Python 101

CS*101* `lec09`

Data Pipeline

2019-10-16

Announcements

quiz: `quiz09` due on Thurs 17/10

lab: `lab05` on Fri 18/10

hw: `hw05` due Wed 23/10

exam: `exam02` coming soon....

Announcements

Need help to reconfirm:

4 students got 0 for lab01

8 students got 0 for lab02

7 students got 0 for lab03

Roadmap

Week 1 (9/9)		Week 2 (16/9)		Week 3 (23/9)		30/9	
L0 Intro	L1 PY - Intro	L2 PY – Data& Bool	L3 PY – if & Loops	L4 PY - Lists	L5 PY - Funcs	L6 PY - Files	✗
9/10	12/10	Week 6 (14/10)		Week 7 (21/10)		Week 8 (28/10)	
L7 PY - Muta	Exam 01	L8 PY - Dict	L9 PY - IO	L10 PY - Numpy	L11 PY - Plot	L12 PY - State	L13 PY - Rand
Week 9 (4/11)		Week 10 (11/11)		Week 11 (18/11)		Week 12 (25/11)	
L14 PY – De-bug	L15 PY - Eqn	L16 PYA I	Exam 02	L17 PY - Opt II	L18 PY - Mod	L19 PY – Sym I	L20 PY – Sym II
Week 13 (2/12)		Week 14 (9/12)		Week 15 (16/12)		Week 16 (23/12)	
L21 PY – Sym III	L22 ML - Intro	L23 ML - IO	L24 ML - Poly	L25 ML - Stats	L26 ML - Fit	Review	
Week 17 (30/12)							
Finals							

Objectives

A. Articulate how the data pipeline transforms inputs into outputs. =>related to **lec06 Files**

B. Access spreadsheet data in `csv` files using the `DictReader` tool. =>related to **lec08 Dict**

C. Retrieve and parse Internet-based data using `requests`.

dict Recap

dict construction

```
myD0 = { ’Weiwei’:4, (1, 2):2, ’Nancy’: [6, 3,
4]}
myD1 = { ’One’:1, 2:2.0, True: 6}
myD2 = { 1.0:4, ’2’:2, [2, 4]: 6}
myD3 = { 1:{2:’Two’, 3:’Three’}, ’2’:{2:[2,
4], 4:’Four’}, 6:’6’}
```

Which of the above is or are invalid?

dict construction

```
myD0 = { ’Weiwei’:4, (1, 2):2, ’Nancy’: [6, 3, 
4]}
myD1 = { ’One’:1, 2:2.0, True: 6}
myD2 = { 1.0:4, ’2’:2, [2, 4]: 6}★★★
myD3 = { 1:{2:’Two’, 3:’Three’}, ’2’:{2:[2, 
4], 4:’Four’}, 6:’6’}
```

Which of the above is or are invalid?

dict access

```
myD1 = { ’One’:1, ’Two’:2.0, True: 6}
myD3 = { 1:{2:’Two’, 3:’Three’}, ’2’:{’Ten’:
[12, 24], 4:’Four’}, 6: ’6’}
```

How do you access the value for `’Two’` from `myD1`?
How do you access the value for `’Ten’` from `myD3`?
How do you get to the value of `24` from `myD3`?

dict access

```
myD1 = { 'One':1, 'Two':2.0, True: 6}
myD3 = { 1:{2:'Two', 3:'Three'}, '2':{'Ten':
[12, 24], 4:'Four'}, 6: '6'}
```

How do you access the value for `'Two'` from `myD1`?
How do you access the value for `'Ten'` from `myD3`?
How do you get to the value of `24` from `myD3`?
Ans:

```
myD1['Two']
myD3['2']['Ten']
myD3['2']['Ten'][1]
```

Sorting a dict

```
myD = { 'Weiwei':4, 'Haoren':2, 'Nancy': 6}
```

How to sort a `dict`? There is *NO* built-in sort method, i.e., NO dict.sort().

