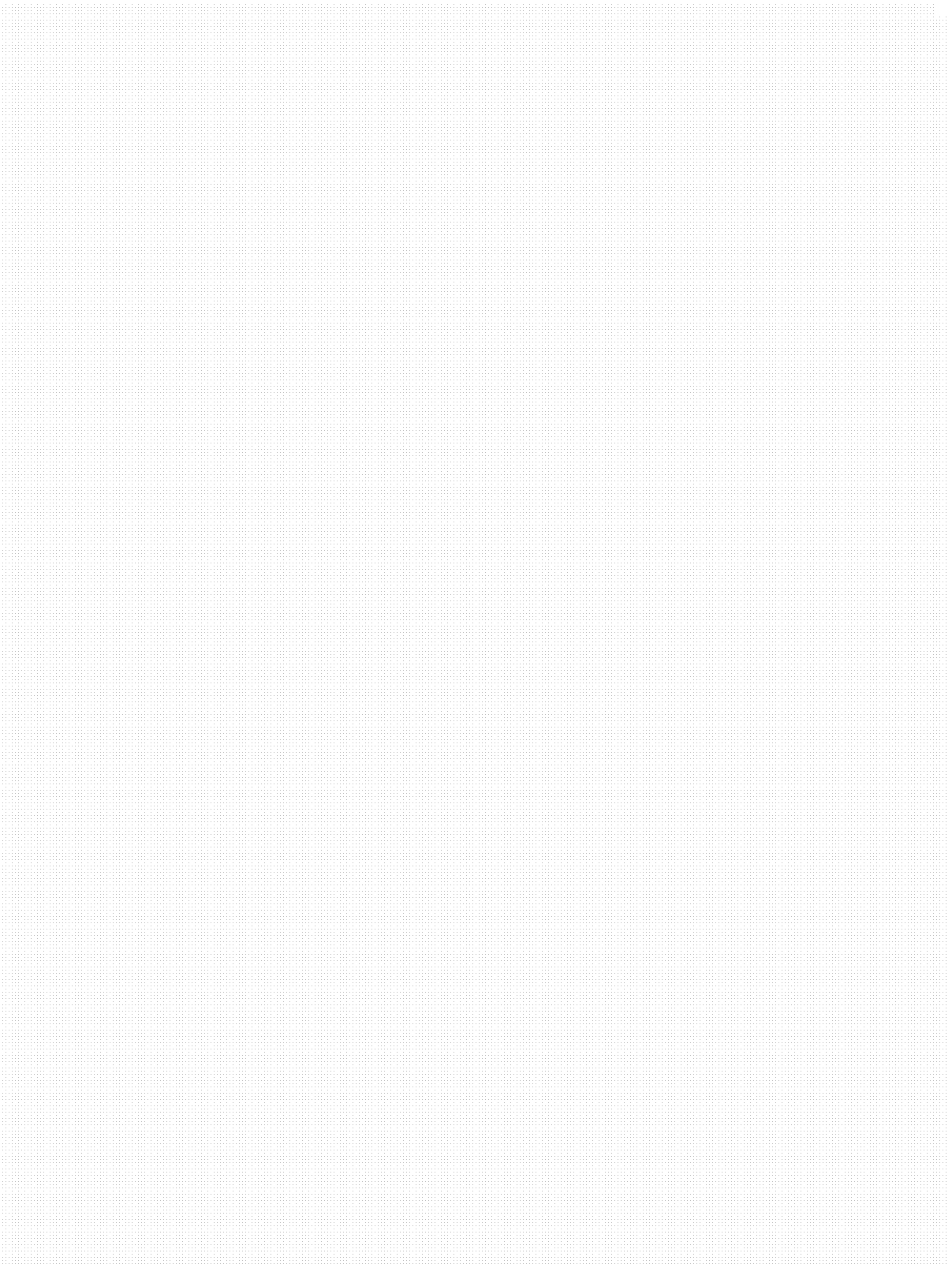
**,**

files

=

fs

**)**



requests.request(

**method**

,

**url**

,

**\*\*kwargs**

)

**∙**

**\*\*kwargs**

:

控制访问的参数（续）

files

:

字典类型，传输文件

timeout

:

设定超时时间，秒为单位

>>>

r

=

requests

.

request

**(**

'GET'

**,**

'http://www.baidu.com'

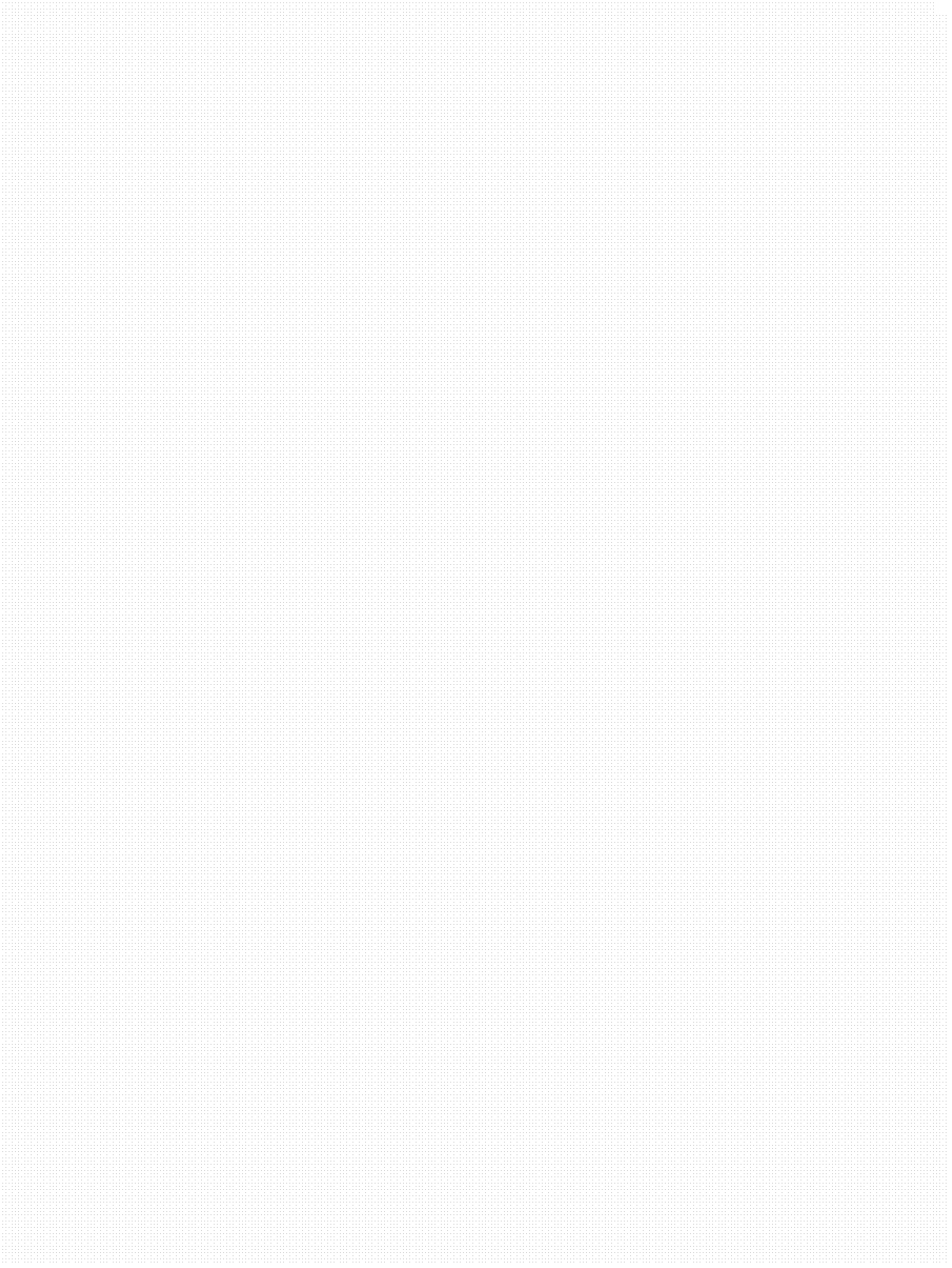
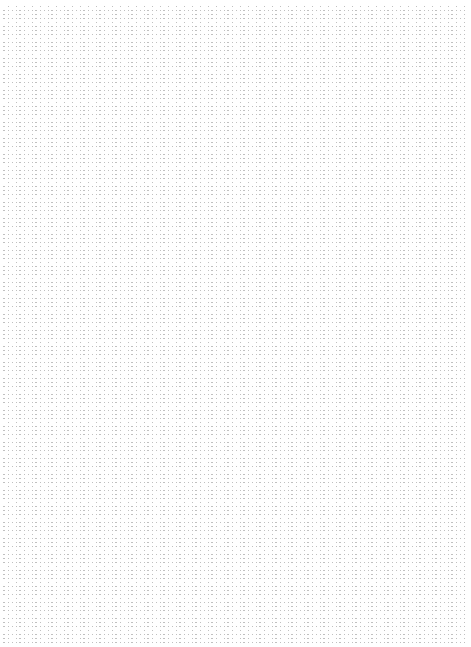
**,**

