Metrics_of_successful_sites_and_companies

April 14, 2017

In [1]:
```
#First we import the libraries we will need
import urllib
import urllib2
import time
import os
from bs4 import BeautifulSoup
import re
import numpy as np
import pandas as pd
```

In [2]:
```
#First of all we need to find all the name of the sites that belong to fort
#The information needed from the below link
url = "http://www.zyxware.com/articles/4344/list-of-fortune-500-companies-a
list_company_number =[]
list_company_name = []
list_company_website = []
```

In [3]:
```
#In order to extract the needed informations we will create 3 lists. The fi
#second one will contain the name of the company and the 3rd one will conta
#For achieving this purpose we will create a funstion that will in its turn
#In order to know if the function worked we will ask it to return the first
def websites (url):
    from time import time # I used it to see how much time it does to run t
    start = time ()
    browser = urllib2.build_opener()
    browser.addheaders = [('User-agent', 'Mozilla/5.0')]
    response = browser.open(url)# this might throw an exception if somethin
    myHTML = response.read()
    soup = BeautifulSoup(myHTML,"lxml")
    o = 0
    td_list =[]
    for row2 in soup.html.body.findAll('td'):
        td_list.insert(o, row2)
        o = o + 1
    a = 0
    b = 1
    c = 2
```

```
            list_numbering = 0
            for i in range (0,500):
                num = str(td_list[a])
                company = str(td_list[b])
                site = str(td_list[c])
                c_num = re.findall('>(.+?)</td>',num)
                c_num = str(c_num[0])
                c_name = re.findall('>(.+?)</td>',company)
                c_name = str(c_name[0])
                c_site = re.findall('"">(.+?)</a>',site)
                c_site = str(c_site[0])
                list_company_number.insert(list_numbering,c_num)
                list_company_name.insert(list_numbering,c_name)
                list_company_website.insert(list_numbering,c_site)
                a = a + 3
                b = b + 3
                c = c + 3
                list_numbering =  list_numbering + 1
            end = time ()
            duration = round (end - start, 1)
            minutes = round (duration /60, 1)
            print 'The lists are ready in ', duration, ' seconds'
            print 'The lists are ready in ', minutes, ' minutes'
```

```
In [4]: # After creating the function we should now test that it actually works co
        websites (url)
```

```
The lists are ready in  1.7  seconds
The lists are ready in  0.0  minutes
```

```
In [5]: #Try to validate each page url #pip install validators
        import validators
        nv = 0
        for num in range(len(list_company_website)):
            line = 'http://' + str(list_company_website[num])
            x = validators.url(line)
            if x != True:
                nv = nv +1
        print "The validation is complete! There were" , nv, "not valid pages"
```

```
The validation is complete! There were 0 not valid pages
```

```
In [6]: list500_sites = []
        list500_names = []
        list500_num = []
        list500_url = []
```

```
In [7]: #def list_company_HTML (list_company_website,list_company_name,start,end):
        import time
        browser2 = urllib2.build_opener()
        browser2.addheaders = [('User-agent', 'Mozilla/5.0')]
        for i in range (0,500):
            k = str(i + 1)
            lc = str(list_company_website[i])
            lc = lc.replace("'","")
            lc = lc.replace("[","")
            lc = lc.replace("]","")
            lcn = str(list_company_name[i])
            lcn = lcn.replace("'","")
            lcn = lcn.replace("[","")
            lcn = lcn.replace("]","")
            url2= 'http://' + lc
            list500_names.insert(i,lcn)
            list500_url.insert(i,lc)
            list500_num.insert(i,k)
            if i == 118 or i == 464 or i == 268 or i == 70:
                #These sites have a problem and the whole code is stacking
                #when I run it so we will thing of this site as a not downloadable
                list500_sites.insert(i,0)
                print ("The site " + str(i) + " has NOT been downloaded!")
            else:
                #an exception might be thrown, so the code should be in a try-excep
                try:
                    response2=browser2.open(url2)
                    print ("The site " + str(i) + " has been downloaded!")
                except Exception: # this describes what to do if an exception is th
                    list500_sites.insert(i,0)
                    print ("The site " + str(i) + " has NOT been downloaded from ex
                    continue
                    #if it goes into to exception it does not continue below
                myHTML2=response2.read()
                list500_sites.insert(i,myHTML2)
                #wait for 2 seconds
                time.sleep(2)

The site 0 has been downloaded!