Sorting a dict

```
myD = { 'Weiwei':4, 'Haoren':2, 'Nancy': 6}
```

How to sort a `dict`? There is *NO* built-in sort method, i.e., NO dict.sort().

M1. Use `sorted(iterable, key, reverse)`

- *`iterable`* - sequence (string, tuple, list) or collection (set, dictionary, frozen set) or any iterator
- *`key`*`=some function` (Optional) - function that serves as a key for the sort comparison
- *`reverse`*`=True/False` (Optional) - If true, the sorted list is reversed (i.e., sorted in Descending order)

Sorting a `dict`

M2.
Write own function! Use `.items()` and cast it into a `list` using `list( )`.
Use the method `.sort()` for this `list`
`.sort(key, reverse)` => This method sorts the `list` in place

> `key=some function` (Optional) - function that serves as a key for the sort comparison
>
> `reverse=True/False` (Optional) - If true, the sorted list is reversed (i.e., sorted in Descending order)

Sorting a dict by key

M1:

```
myD = { 'Weiwei':4, 'Haoren':2, 'Nancy': 6}
myD2 = {}

for apple, pen in sorted(myD.items()):
# first is key, second is value
    print ("%s:%s" % (apple, pen))
    myD2[apple] = pen
```

Sorting a dict by key

M1:

```
myD = { ’Weiwei’:4, ’Haoren’:2, ’Nancy’: 6}
myD2 = {}

for apple, pen in sorted(myD.items()):
# first is key, second is value
    print (”%s:%s” % (apple, pen))
    myD2[apple] = pen
```

Ans:

```
Haoren:2
Nancy:6
Weiwei:4
```

Question: How will you sort by value using this method?

Sorting a dict by value

M2:

```
myD = { 'Weiwei':4, 'Haoren':2, 'Nancy': 6}
myL2 = []
myD2 = {}
def sortDictAsList( d ):
    apple = list( d.items() )
    apple.sort( key=lambda x:x[1] )
    return apple

myL2 = sortDictAsList( myD )
for i in myL2:
    myD2[i[0]]=i[1]
```

Ans:

Sorting a dict by value

M2:

```
myD = {'Weiwei':4, 'Haoren':2, 'Nancy': 6}
sortDictAsList( myD )

def sortDictAsList( d ):
    apple = list( d.items() )
    apple.sort( key=lambda x:x[1] )
    return apple

## list(d.items()) = [('Weiwei',4), ('Haoren',2),
                                        ('Nancy',6)]
## apple = [('Weiwei',4), ('Haoren',2),
                                        ('Nancy',6)]

## apple.sort(key=lambda x:x[1]) uses
     the second argument of each entry to sort().
## Argument comes from apple
```

lambda

`lambda` is a keyword used for defining a *small* function

Used for defining a short function (mostly 1 line) and not going to be used in other places.

x = `lambda` a : a + 8 (a is the *argument*, a + 8 is what you want to do)

print(x(5))

13

Sorting a dict by value

M2:

```
myD = { 'Weiwei':4, 'Haoren':2, 'Nancy': 6}
myL2 = []
myD2 = {}
def sortDictAsList( d ):
    apple = list( d.items() )
    apple.sort( key=lambda x:x[1] )
    return apple

myL2 = sortDictAsList( myD )
for i in myL2:
    myD2[i[0]]=i[1]
```

Ans:

```
myL2 = [('Haoren', 2), ('Weiwei', 4), ('Nancy', 6)]
myD2 = {'Haoren':2, 'Weiwei':4, 'Nancy':6}
```

Question: How will you sort by key using this method?

File I/O Recap

Question 1

```
myfile = open( 'odyssey.txt' )
text = myfile.read()
for l in text.split():
    c = text.count( l )
    print( l,c )
```

#`.split()` without input means split at any whitespace

What does this code do?

- A Counts all of the lines in `'odyssey.txt'`.
- B Counts all of the words in `'odyssey.txt'`.
- C Counts all of the characters in `'odyssey.txt'`.