timeout

=

10

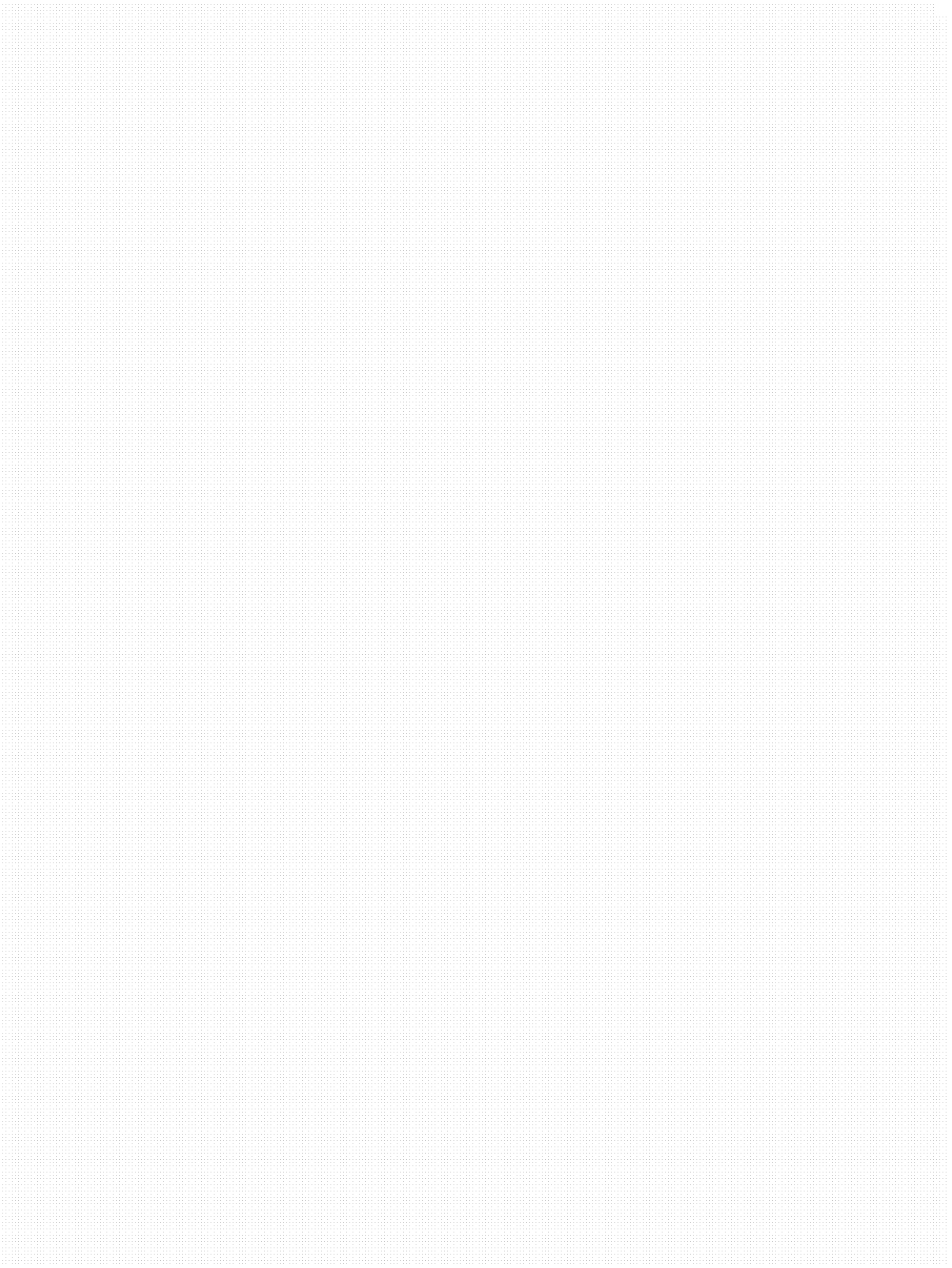
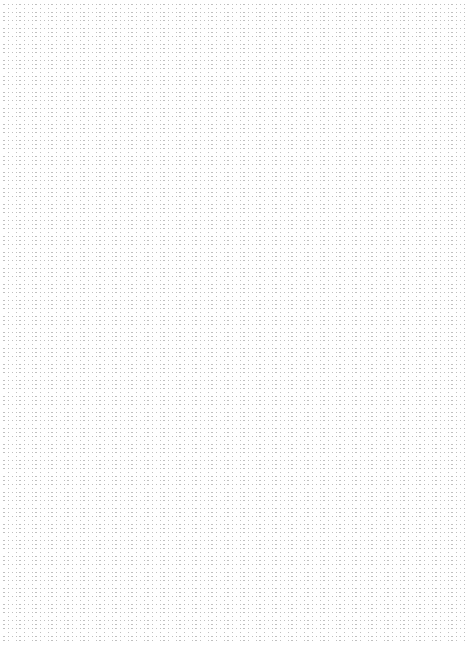
**)**



requests.request(**method**, **url**, **\*\*kwargs**)

# **∙ \*\*kwargs**: 控制访问的参数（续）

files : 字典类型，传输文件 timeout : 设定超时时间，秒为单位 proxies : 字典类型，设定访问代理服务器，可以增加登录认证



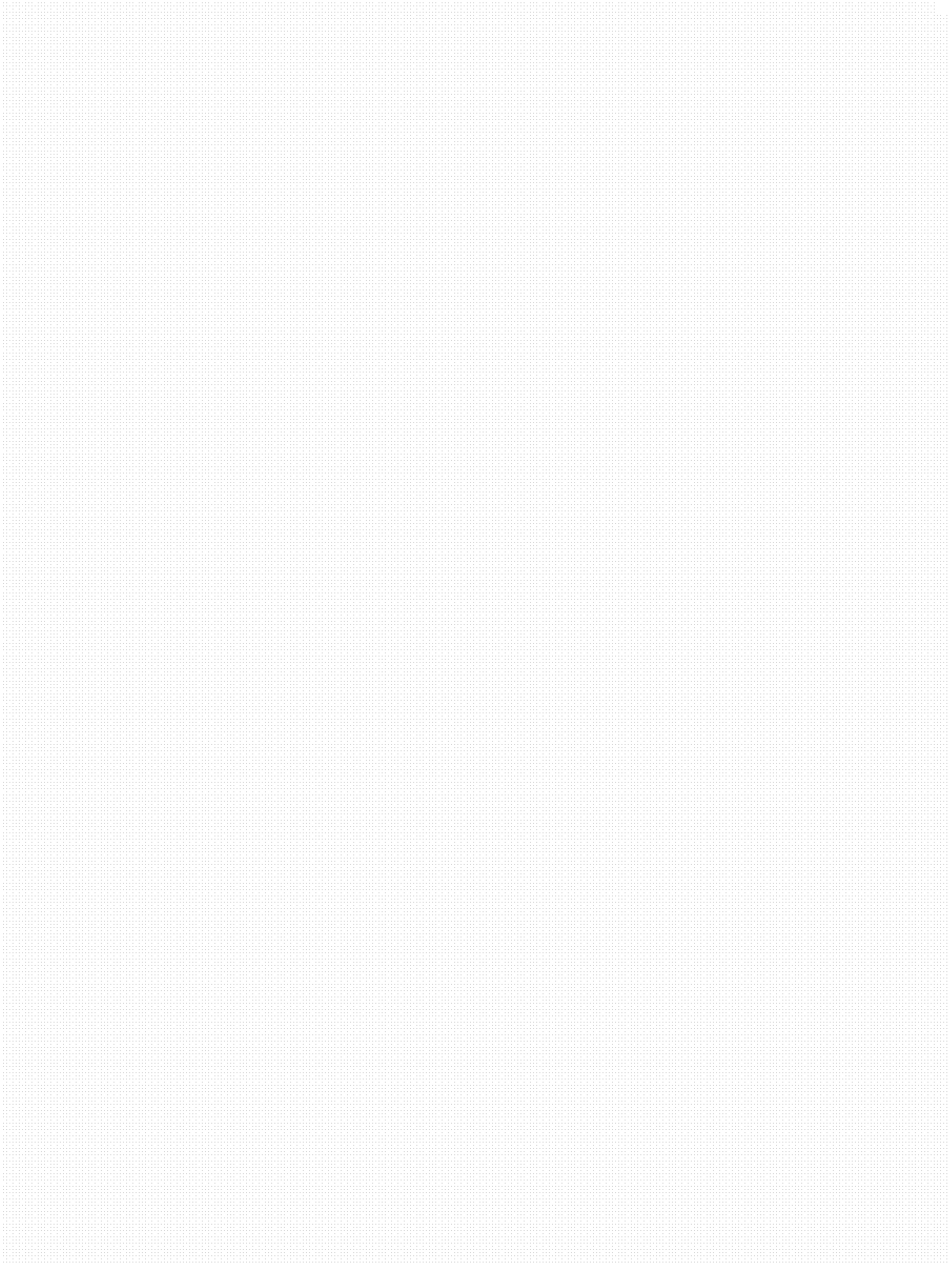
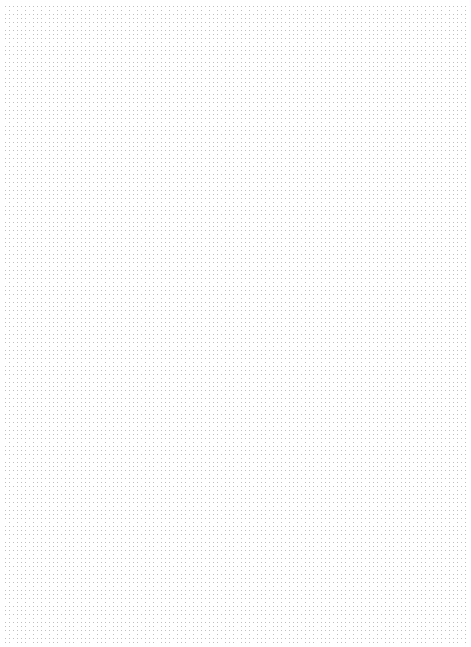
|  |
| --- |
| >>> pxs = **{** 'http'**:** 'http://user:pass@10.10.10.1:1234'  'https'**:** 'https://10.10.10.1:4321' **}**  >>> r = requests.request**(**'GET'**,** 'http://www.baidu.com'**,** proxies=pxs**)** |

