Click Here

Python3 my_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] 2 6 2 6 Python my_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]extracted_elements = my_list[2:7]print(extracted_elements) my_list[2:7] Python 2 2 7 7 2345 6 print(extracted_elements) Python3 Python Web Scraping Python HTTP Python HTTP HTTP HTML requests HTML HTML BeautifulSouplxmlScrapy HTML CSV JSON BeautifulSoup HTML XML Python BeautifulSoup BeautifulSoup Python HTML XML BeautifulSoup API BeautifulSoup BeautifulSoup beautifulsoup4 lxml html.parser HTML pip pip install beautifulsoup4pip install lxml # lxml lxml Python html.parser BeautifulSoup HTML XML BeautifulSoup BeautifulSoup BeautifulSoup HTML BeautifulSoup requests: from bs4 import BeautifulSoupimport requests# requests url = ' # bingresponse = requests.get(url)# BeautifulSoup soup = BeautifulSoup(response.text, 'lxml') # lxml # html.parser # soup = BeautifulSoup(response.text, 'html.parser') from bs4 import BeautifulSoupimport requests# url = ' # bing# HTTPresponse = requests.get(url)# response.encoding = 'utf-8' # if response.status_code == 200: # BeautifulSoup soup = BeautifulSoup(response.text, 'lxml') # title_tag = soup.find('title') # if title_tag: print(title_tag.get_text()) else: print("")else: print("", response.status_code) - Microsoft requests requests Content-Type chardet import requestsurl = ' response = requests.get(url)# chardet import chardetencoding = chardet.detect(response.content)['encoding']print(encoding)response.encoding = encoding utf-8 utf-8 gbk response.encoding: response.encoding = 'utf-8' # 'gbk' BeautifulSoup find() find_all() find() find_all() from bs4 import BeautifulSoupimport requests# url = ' # bing# HTTPresponse = requests.get(url)# response.encoding = 'utf-8'soup = BeautifulSoup(response.text, 'lxml')# first_link = soup.find('a')print(first_link)print("----------")# href first_link_url = first_link.get('href')print(first_link_url)print("----------")# all_links = soup.find_all('a')print(all_links) ---------- [, hao123, , , get_text() from bs4 import BeautifulSoupimport requests# url = ' # bing# HTTPresponse = requests.get(url)# response.encoding = 'utf-8'soup = BeautifulSoup(response.text, 'lxml')# paragraph_text = soup.find('p').get_text()# all_text = soup.get_text()print(all_text) ... parent children # parent_tag = first_link.parent# children = first_link.children from bs4 import BeautifulSoupimport requests# url = ' # bing# HTTPresponse = requests.get(url)# response.encoding = 'utf-8'soup = BeautifulSoup(response.text, 'lxml')# first_link = soup.find('a')print(first_link)print("----------")# parent_tag = first_link.parentprint(parent_tag.get_text()) ---------- hao123 example-class div # class="example-class" divs with_class = soup.find_all('div', class_='example-class')# id="unique-id" unique_paragraph = soup.find('p', id='unique-id') id su from bs4 import BeautifulSoupimport requests# url = ' # bing# HTTPresponse = requests.get(url)# response.encoding = 'utf-8'soup = BeautifulSoup(response.text, 'lxml')# id="unique-id" unique_input = soup.find('input', id='su')input_value = unique_input['value'] # input print(input_value) CSS BeautifulSoup CSS select() jQuery # CSS class 'example' example_divs = soup.select('div.example')# href links = soup.select('a[href]') BeautifulSoup HTML # nested_divs = soup.find_all('div', class_='nested')for div in nested_divs: print(div.get_text()) BeautifulSoup HTML # href first_link['href'] = ' '# first_paragraph = soup.find('p')first_paragraph.string = 'Updated content'# first_paragraph.decompose() BeautifulSoup HTML # html_str = str(soup) BeautifulSoup BeautifulSoup :/BeautifulSoup() HTML XML BeautifulSoup soup = BeautifulSoup(html_doc, 