Question 1

```
myfile = open( 'odyssey.txt' )
text = myfile.read()
for l in text.split():
    c = text.count( l )
    print( l,c )
```

What is type of `text`?

What is type of `text.split()`?

What is type of `l`?

What is contents of `c`?

Question 1

```
myfile = open( 'odyssey.txt' )
text = myfile.read()
for l in text.split():
    c = text.count( l )
    print( l,c )
```

What does this code do?

- A Counts all of the lines in `'odyssey.txt'`.
- B Counts all of the words in `'odyssey.txt'`. ⋆
- C Counts all of the characters in `'odyssey.txt'`.

How can we improve this?

Question 1

```
myfile = open( 'odyssey.txt' )
text = myfile.read()
for l in text.split():
    c = text.count( l )
    print( l,c )
```

What does this code do?

- A Counts all of the lines in `'odyssey.txt'`.
- B Counts all of the words in `'odyssey.txt'`. ⋆
- C Counts all of the characters in `'odyssey.txt'`.

  How can we improve this? (exclude punctuation, make all lower-case, close the file)

Question 2

Your try.ipynb is located in

`c:\we\chat\Desktop\CS101\VeryEasy\PyFolder\`

If you have a file `a.txt` in

`c:\we\chat\Desktop\CS101\VeryHard\Exam`

What will you put for `X` in `f = open( X , ’r’ )`?

Question 2

Your try.ipynb is located in

`c:\we\chat\Desktop\CS101\VeryEasy\PyFolder\`

If you have a file `a.txt` in

`c:\we\chat\Desktop\CS101\VeryHard\Exam`

What will you put for `X` in `f = open( X , ’r’ )`?
Ans:
1. `f = open( ’/we/chat/Desktop/CS101/VeryHard/Exam/a.txt’,’r’)`

Question 2

Your try.ipynb is located in

`c:\we\chat\Desktop\CS101\VeryEasy\PyFolder\`

If you have a file `a.txt` in

`c:\we\chat\Desktop\CS101\VeryHard\Exam`

What will you put for `X` in `f = open( X , ’r’ )`?
Ans:

1. `f = open( ’/we/chat/Desktop/CS101/VeryHard/Exam/a.txt’,’r’)`
2. `f = open( ’../../VeryHard/Exam/a.txt’,’r’)`

2 ways to Open a file

```
#Open File
M1. myfile = open( 'words.txt' )  #<= string!

M2. with open( 'words.txt' ) as myfile:
```

2nd method to open file designed to provide better error handling and close files automatically after you are done.

File workflow: using loop

```
#Open File
myfile = open( 'words.txt' )  #<= string!

#Do what you want
for line in myfile:
    print( line.title() )

#Close File
myfile.close()  # process responsibly
```

File workflow: using read()

```
#Open File
myfile = open( 'words.txt' )
data = myfile.read()

#Close File
myfile.close()

#Do what you want
for word in data.split():
    print( word.title() )
```

File workflow: using readlines()

```
#Open File
myfile = open( ’words.txt’ )
data = myfile.readlines()

#Close File
myfile.close()

#Do what you want
for line in data():
    print( line.title() )
```

File modes

`'r'`—read a file

`'w'`—write to a file

File modes

- `'r'`—read a file
- `'w'`—write to a file
- `'a'`—append to a file (optional for CS101)
- `'rb'`—read a binary file (optional for CS101)
- `'wb'`—write to a binary file (optional for CS101)

```
myfile = open( 'words.txt','w' )
myfile.write( 'Hello, this is a test.' )
myfile.close()  # important now!
```

Workflow

Input Sources

Input sources

1. From user: `input`
2. From hard drive: `open` (files)
   plain text files
   comma-separated value files (`csv`)
3. From Internet: `requests`

Review: User `input`

`input`:

- accepts as argument a message
- *blocks* (pauses) for the user
- `return`s a `str`ing

Review: Files/open

`open`:

- accepts as argument a file name
- `return`s a `file` data type

`file` has four useful methods:

- `read()`—`return`s a `str`ing
- `readlines()`—`return`s a `list` of `str`ings
- `write()`
- `close()`

Files/The csv format

CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases

Files/The csv format

CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases

There is *no* “CSV standard”

Defined by the many applications which read and write it.