The site 1 has been downloaded!
The site 2 has been downloaded!
The site 3 has been downloaded!
The site 4 has been downloaded!
The site 5 has been downloaded!
The site 6 has been downloaded!
The site 7 has been downloaded!
The site 8 has been downloaded!
The site 9 has been downloaded!
```

```
The site 10 has been downloaded!
The site 11 has been downloaded!
The site 12 has been downloaded!
The site 13 has been downloaded!
The site 14 has been downloaded!
The site 15 has NOT been downloaded from exception!
The site 16 has been downloaded!
The site 17 has been downloaded!
The site 18 has been downloaded!
The site 19 has been downloaded!
The site 20 has been downloaded!
The site 21 has been downloaded!
The site 22 has been downloaded!
The site 23 has been downloaded!
The site 24 has been downloaded!
The site 25 has been downloaded!
The site 26 has been downloaded!
The site 27 has been downloaded!
The site 28 has been downloaded!
The site 29 has been downloaded!
The site 30 has been downloaded!
The site 31 has been downloaded!
The site 32 has been downloaded!
The site 33 has been downloaded!
The site 34 has been downloaded!
The site 35 has been downloaded!
The site 36 has been downloaded!
The site 37 has been downloaded!
The site 38 has been downloaded!
The site 39 has been downloaded!
The site 40 has been downloaded!
The site 41 has been downloaded!
The site 42 has been downloaded!
The site 43 has been downloaded!
The site 44 has been downloaded!
The site 45 has been downloaded!
The site 46 has been downloaded!
The site 47 has been downloaded!
The site 48 has been downloaded!
The site 49 has been downloaded!
The site 50 has been downloaded!
The site 51 has NOT been downloaded from exception!
The site 52 has been downloaded!
The site 53 has been downloaded!
The site 54 has been downloaded!
The site 55 has been downloaded!
The site 56 has been downloaded!
The site 57 has been downloaded!
```

```
The site 58 has been downloaded!
The site 59 has been downloaded!
The site 60 has been downloaded!
The site 61 has been downloaded!
The site 62 has NOT been downloaded from exception!
The site 63 has been downloaded!
The site 64 has been downloaded!
The site 65 has been downloaded!
The site 66 has been downloaded!
The site 67 has been downloaded!
The site 68 has been downloaded!
The site 69 has been downloaded!
The site 70 has NOT been downloaded!
The site 71 has been downloaded!
The site 72 has been downloaded!
The site 73 has been downloaded!
The site 74 has been downloaded!
The site 75 has been downloaded!
The site 76 has been downloaded!
The site 77 has been downloaded!
The site 78 has been downloaded!
The site 79 has been downloaded!
The site 80 has been downloaded!
The site 81 has been downloaded!
The site 82 has been downloaded!
The site 83 has been downloaded!
The site 84 has been downloaded!
The site 85 has been downloaded!
The site 86 has been downloaded!
The site 87 has been downloaded!
The site 88 has been downloaded!
The site 89 has been downloaded!
The site 90 has NOT been downloaded from exception!
The site 91 has been downloaded!
The site 92 has been downloaded!
The site 93 has been downloaded!
The site 94 has been downloaded!
The site 95 has been downloaded!
The site 96 has been downloaded!
The site 97 has NOT been downloaded from exception!
The site 98 has been downloaded!
The site 99 has been downloaded!
The site 100 has been downloaded!
The site 101 has been downloaded!
The site 102 has been downloaded!
The site 103 has been downloaded!
The site 104 has been downloaded!
The site 105 has been downloaded!
```

```
The site 106 has been downloaded!
The site 107 has been downloaded!
The site 108 has been downloaded!
The site 109 has been downloaded!
The site 110 has been downloaded!
The site 111 has been downloaded!
The site 112 has been downloaded!
The site 113 has been downloaded!
The site 114 has been downloaded!
The site 115 has been downloaded!
The site 116 has been downloaded!
The site 117 has been downloaded!