'html.parser').prettify()print(soup.prettify()).find()tag = soup.find('a').find_all()tags = soup.find_all('a').find_all_next()tags = soup.find('div').find_all_next('p').find_all_previous()tags = soup.find('div').find_all_previous('p').find_parent()parent = tag.find_parent().find_all_parents()parents = tag.find_all_parents().find_next_sibling()next_sibling = tag.find_next_sibling().find_previous_sibling()prev_sibling = tag.find_previous_sibling().parentparent = tag.parent.next_siblingnext_sibling = tag.next_sibling.previous_siblingprev_sibling = tag.previous_sibling.get_text()HTMLtext = tag.get_text().attrshref = tag.attrs['href'].stringstring_content = tag.string.nametag_name = tag.name.contentschildren = tag.contents.descendantsfor child in tag.descendants: print(child).parentparent = tag.parent.previous_elementprev_elem = tag.previous_element.next_elementnext_elem = tag.next_element.decompose()tag.decompose().unwrap()tag.unwrap().insert()tag.insert(0, new_tag).insert_before()tag.insert_before(new_tag).insert_after()tag.insert_after(new_tag).extract()extracted_tag = tag.extract().replace_with()tag.replace_with(new_tag).has_attr()if tag.has_attr('href'):.get()href = tag.get('href').clear()tag.clear().encode()encoded = tag.encode().is_empty_element if tag.is_empty_element:.is_ancestor_of()if tag.is_ancestor_of(another_tag):.is_descendant_of()if tag.is_descendant_of(another_tag):/.stylestyle = tag['style'].id id id = tag['id'].class_ class class_name = tag['class'].stringcontent = tag.string.parentparent = tag.parent/find_all(string)tag = soup.find_all('div', class_='container')find_all(id) id tag = soup.find_all(id='main')find_all(attrs)tag = soup.find_all(attrs={"href": " "}) Python3 Python3 range() Python3 list() range() Python2 range() range(stop)range(start, stop[, step]) start: start 0 range(5) range(0 5) stop: stop stoprange(0 5) [0, 1, 2, 3, 4] 5 step 1range(0 5) range(0, 5, 1) 1 for number in range(1, 6): print(number) 12345 0 1 for number in range(6): print(number) 012345 for number in range(1, 6, 2): print(number) 135 6 for number in range(6, 1, -1): print(number) 65432 for range() >>> numbers = list(range(1, 6))>>> numbers[1, 2, 3, 4, 5] >>> numbers = tuple(range(1, 6))>>> numbers(1, 2, 3, 4, 5) Python3 (a, b) a < x < b [a,b] a >> tinydict['b']'3'>>> tinydict{'a': 1, 'b': '3'} tinydict = {'Alice': '2341', 'Beth': '9102', 'Cecil': '3258'} tinydict1 = { 'abc': 456 }tinydict2 = { 'abc': 123, 98.6: 37 } : tinydict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'} print "tinydict['Name']: ", tinydict['Name']print "tinydict['Age']: ", tinydict['Age'] tinydict['Name']: Zaratinydict['Age']: 7 tinydict = {'Name': 'Runoob', 'Age': 7, 'Class': 'First'} print "tinydict['Alice']: ", tinydict['Alice'] tinydict['Alice']: Traceback (most recent call last): File "test.py", line 5, in print "tinydict['Alice']: ", tinydict['Alice']KeyError: 'Alice' //: tinydict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'} tinydict['Age'] = 8 tinydict['School'] = "RUNOOB" print "tinydict['Age']: ", tinydict['Age']print "tinydict['School']: ", tinydict['School'] tinydict['Age']: 8tinydict['School']: RUNOOB del tinydict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'} del tinydict['Name'] tinydict.clear() del tinydict print "tinydict['Age']: ", tinydict['Age'] print "tinydict['School']: ", tinydict['School'] del tinydict['Age']: Traceback (most recent call last): File "test.py", line 10, in print "tinydict['Age']: ", tinydict['Age'] NameError: name 'tinydict' is not defined del() python 1 tinydict = {'Name': 'Runoob', 'Age': 7, 'Name': 'Manni'} print "tinydict['Name']: ", tinydict['Name'] tinydict['Name']: Manni 2 tinydict = {['Name']: 'Zara', 'Age': 7} print "tinydict['Name']: ", tinydict['Name'] Traceback (most recent call last): File "test.py", line 3, in tinydict = {['Name']: 'Zara', 'Age': 7} TypeError: unhashable type: 'list' & Python Python Python for for for iterating_var in sequence: statements(s) for letter in 'Python': print(": %s" % letter) fruits = ['banana', 'apple', 'mango']for fruit in fruits: print (': %s'% fruit) print ("Good bye!") : : P: y: t: h: o: n: banana: apple: mangoGood bye! fruits = ['banana', 'apple', 'mango']for index in range(len(fruits)): print (' : %s' % fruits[index]) print ("Good bye!") : banana : apple : mangoGood bye! len() range(), len() range else python for else for else for break while else for num in range(10,20): for i in range(2,num): if num%i == 0: j=num/i print ('%d %d * %d' % (num,i,j)) break else: print ('%d ' % num) 10 2 * 511 12 2 * 613 14 2 * 715 3 * 516 2 * 817 18 2 * 919 python Python3 for i in range(1, 10): for j in range(1, i+1): print('{}x{}={}\t'.format(j, i, i*j), end='') print() 1x1=11x2=22x2=41x3=32x3=63x3=91x4=42x4=83x4=124x4=161x5=52x5=103x5=154x5=205x5=251x6=62x6=123x6=184x6=245x6=306x6=361x7=72x7=143x7=214x7=285x7=356x7=427x7=491x8=82x8=163x8=244x8=325x8=406x8=487x8=568x8=641x9=92x9=183x9=274x9=365x9=456x9=547x9=638x9=729x9=81 end Python3 Python random Python random.randint() randint(start, stop) randrange(start, stop+1) random.randint() random.randint(start, stop) start -- stop -- 1 10 1 10 # random import random# 110random_number = random.randint(1, 10)print(random_number) 8 random.randint(1, 10) 1 10110 random_number a > b ValueErrora b random.randint() random random Python random >>>range(10) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]>>> range(1, 11) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]>>> range(0, 30, 5) [0, 5, 10, 15, 20, 25]>>> range(0, 10, 3) [0, 3, 6, 9]>>> range(0, -10, -1) [0, -1, -2, -3, -4, -5, -6, -7, -8, -9]>>> range(0)[]>>> range(1, 0)[] pip Python Python python.org Python 2.7.9 + Python 3.4+ pip pip pip --version # Python2.x pip3 --version # Python3.x $ curl -o get-pip.py # $ sudo python get-pip.py # Python pip Python3 $ sudo python3 get-pip.py # pip Python 2.7pip3 Python 3.x Linux pip Debian Ubuntu sudo apt-get install python-pip pip pip --version pip --help pip pip install -U pip sudo easy_install --upgrade pip pip install SomePackage # pip install SomePackage==1.0.4 # pip install 'SomePackage>=1.0.4' # Django pip install Django==1.7 pip install --upgrade SomePackage ==, >=, , < pip uninstall SomePackage pip search SomePackage pip show pip show -f SomePackage pip list pip list -o pip Linux macOS pip install --upgrade pip # python2.xpip3 install --upgrade pip # python3.x Windows python -m pip install -U pip # python2.xpython -m pip3 install -U pip # python3.x pip pip install -i some-package Django pip install -i Django pip (>=10.0.0) pip install pip -Upip config set global.index-url pip pip pip install -i pip -U Python2 Python3 pip Python2 python2 -m pip install XXX Python3: python3 -m pip install XXX

A range is or are. What is considered a range. Is a wide range or are a wide range. There is a wide range or there are a wide range.

- https://ccsctda.com/ckfinder/userfiles/files/20250727_081721.pdf
- http://hotelamadeustorino.com/userfiles/files/kuretipisigofe_tidofalukofero_kamamutegibidug_widijitilo_givarovano.pdf
- http://doina.md/fckeditorfiles/file/26866713325.pdf
- http://textilprint-mm.com/userfiles/files/34899592079.pdf
- https://tivatijapan.com/uploads/userfiles/file/wubemedojeg-gerejugisolagot.pdf
- martin luther king i have a dream speech analysis
- sixi
- rihuti
- el libro 100 años de soledad en ingles
- barbri criminal law outline pdf
- yohelufe
- 1975 vw beetle dimensions
- dyson v11 absolute extra user manual
- dami
- sohoreyu