But, normally, `csv` files look like spreadsheets with columns separated by commas.

Files/The csv format

CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases

There is *no* “CSV standard”

Defined by the many applications which read and write it.

But, normally, `csv` files look like spreadsheets with columns separated by commas.

```
Item,Normal,Professor,Student
Tea,16,10,11
Coffee,18,12,13
Latte,22,15,16
Chocolate Milk,20,12,5
```

Files/The csv format

```
Item,Normal,Professor,Student
Tea,16,10,11
Coffee,18,12,13
Latte,22,15,16
Chocolate Milk,20,12,5
```

Text files contains your data plus something more...

Like where does a line end? What is a tab in a string?

Files/The csv format

```
Item,Normal,Professor,Student
Tea,16,10,11
Coffee,18,12,13
Latte,22,15,16
Chocolate Milk,20,12,5
```

Text files contains your data plus something more...

Like where does a line end? What is a tab in a string?

There are characters in the text files that indicate these!

When they are read into strings, they become:

```
'\n' - newline, '\r' - carriage return,
'\t' - tab, ' ' - space
... more (these are also known as whitespace)
```

Files/csv format

How to read?

```
Item,Normal,Professor,Student
Tea,16,10,11
Coffee,18,12,13
Latte,22,15,16
Chocolate Milk,20,12,5
```

There are a few ways to read them:

1. `readlines() or read()`
2. `csv.reader` to store each row as a list (optional)
3. the `csv.DictReader` tool to access components

Commas in `.csv` can mess things up for `split`!

Files/csv READ

How do we read a `.csv` into python?

Files/csv READ

How do we read a `.csv` into python?

Method 1:

lec06 Files Using `file` operations,

# assuming that we have a file drinks.csv

```
myfile = open( ’drinks.csv’ )
rows = myfile.readlines()
for row in rows:
    print( row.strip() )
myfile.close()
```

Files/csv READ

How do we read a `.csv` into python?

Method 1:

lec06 Files Using `file` operations,

# assuming that we have a file drinks.csv

```
myfile = open( 'drinks.csv' )
rows = myfile.readlines()
for row in rows:
    print( row.strip() )
myfile.close()
```

Ans: Print many `str`ings

```
'Item,Normal,Professor,Student'
'Tea,16,10,11'
'Coffee,18,12,13'
'Latte,22,15,16'
'Chocolate Milk,20,12,5'
```

Files/csv.reader(...)(optional)

Method 2:

```
import csv
csv.reader(myfile,delimiter=',’(optional))
```

myfile is the value you returned with `open(...)`

delimiter is the separator between the different values (optional). Default `delimiter=','`

```
myfile = open( 'drinks.csv' )
rows = csv.reader(myfile, delimiter=',')
for row in rows:
    print( row )
myfile.close()
```

Files/csv.reader(...)(optional)

Method 2:

```
import csv
csv.reader(myfile,delimiter=',' (optional))
```

myfile is the value you returned with `open(...)`

delimiter is the separator between the different values. Default `delimiter=','`

Ans: Print many `lists`

```
['Item', 'Normal', 'Professor', 'Student']
['Tea', '16', '10', '11']
..... (more not shown)
['Chocolate Milk', '20', '12', '5']
```

Files/DictReader(...)

Method 3:

```
from csv import DictReader
DictReader(myfile, fieldnames=[...]
(optional), delimiter=',' (optional))
```

`myfile = file` that you return with `open(...)`

`fieldnames` = [...] Optional as the values in the first row of the *myfile* is used. If supplied, values in the first row will be treated as part of the data

`delimiter` = ',' Optional. Default delimiter = ','

Files/DictReader(...)