requests.request(**method**, **url**, **\*\*kwargs**)

# **∙ \*\*kwargs**: 控制访问的参数（续）

files : 字典类型，传输文件 timeout : 设定超时时间，秒为单位 proxies : 字典类型，设定访问代理服务器，可以增加登录认证

allow\_redirects : True/False，默认为True，重定向开关 stream : True/False，默认为True，获取内容立即下载开关 verify : True/False，默认为True，认证SSL证书开关 cert : 本地SSL证书路径



params data json

requests.request(

**method**

,

**url**

,

**\*\*kwargs**

)

**∙**

**\*\*kwargs**

:

控制访问的参数，均为可选项

headers

cookies

auth

files

timeout

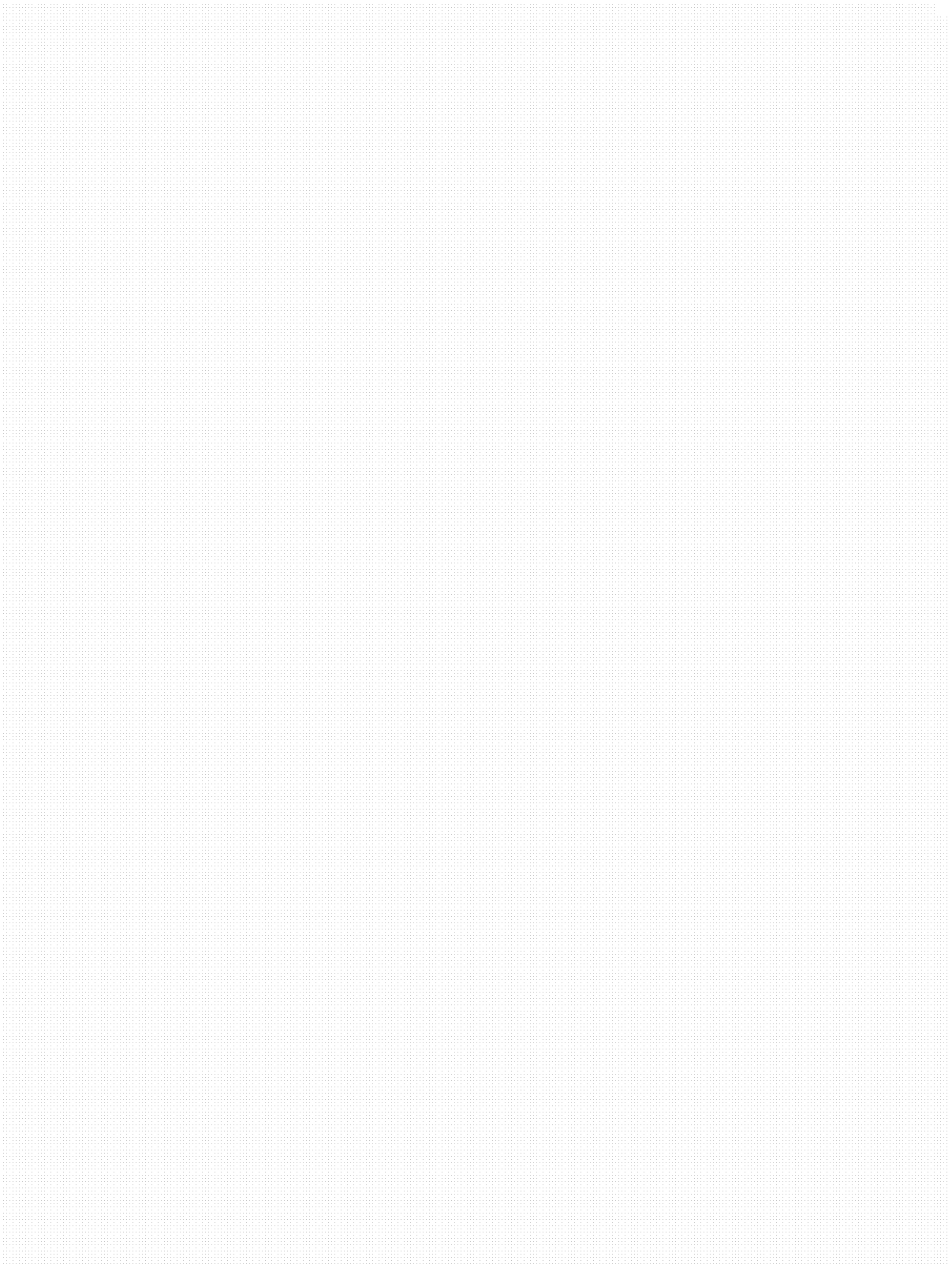
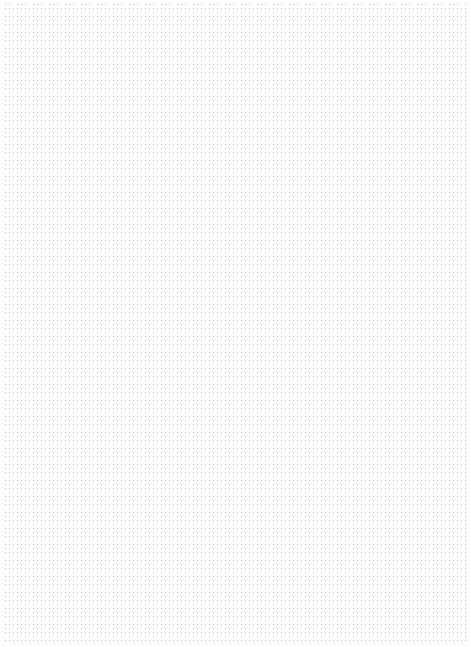
proxies

allow\_redirects

stream

verify

cert



**∙ url ∙ url ∙ url**

requests.post(

**url**

,

**data=None**

,

**json=None**

,

**\*\*kwargs**

)

:

拟更新页面的

url

链接

:

字典、字节序列或文件，

Request

的内容

:

JSON

格式的数据，

Request

的内容

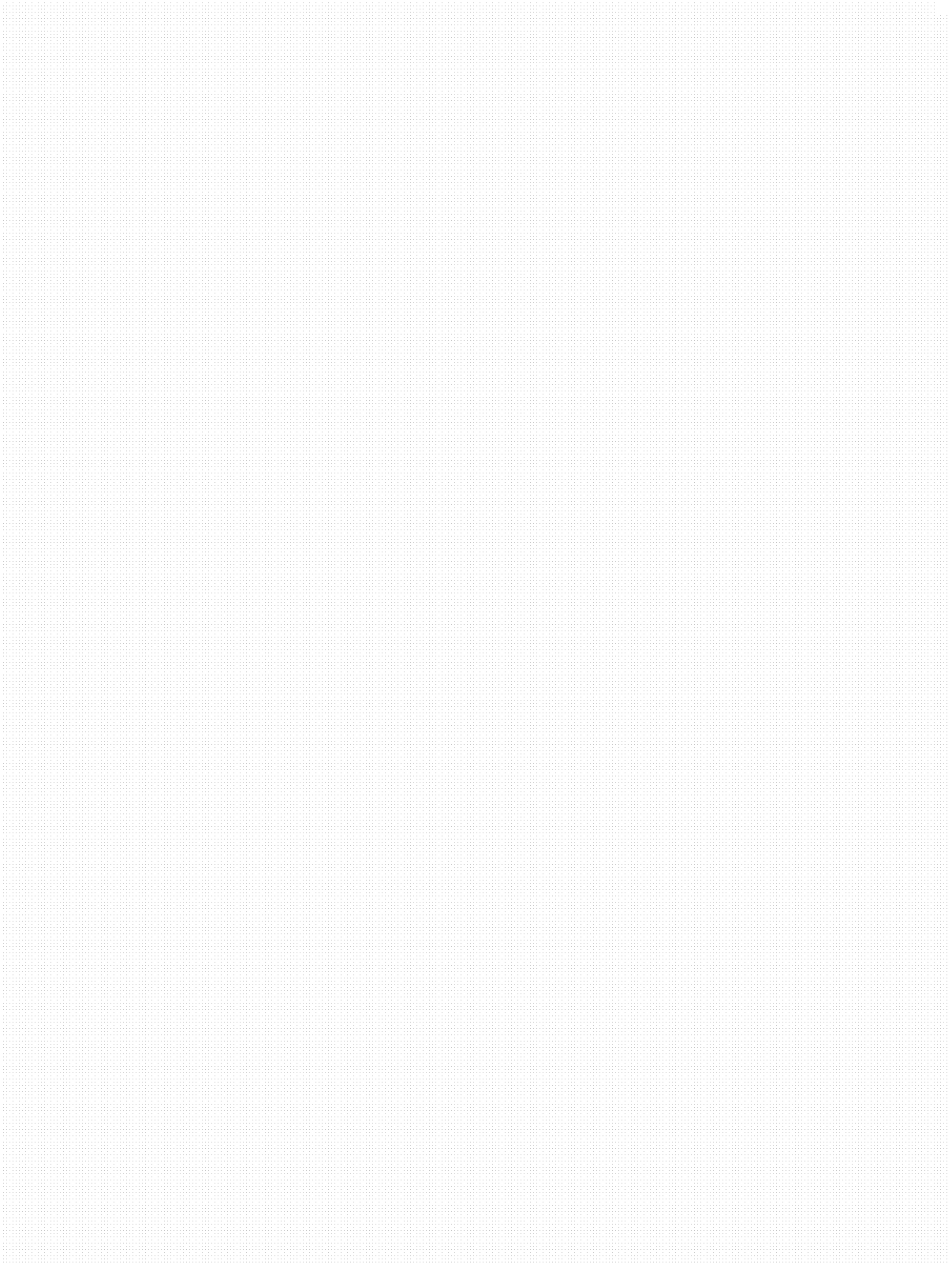
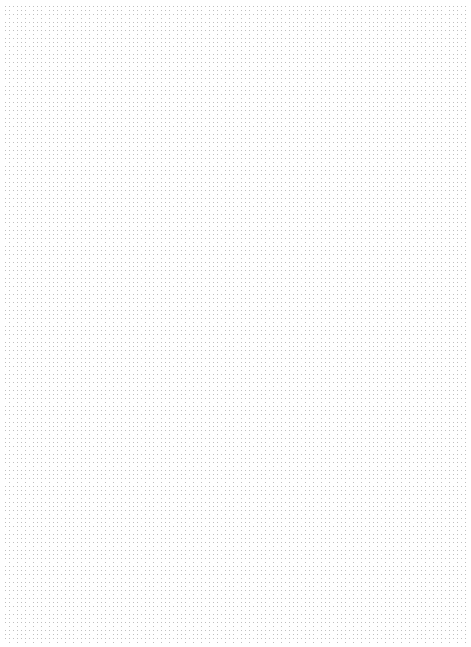
**∙**

**\*\*kwargs**

:

12

个控制访问的参数



requests.get(

**url**

,

**params=None**

,

**\*\*kwargs**

)

:

拟获取页面的

url

链接

**∙**

**params**

:

url

中的额外参数，字典或字节流格式，可选

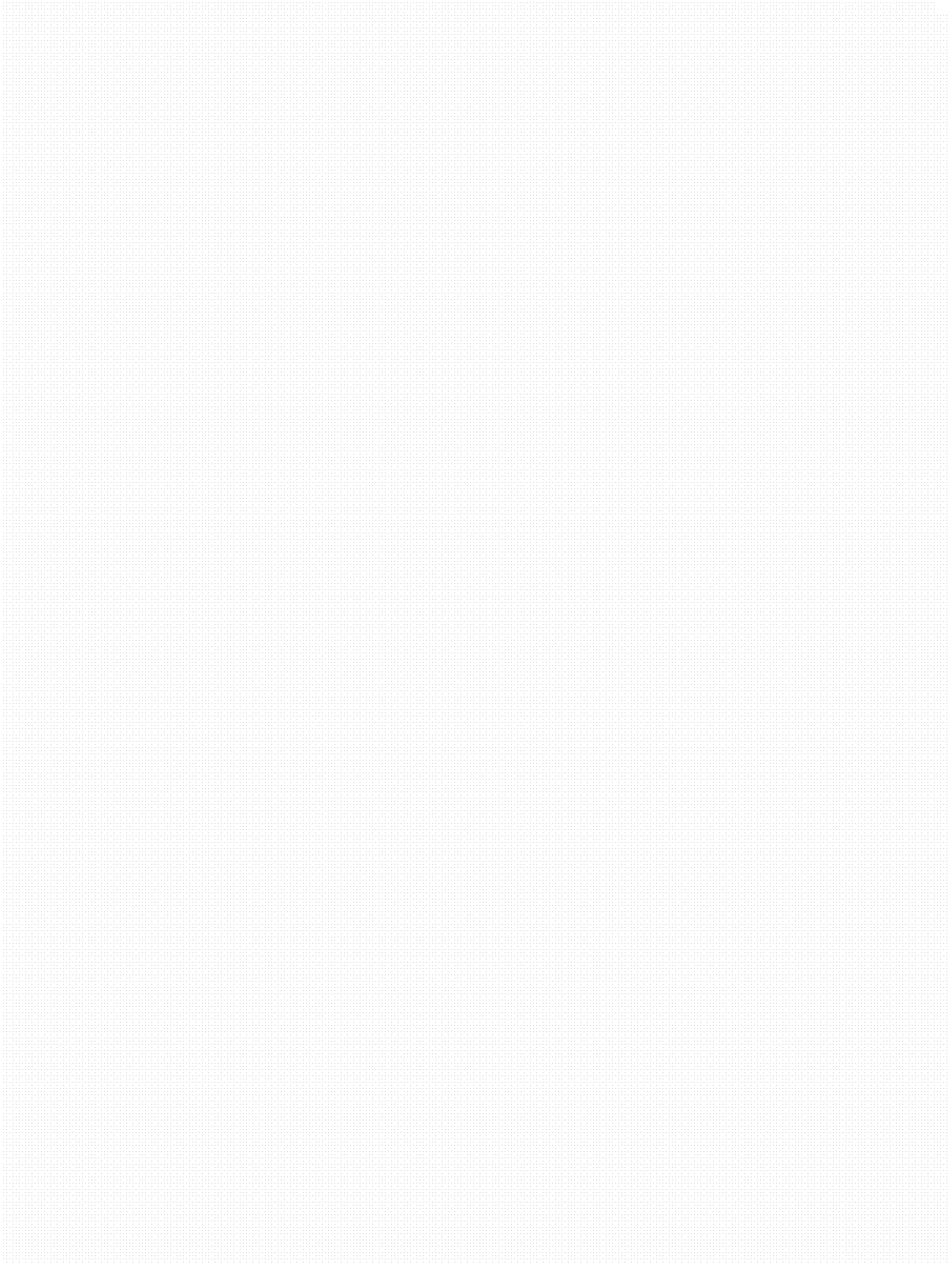
**∙**

**\*\*kwargs**

:

12

个控制访问的参数



requests.head(

**url**

,

**\*\*kwargs**

)

:

拟获取页面的

url

链接

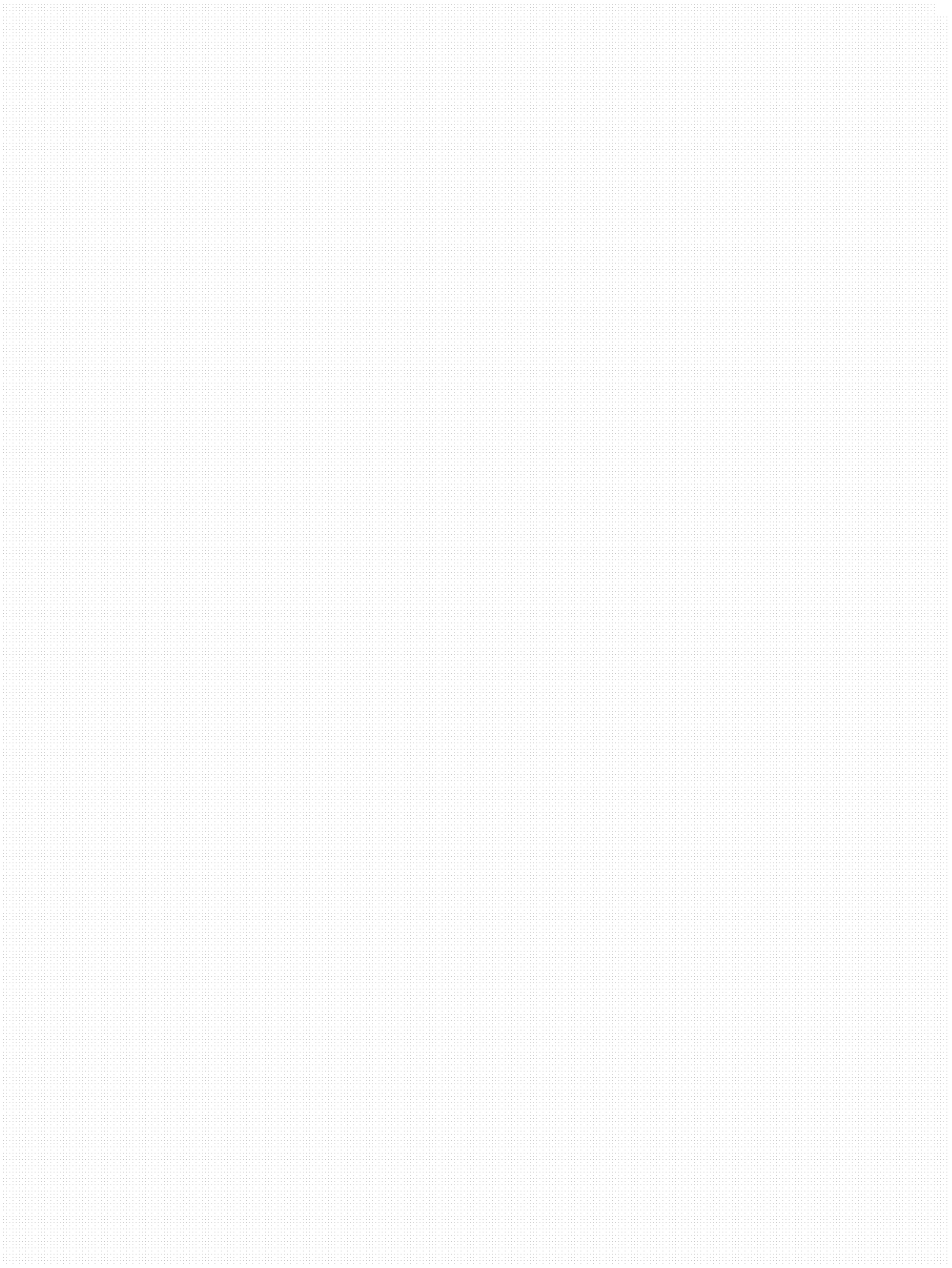
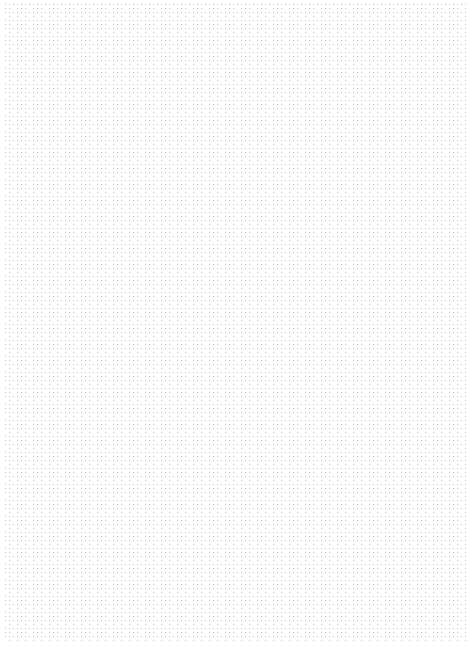
**∙**

**\*\*kwargs**

:

12

个控制访问的参数



**∙ data**

**∙ json**

**∙ url**

requests.put(

**url**

,

**data=None**

,

**\*\*kwargs**

)

:

拟更新页面的

url

链接

:

字典、字节序列或文件，

Request

的内容

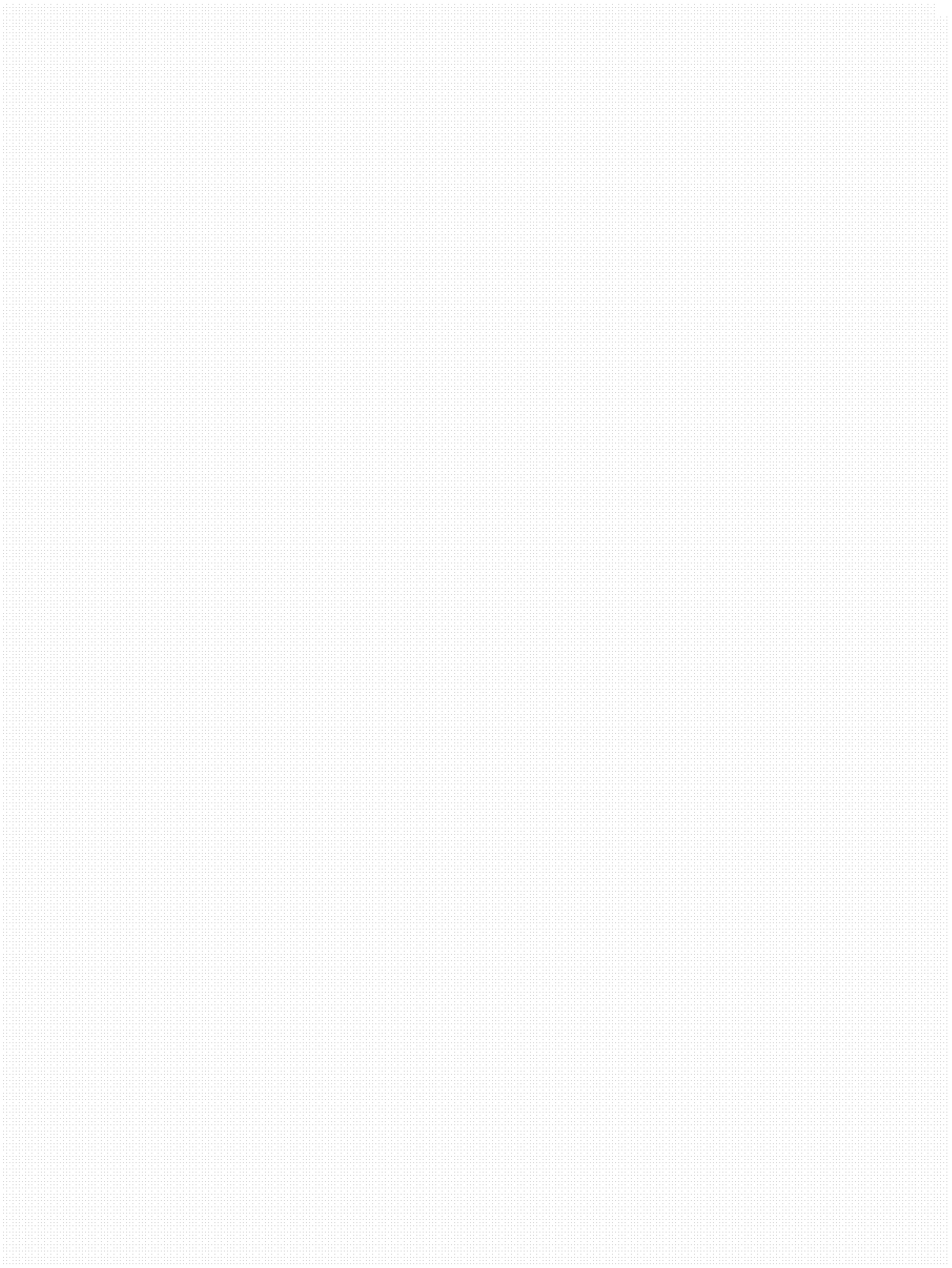
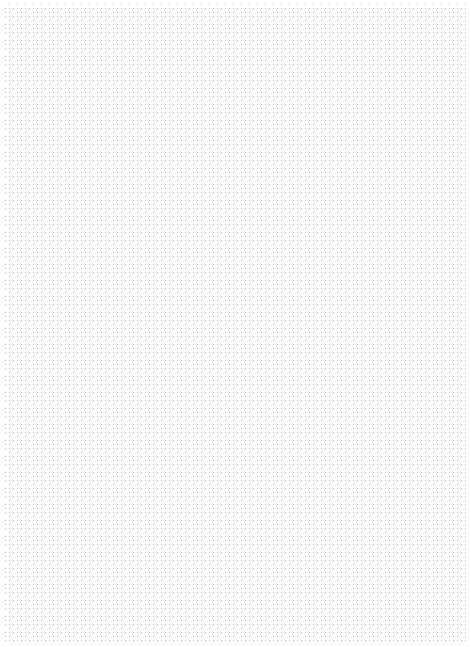
**∙**