The site 118 has NOT been downloaded!
The site 119 has been downloaded!
The site 120 has been downloaded!
The site 121 has been downloaded!
The site 122 has been downloaded!
The site 123 has been downloaded!
The site 124 has been downloaded!
The site 125 has been downloaded!
The site 126 has been downloaded!
The site 127 has been downloaded!
The site 128 has been downloaded!
The site 129 has been downloaded!
The site 130 has been downloaded!
The site 131 has been downloaded!
The site 132 has been downloaded!
The site 133 has been downloaded!
The site 134 has been downloaded!
The site 135 has NOT been downloaded from exception!
The site 136 has been downloaded!
The site 137 has been downloaded!
The site 138 has been downloaded!
The site 139 has been downloaded!
The site 140 has been downloaded!
The site 141 has NOT been downloaded from exception!
The site 142 has been downloaded!
The site 143 has been downloaded!
The site 144 has NOT been downloaded from exception!
The site 145 has been downloaded!
The site 146 has been downloaded!
The site 147 has been downloaded!
The site 148 has been downloaded!
The site 149 has been downloaded!
The site 150 has been downloaded!
The site 151 has been downloaded!
The site 152 has been downloaded!
The site 153 has been downloaded!
```

```
The site 154 has been downloaded!
The site 155 has been downloaded!
The site 156 has been downloaded!
The site 157 has been downloaded!
The site 158 has been downloaded!
The site 159 has been downloaded!
The site 160 has been downloaded!
The site 161 has been downloaded!
The site 162 has been downloaded!
The site 163 has been downloaded!
The site 164 has NOT been downloaded from exception!
The site 165 has been downloaded!
The site 166 has been downloaded!
The site 167 has been downloaded!
The site 168 has been downloaded!
The site 169 has been downloaded!
The site 170 has been downloaded!
The site 171 has been downloaded!
The site 172 has been downloaded!
The site 173 has been downloaded!
The site 174 has been downloaded!
The site 175 has been downloaded!
The site 176 has been downloaded!
The site 177 has been downloaded!
The site 178 has been downloaded!
The site 179 has been downloaded!
The site 180 has been downloaded!
The site 181 has been downloaded!
The site 182 has been downloaded!
The site 183 has been downloaded!
The site 184 has been downloaded!
The site 185 has been downloaded!
The site 186 has been downloaded!
The site 187 has been downloaded!
The site 188 has been downloaded!
The site 189 has been downloaded!
The site 190 has been downloaded!
The site 191 has been downloaded!
The site 192 has been downloaded!
The site 193 has been downloaded!
The site 194 has been downloaded!
The site 195 has NOT been downloaded from exception!
The site 196 has been downloaded!
The site 197 has been downloaded!
The site 198 has been downloaded!
The site 199 has been downloaded!
The site 200 has been downloaded!
The site 201 has been downloaded!
```

```
The site 202 has been downloaded!
The site 203 has been downloaded!
The site 204 has been downloaded!
The site 205 has been downloaded!
The site 206 has been downloaded!
The site 207 has been downloaded!
The site 208 has been downloaded!
The site 209 has been downloaded!
The site 210 has been downloaded!
The site 211 has been downloaded!
The site 212 has been downloaded!
The site 213 has been downloaded!
The site 214 has been downloaded!
The site 215 has been downloaded!
The site 216 has NOT been downloaded from exception!
The site 217 has been downloaded!
The site 218 has been downloaded!
The site 219 has been downloaded!
The site 220 has been downloaded!
The site 221 has been downloaded!
The site 222 has been downloaded!
The site 223 has been downloaded!
The site 224 has been downloaded!
The site 225 has been downloaded!
The site 226 has been downloaded!
The site 227 has been downloaded!
The site 228 has NOT been downloaded from exception!
The site 229 has been downloaded!
The site 230 has been downloaded!
The site 231 has been downloaded!
The site 232 has been downloaded!
The site 233 has been downloaded!
The site 234 has been downloaded!
The site 235 has been downloaded!
The site 236 has been downloaded!
The site 237 has been downloaded!
The site 238 has been downloaded!
The site 239 has been downloaded!
The site 240 has been downloaded!
The site 241 has been downloaded!