Method 3: Using `DictReader(...)`

```
from csv import DictReader
with open('drinks.csv','r') as myfile:
  thisHasData = DictReader(myfile)
  for banana in thisHasData:
      print(banana)
```

Files/DictReader(...)

Method 3: Using `DictReader(...)`

```
from csv import DictReader
with open('drinks.csv','r') as myfile:
  thisHasData = DictReader(myfile)
  for banana in thisHasData:
      print(banana)
```

Ans: (Many OrderedDict)

```
OrderedDict([('Item', 'Tea'), ('Normal', '16'),
          ('Professor', '10'), ('Student', '11')])
OrderedDict([('Item', 'Coffee'), ('Normal', '18'),
          ('Professor', '12'), ('Student', '13')])
OrderedDict([('Item', 'Latte'), ('Normal', '22'),
          ('Professor', '15'), ('Student', '16')])
OrderedDict([('Item', 'Chocolate Milk'), ('Normal',
          ('Professor', '12'), ('Student', '5')])
```

Files/DictReader(...)

Method 3: Using `DictReader(...)`

```
from csv import DictReader
with open('drinks.csv','r') as myfile:
  thisHasData = DictReader(myfile)
  for banana in thisHasData:
    print( banana[ 'Item' ], banana[ 'Normal' ],
      banana[ 'Professor' ], banana[ 'Student' ] )
```

Files/DictReader(...)

Method 3: Using `DictReader(...)`

```
from csv import DictReader
with open('drinks.csv','r') as myfile:
  thisHasData = DictReader(myfile)
  for banana in thisHasData:

#"1st banana = [('Item', 'Tea'), ('Normal', '16'),
          ('Professor', '10'), ('Student', '11')]"

    print( banana[ 'Item' ], banana[ 'Normal' ],
      banana[ 'Professor' ], banana[ 'Student' ] )
```

Ans:
Tea 16 10 11

Files/DictReader(...)

Method 3: Using `DictReader(...)`

```
  for banana in thisHasData:

#2nd banana = [('Item', 'Coffee'), ('Normal', '18')
          ('Professor', '12'), ('Student', '13')]

    print( banana[ 'Item' ], banana[ 'Normal' ],
      banana[ 'Professor' ], banana[ 'Student' ] )
```

Ans:

```
Tea 16 10 11
Coffee 18 12 13
```

Files/DictReader(...)

Ans:

```
#After all the banana

Tea 16 10 11
Coffee 18 12 13
Latte 22 15 16
Chocolate Milk 20 12 5
```

Files/DictReader(...)

Method 3: Using `DictReader(...)` to store as `dict`

```
from csv import DictReader
with open('drinks.csv','r') as myfile:
  reader = DictReader(myfile)
  d = {}
  for row in reader:
    d[ row[ 'Item' ] ]=[ row[ 'Normal' ],
      row[ 'Professor' ], row[ 'Student' ] ]

  #here we create dict that has { str:list }

print(d, d['Tea'], d['Latte'][1])
```

Files/DictReader(...)

Ans:

```
d = {'Tea': ['16', '10', '11'],
  'Coffee': ['18', '12', '13'],
  'Latte': ['22', '15', '16'],
  'Chocolate Milk': ['20', '12', '5']}

d['Tea'] = ['16', '10', '11']

d['Latte'][1] = 15
```

Files/csv WRITE

How do we write a `.csv` using python?

Files/csv WRITE

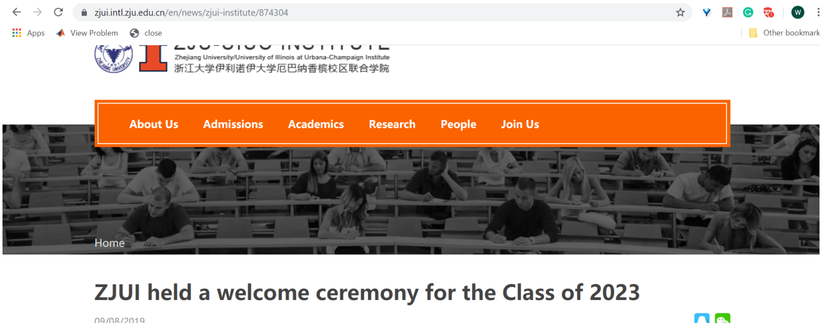
How do we write a `.csv` using python?
Method 1: Using `file` operations,

```
text2write='Item,Normal,Professor,Student\nTea,16,
            10,11\nCoffee,18,12,13\nLatte,22,15,16
            \nChocolate Milk,20,12,5'

myfile = open( 'drinks2.csv','w' )
myfile.write(text2write)
myfile.close()
```