**\*\*kwargs**

:

12

个控制访问的参数



**∙ data ∙ url**

requests.patch(

**url**

,

**data=None**

,

**\*\*kwargs**

)

:

拟更新页面的

url

链接

:

字典、字节序列或文件，

Request

的内容

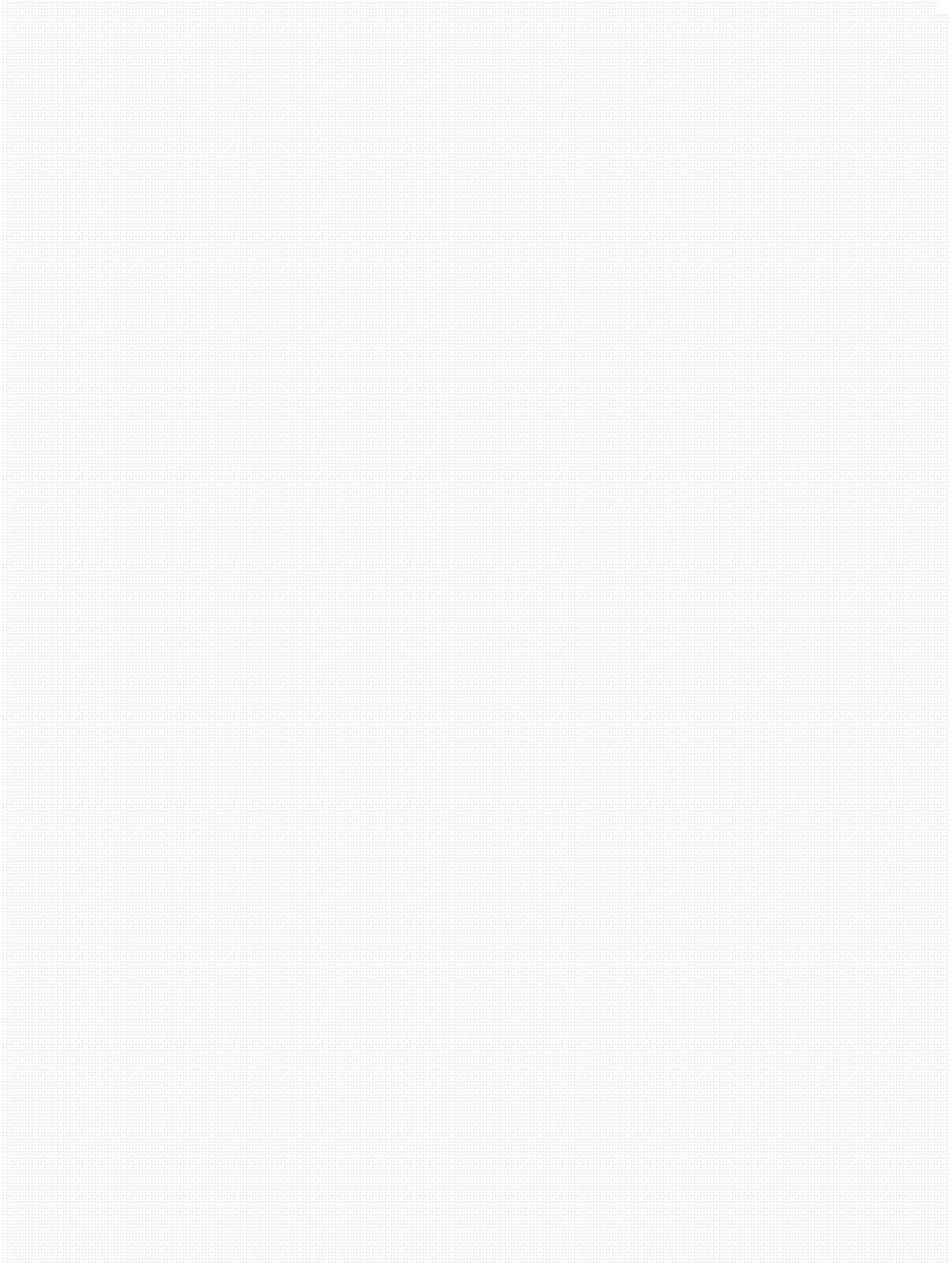
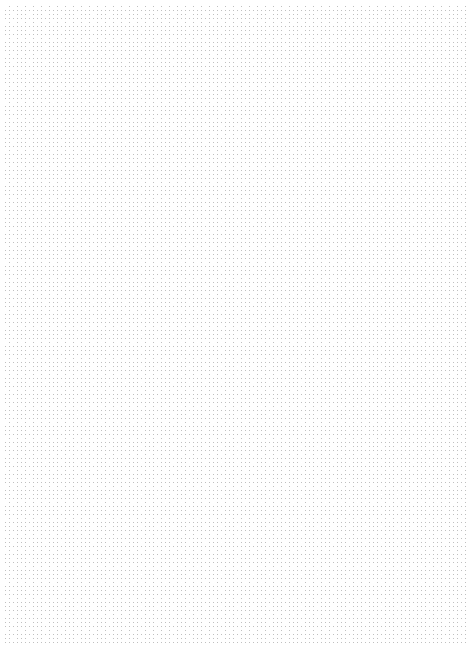
**∙**

**\*\*kwargs**

:

12

个控制访问的参数



**∙ data ∙ url ∙ url**

requests.delete(

**url**

,

**\*\*kwargs**

)

:

拟删除页面的

url

链接

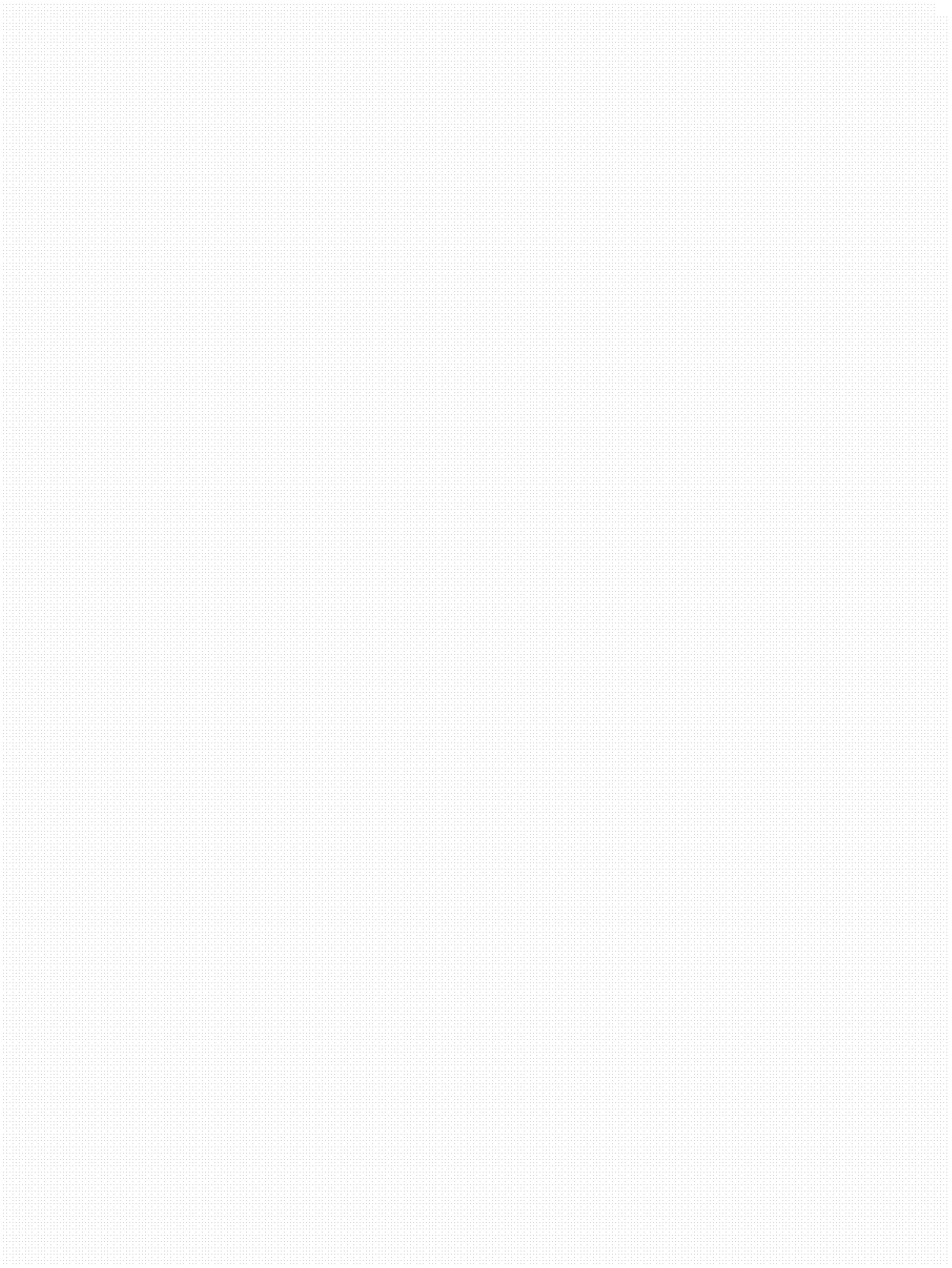
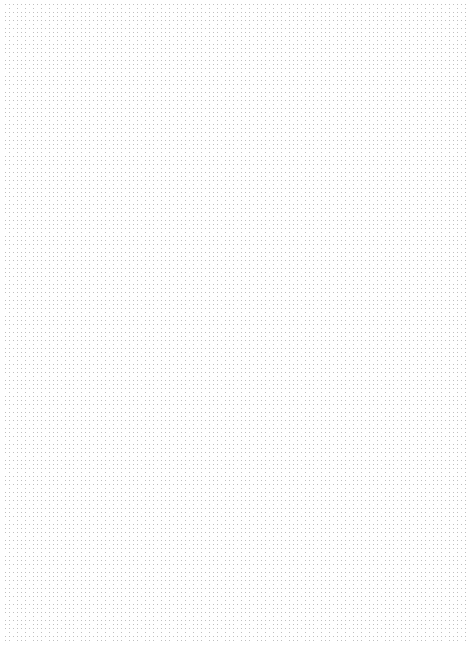
**∙**

**\*\*kwargs**

:

12

个控制访问的参数



requests.get(

**url**

,

**params=None**

,

**\*\*kwargs**

)

:

拟获取页面的

url

链接

**∙**

**params**

:

url

中的额外参数，字典或字节流格式，可选

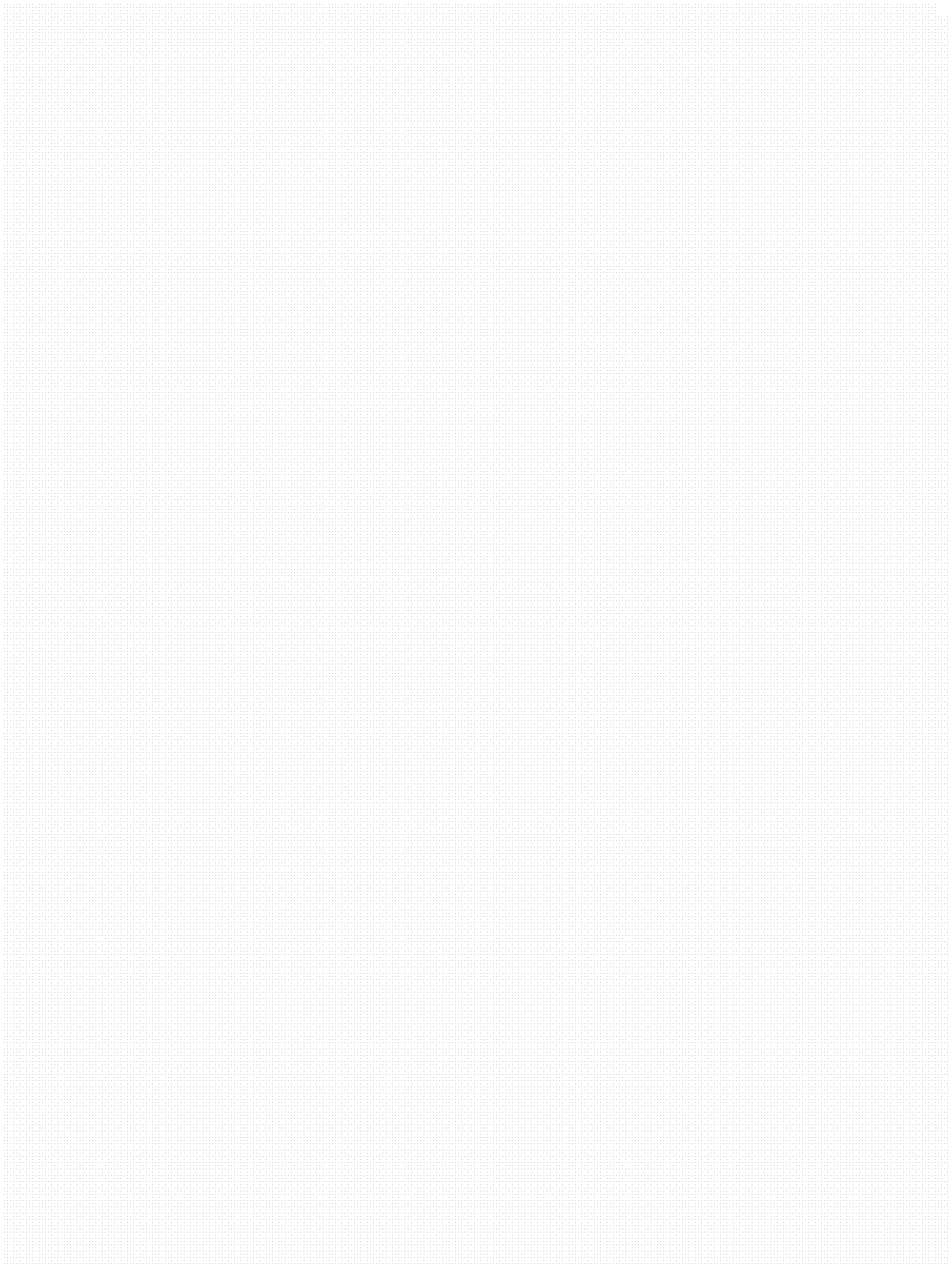
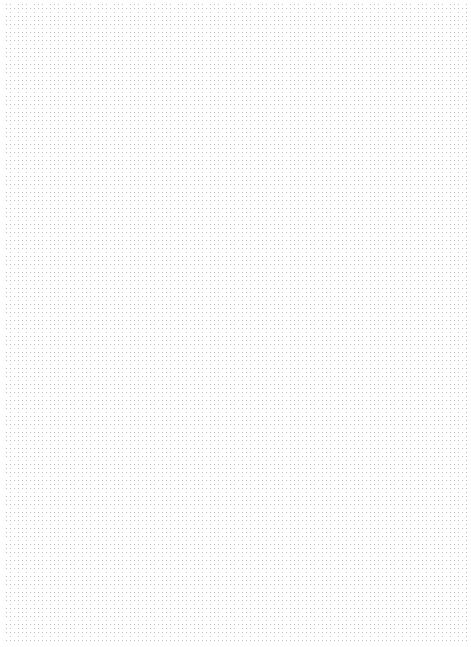
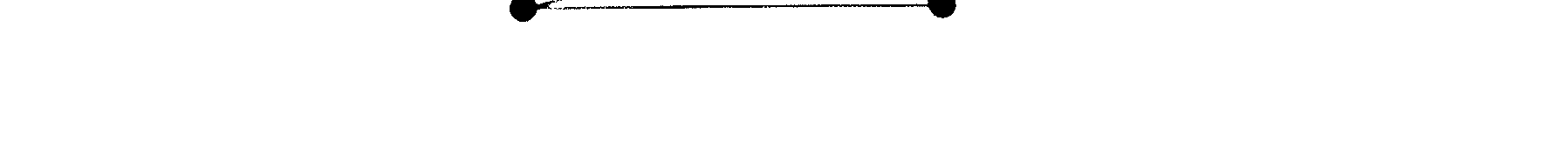
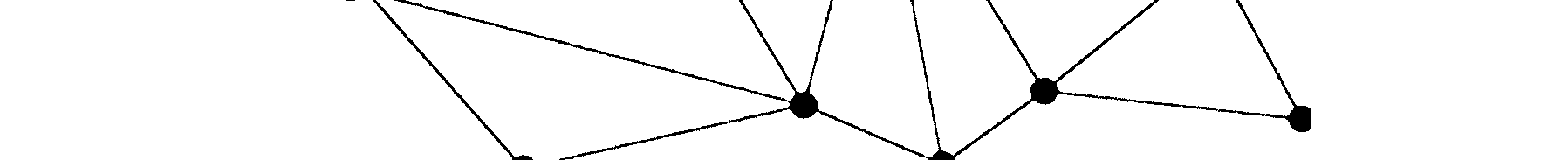
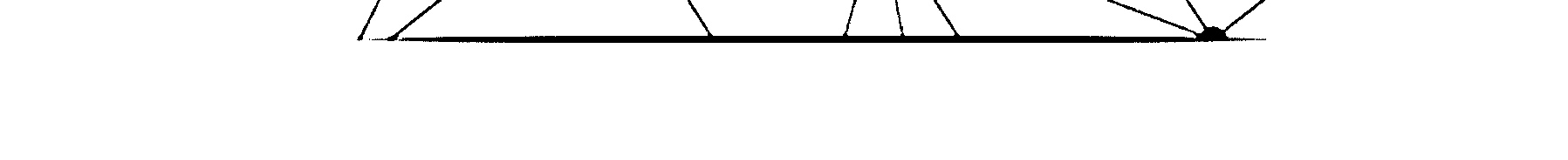
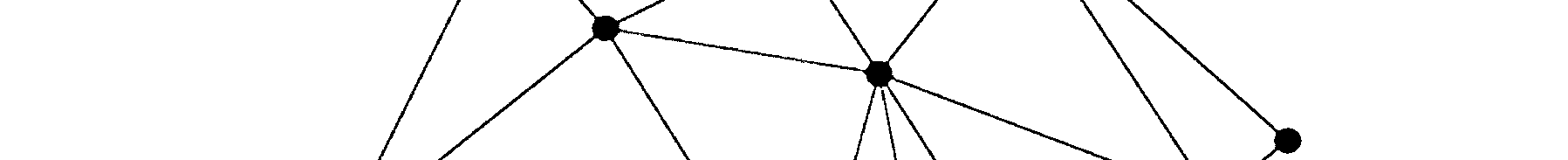
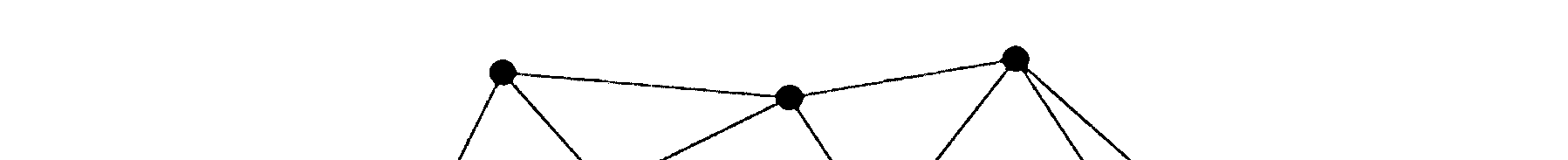
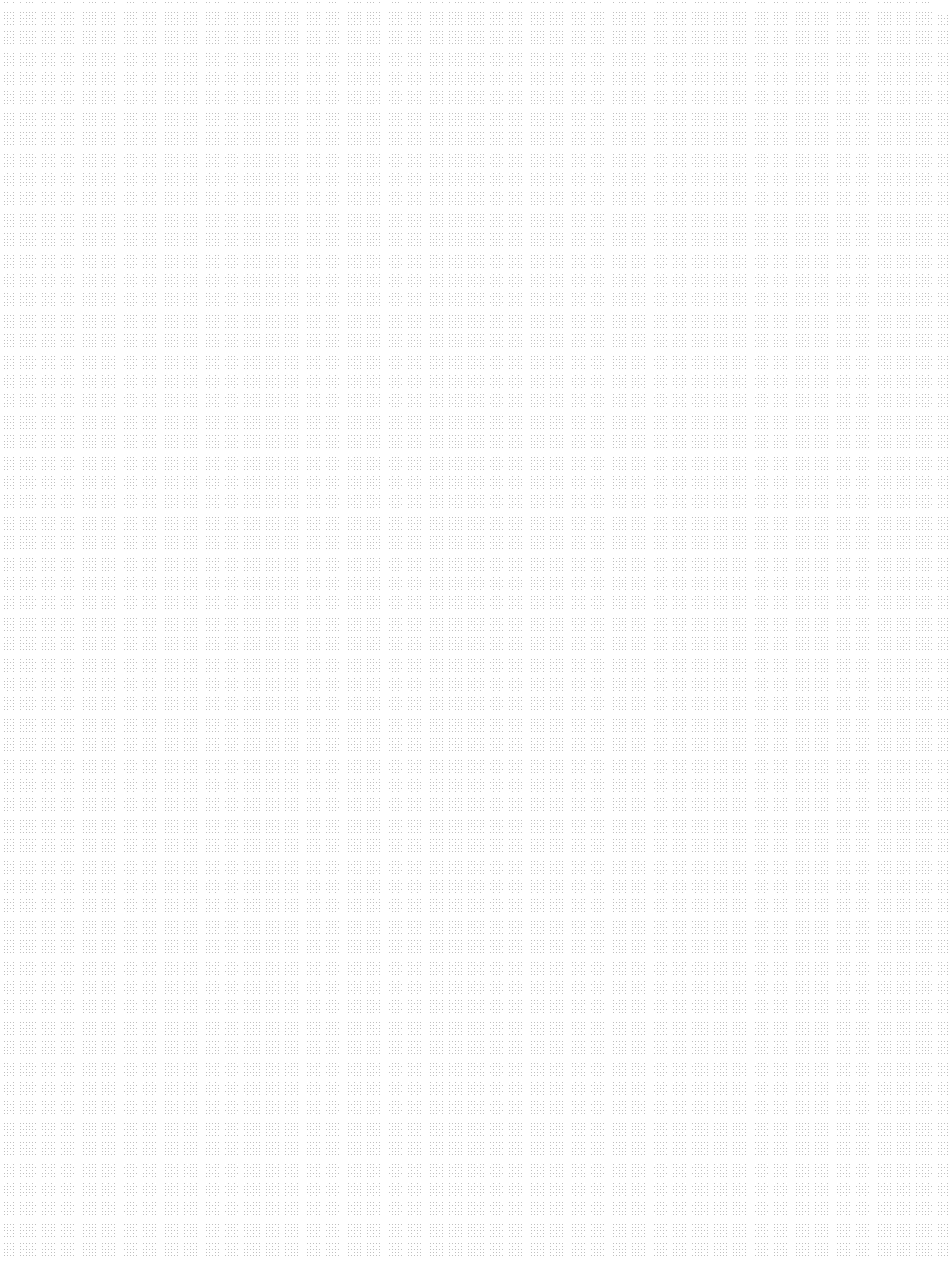
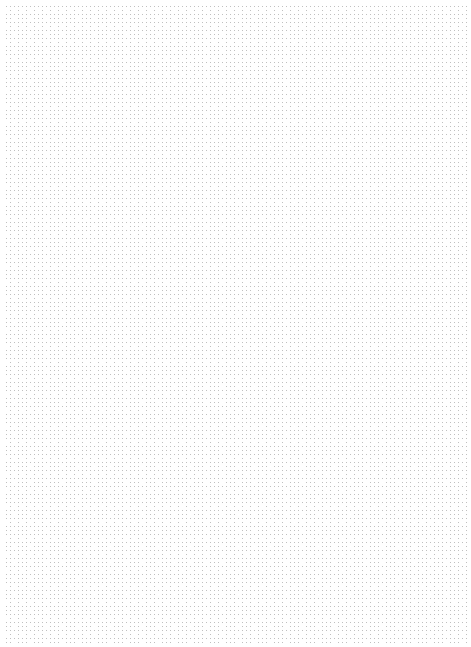
**∙**

**\*\*kwargs**

:

12

个控制访问的参数



requests.request()

requests.get()

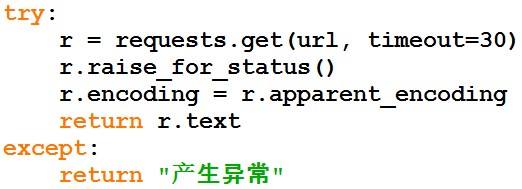
requests.head()

requests.post()

requests.put()

requests.patch()

requests.delete()



爬取网页的通用代码框架