The site 242 has NOT been downloaded from exception!
The site 243 has been downloaded!
The site 244 has NOT been downloaded from exception!
The site 245 has been downloaded!
The site 246 has been downloaded!
The site 247 has been downloaded!
The site 248 has been downloaded!
The site 249 has been downloaded!
```

```
The site 250 has been downloaded!
The site 251 has been downloaded!
The site 252 has been downloaded!
The site 253 has been downloaded!
The site 254 has been downloaded!
The site 255 has been downloaded!
The site 256 has been downloaded!
The site 257 has been downloaded!
The site 258 has been downloaded!
The site 259 has been downloaded!
The site 260 has been downloaded!
The site 261 has been downloaded!
The site 262 has been downloaded!
The site 263 has been downloaded!
The site 264 has been downloaded!
The site 265 has been downloaded!
The site 266 has been downloaded!
The site 267 has been downloaded!
The site 268 has NOT been downloaded!
The site 269 has been downloaded!
The site 270 has been downloaded!
The site 271 has been downloaded!
The site 272 has been downloaded!
The site 273 has been downloaded!
The site 274 has been downloaded!
The site 275 has NOT been downloaded from exception!
The site 276 has been downloaded!
The site 277 has been downloaded!
The site 278 has been downloaded!
The site 279 has been downloaded!
The site 280 has been downloaded!
The site 281 has been downloaded!
The site 282 has NOT been downloaded from exception!
The site 283 has been downloaded!
The site 284 has been downloaded!
The site 285 has been downloaded!
The site 286 has been downloaded!
The site 287 has been downloaded!
The site 288 has been downloaded!
The site 289 has been downloaded!
The site 290 has been downloaded!
The site 291 has been downloaded!
The site 292 has been downloaded!
The site 293 has been downloaded!
The site 294 has been downloaded!
The site 295 has been downloaded!
The site 296 has been downloaded!
The site 297 has been downloaded!
```

```
The site 298 has been downloaded!
The site 299 has been downloaded!
The site 300 has been downloaded!
The site 301 has been downloaded!
The site 302 has been downloaded!
The site 303 has been downloaded!
The site 304 has been downloaded!
The site 305 has been downloaded!
The site 306 has NOT been downloaded from exception!
The site 307 has been downloaded!
The site 308 has been downloaded!
The site 309 has been downloaded!
The site 310 has been downloaded!
The site 311 has NOT been downloaded from exception!
The site 312 has been downloaded!
The site 313 has been downloaded!
The site 314 has been downloaded!
The site 315 has been downloaded!
The site 316 has been downloaded!
The site 317 has been downloaded!
The site 318 has been downloaded!
The site 319 has been downloaded!
The site 320 has been downloaded!
The site 321 has been downloaded!
The site 322 has been downloaded!
The site 323 has been downloaded!
The site 324 has been downloaded!
The site 325 has been downloaded!
The site 326 has NOT been downloaded from exception!
The site 327 has been downloaded!
The site 328 has been downloaded!
The site 329 has been downloaded!
The site 330 has been downloaded!
The site 331 has been downloaded!
The site 332 has been downloaded!
The site 333 has been downloaded!
The site 334 has been downloaded!
The site 335 has been downloaded!
The site 336 has been downloaded!
The site 337 has been downloaded!
The site 338 has been downloaded!
The site 339 has been downloaded!
The site 340 has been downloaded!
The site 341 has been downloaded!
The site 342 has been downloaded!
The site 343 has been downloaded!
The site 344 has been downloaded!
The site 345 has been downloaded!
```

```
The site 346 has been downloaded!
The site 347 has been downloaded!
The site 348 has been downloaded!
The site 349 has been downloaded!
The site 350 has been downloaded!
The site 351 has been downloaded!
The site 352 has been downloaded!
The site 353 has been downloaded!
The site 354 has been downloaded!
The site 355 has been downloaded!
The site 356 has been downloaded!
The site 357 has been downloaded!
The site 358 has been downloaded!
The site 359 has been downloaded!
The site 360 has been downloaded!
The site 361 has been downloaded!
The site 362 has been downloaded!
The site 363 has NOT been downloaded from exception!
The site 364 has been downloaded!
The site 365 has been downloaded!