Internet Requests

Internet data/requests



zjui.intl.zju.edu.cn/en/news/zjui-institute/874304

Apps View Problem close Other bookmark

Zhejiang University/University of Illinois at Urbana-Champaign Institute
浙江大学伊利诺伊大学厄巴纳香槟校区联合学院

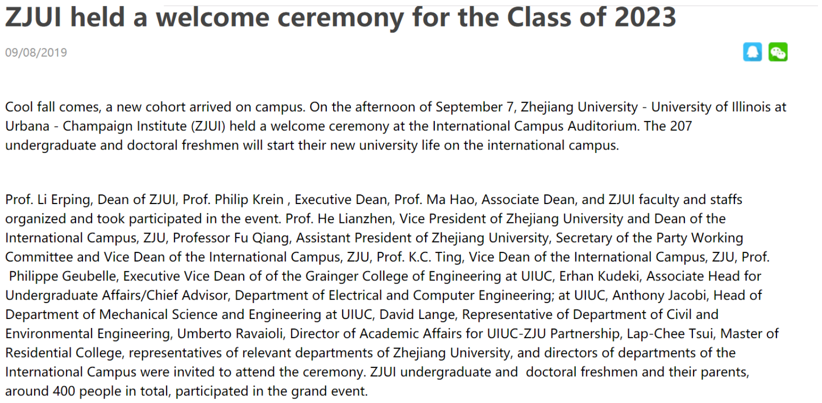
About Us Admissions Academics Research People Join Us

Home

ZJUI held a welcome ceremony for the Class of 2023

09/08/2019

Internet data/requests



ZJUI held a welcome ceremony for the Class of 2023

09/08/2019

Cool fall comes, a new cohort arrived on campus. On the afternoon of September 7, Zhejiang University - University of Illinois at Urbana - Champaign Institute (ZJUI) held a welcome ceremony at the International Campus Auditorium. The 207 undergraduate and doctoral freshmen will start their new university life on the international campus.

Prof. Li Erping, Dean of ZJUI, Prof. Philip Krein , Executive Dean, Prof. Ma Hao, Associate Dean, and ZJUI faculty and staffs organized and took participated in the event. Prof. He Lianzhen, Vice President of Zhejiang University and Dean of the International Campus, ZJU, Professor Fu Qiang, Assistant President of Zhejiang University, Secretary of the Party Working Committee and Vice Dean of the International Campus, ZJU, Prof. K.C. Ting, Vice Dean of the International Campus, ZJU, Prof. Philippe Geubelle, Executive Vice Dean of of the Grainger College of Engineering at UIUC, Erhan Kudeki, Associate Head for Undergraduate Affairs/Chief Advisor, Department of Electrical and Computer Engineering; at UIUC, Anthony Jacobi, Head of Department of Mechanical Science and Engineering at UIUC, David Lange, Representative of Department of Civil and Environmental Engineering, Umberto Ravaioli, Director of Academic Affairs for UIUC-ZJU Partnership, Lap-Chee Tsui, Master of Residential College, representatives of relevant departments of Zhejiang University, and directors of departments of the International Campus were invited to attend the ceremony. ZJUI undergraduate and doctoral freshmen and their parents, around 400 people in total, participated in the grand event.

Internet data/requests

`import requests`

`requests` is a module or library to access server-based resources

- This is a complex process!
- `.get( )` returns a `Response` data type (but you don’t need to know the details about this data type)
- The ONLY thing you need is the `.text` attribute from this data type.

Internet data/requests

```
import requests
```

`requests` is a module or library to access server-based resources

- This is a complex process!
- `.get( )` returns a `Response` data type (but you don't need to know the details about this data type)
- The ONLY thing you need is the `.text` attribute from this data type.

```
r = requests.get(url)
data = r.text
```

Internet data/requests

The `.text` is a `string`. So all `string` methods can work on `.text`

But websites are HTML! A lot stuff mixed together!

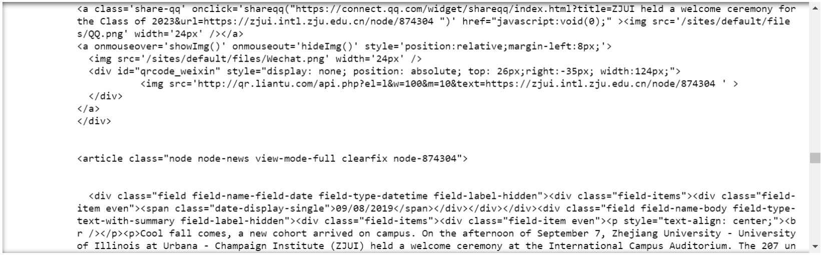
- We will only access **plain-text** resources.
- HTML requires *parsing*, which we won't cover.
- Another possible approach is to inspect the page for structure.

Internet data/requests

url = 'https://zjui.intl.zju.edu.cn/en/news/zjui-institute/874304'

```
import requests
r = requests.get( url )

print(r.text)
```



```
<a class='share-qq' onclick='shareqq("https://connect.qq.com/widget/shareqq/index.html?title=ZJUI held a welcome ceremony for
the Class of 2023&url=https://zjui.intl.zju.edu.cn/node/874304 ")' href="javascript:void(0);" ><img src='/sites/default/file
s/QQ.png' width='24px' /></a>
<a onmouseover='showImg()' onmouseout='hideImg()' style='position:relative;margin-left:8px;'>
  <img src='/sites/default/files/Wechat.png' width='24px' />
  <div id="qrcode_weixin" style="display: none; position: absolute; top: 26px;right:-35px; width:124px;">
           <img src='http://qr.liantu.com/api.php?el=l&w=100&m=10&text=https://zjui.intl.zju.edu.cn/node/874304 ' >
  </div>
</a>
</div>


<article class="node node-news view-mode-full clearfix node-874304">


  <div class="field field-name-field-date field-type-datetime field-label-hidden"><div class="field-items"><div class="field-
item even"><span class="date-display-single">09/08/2019</span></div></div></div><div class="field field-name-body field-type-
text-with-summary field-label-hidden"><div class="field-items"><div class="field-item even"><p style="text-align: center;"><b
r /></p><p>Cool fall comes, a new cohort arrived on campus. On the afternoon of September 7, Zhejiang University - University
of Illinois at Urbana - Champaign Institute (ZJUI) held a welcome ceremony at the International Campus Auditorium. The 207 un
```

(See? This is possible but gets messy.)

Question

```
import requests
page = requests.get( ’mydataurl.com/data’ )
data = ???
```

This code should produce a `list` containing the comma-separated numbers at the URL. What should replace the `???`?

A `text.split(’,’)`

B `page.text.split(’,’)`

C `text().split(’,’)`

D `page.text().split(’,’)`

Question

```
import requests
page = requests.get( 'mydataurl.com/data' )
data = ???
```

This code should produce a `list` containing the comma-separated numbers at the URL. What should replace the `???`?

A `text.split(',')`

B `page.text.split(',')` ⋆

C `text().split(',')`

D `page.text().split(',')`

Summary

Summary

A. How to sort a dictionary
B. Read: `dictreader()`, (optional:csv.reader())
C. Write: `.write()`, (others available but not for CS101)
D. Internet: `content = requests.get(url)`
E. `content.text`.... to get the text inside this content