The site 366 has been downloaded!
The site 367 has been downloaded!
The site 368 has been downloaded!
The site 369 has been downloaded!
The site 370 has been downloaded!
The site 371 has been downloaded!
The site 372 has been downloaded!
The site 373 has been downloaded!
The site 374 has been downloaded!
The site 375 has been downloaded!
The site 376 has been downloaded!
The site 377 has been downloaded!
The site 378 has been downloaded!
The site 379 has been downloaded!
The site 380 has been downloaded!
The site 381 has been downloaded!
The site 382 has been downloaded!
The site 383 has been downloaded!
The site 384 has been downloaded!
The site 385 has been downloaded!
The site 386 has been downloaded!
The site 387 has been downloaded!
The site 388 has been downloaded!
The site 389 has been downloaded!
The site 390 has been downloaded!
The site 391 has been downloaded!
The site 392 has been downloaded!
The site 393 has been downloaded!
```

```
The site 394 has been downloaded!
The site 395 has been downloaded!
The site 396 has been downloaded!
The site 397 has NOT been downloaded from exception!
The site 398 has been downloaded!
The site 399 has been downloaded!
The site 400 has been downloaded!
The site 401 has been downloaded!
The site 402 has been downloaded!
The site 403 has been downloaded!
The site 404 has been downloaded!
The site 405 has been downloaded!
The site 406 has been downloaded!
The site 407 has been downloaded!
The site 408 has been downloaded!
The site 409 has been downloaded!
The site 410 has been downloaded!
The site 411 has been downloaded!
The site 412 has been downloaded!
The site 413 has been downloaded!
The site 414 has NOT been downloaded from exception!
The site 415 has been downloaded!
The site 416 has been downloaded!
The site 417 has been downloaded!
The site 418 has been downloaded!
The site 419 has been downloaded!
The site 420 has been downloaded!
The site 421 has been downloaded!
The site 422 has been downloaded!
The site 423 has been downloaded!
The site 424 has been downloaded!
The site 425 has been downloaded!
The site 426 has been downloaded!
The site 427 has been downloaded!
The site 428 has been downloaded!
The site 429 has been downloaded!
The site 430 has been downloaded!
The site 431 has been downloaded!
The site 432 has been downloaded!
The site 433 has been downloaded!
The site 434 has been downloaded!
The site 435 has been downloaded!
The site 436 has been downloaded!
The site 437 has been downloaded!
The site 438 has been downloaded!
The site 439 has been downloaded!
The site 440 has been downloaded!
The site 441 has NOT been downloaded from exception!
```

```
The site 442 has been downloaded!
The site 443 has been downloaded!
The site 444 has been downloaded!
The site 445 has been downloaded!
The site 446 has been downloaded!
The site 447 has been downloaded!
The site 448 has been downloaded!
The site 449 has been downloaded!
The site 450 has been downloaded!
The site 451 has been downloaded!
The site 452 has NOT been downloaded from exception!
The site 453 has been downloaded!
The site 454 has been downloaded!
The site 455 has been downloaded!
The site 456 has been downloaded!
The site 457 has been downloaded!
The site 458 has been downloaded!
The site 459 has been downloaded!
The site 460 has been downloaded!
The site 461 has been downloaded!
The site 462 has been downloaded!
The site 463 has been downloaded!
The site 464 has NOT been downloaded!
The site 465 has NOT been downloaded from exception!
The site 466 has been downloaded!
The site 467 has been downloaded!
The site 468 has been downloaded!
The site 469 has been downloaded!
The site 470 has been downloaded!
The site 471 has NOT been downloaded from exception!
The site 472 has been downloaded!
The site 473 has been downloaded!
The site 474 has been downloaded!
The site 475 has been downloaded!
The site 476 has been downloaded!
The site 477 has been downloaded!
The site 478 has been downloaded!
The site 479 has been downloaded!
The site 480 has been downloaded!
The site 481 has been downloaded!
The site 482 has been downloaded!
The site 483 has been downloaded!
The site 484 has been downloaded!
The site 485 has been downloaded!
The site 486 has been downloaded!
The site 487 has been downloaded!
The site 488 has been downloaded!
The site 489 has been downloaded!
```

```
The site 490 has been downloaded!
The site 491 has been downloaded!
The site 492 has been downloaded!
The site 493 has been downloaded!
The site 494 has been downloaded!
The site 495 has been downloaded!
The site 496 has been downloaded!
The site 497 has been downloaded!
The site 498 has been downloaded!
The site 499 has been downloaded!
```

In [8]:
```
#As we can see there is one site that hasn't been downloaded in order
#to keep track of the sites that we could not download
#we will create a new list that we will keep them all together there
not_d = []
not_d_n = []
num = []
def not_downloadables (list500_names,list500_sites):
    met = 0
    for i in range(len(list500_names)):
        if list500_sites[i] == 0:
            ct = list500_names[i]
            not_d.insert(met,ct)
            not_d_n.insert(met,str(i))
            num.insert(met,met)
            met = met + 1
```

In [9]:
```
#Now we will run the function to see which sites havent been downloaded
not_downloadables (list500_names,list500_sites)
d = {'company' : pd.Series(not_d, index=[num]),
     'number' : pd.Series(not_d_n, index=[num])}
nd = pd.DataFrame(d)
nd
```

Out[9]:

	company	number
0	Fannie Mae	15
1	Humana	51
2	HCA Holdings	62
3	Best Buy	70
4	Nike	90
5	Tesoro	97
6	Arrow Electronics	118
7	AutoNation	135
8	Southwest Airlines	141
9	Kohl’s	144
10	American Electric Power	164

```
11                 Office Depot    195
12                   PBF Energy    216
13           Consolidated Edison    228
14            Dominion Resources    242
15            L-3 Communications    244
16         Hertz Global Holdings    268
17               Global Partners    275
18  Discover Financial Services    282
19               PayPal Holdings    306
20             Republic Services    311
21                   News Corp.    326
22                     Williams    363
23         Auto-Owners Insurance    397
24                Tractor Supply    414
25   Old Republic International    441
26             Regions Financial    452
27              St. Jude Medical    464
28             Wyndham Worldwide    465
29    Host Hotels &amp;amp; Resorts    471
```

In [10]:
```
empty=[]
keyf = []
flesch = []
sentence =[]
word = []
unique_w = []
```

In [11]:
```
import time # I used it to see how much time it does to run the function
for num in range(0,500):
    site = list500_sites[num]
    line = list500_url[num]
    url_check = "http://www.webpagefx.com/tools/read-able/check.php?tab=Te
    browser = urllib2.build_opener()
    browser.addheaders = [('User-agent', 'Mozilla/5.0')]
    if site == 0 or num == 107:
        print("Site", str(num), "is not validated from sites")
        flesch.insert(num,"n/a")
        sentence.insert(num,"n/a")
        word.insert(num,"n/a")
        unique_w.insert(num,"n/a")
    else:
        try:
            response = browser.open(url_check)
        except Exception:
            flesch.insert(num,"n/a")
            sentence.insert(num,"n/a")
            word.insert(num,"n/a")
            unique_w.insert(num,"n/a")
```

```
            print("Site", str(num), "is not validated from check")
            continue
        html_r = response.read()
        check = str(html_r)
        if check != empty:
                soup = BeautifulSoup(check,"lxml")
                o = 0
                keyf = []
                for row in soup.html.body.findAll('tr'):
                    keyf.insert(o,row)
                    o = o + 1
                if keyf != empty:
                        #print("Site", str(num), "is validated")
                        #Flesh measurement
                        if keyf[0] != empty:
                            readability = str(keyf[0])
                            split1 = readability.split('>')
                            readability2 = str(split1[4])
                            split2 = readability2.split('<')
                            readability3 = str(split2[0])
                            flesch.insert(num,readability3)
                        else:
                            flesch.insert(num,"n/a")
                            sentence.insert(num,"n/a")
                            word.insert(num,"n/a")
                            unique_w.insert(num,"n/a")
                        #Number of sentences
                        if keyf[6] != empty:
                            sentences = str(keyf[6])
                            spli1 = sentences.split('>')
                            sentences2 = str(spli1[4])
                            spli2 = sentences2.split('<')
                            sentences3 = str(spli2[0])
                            sentence.insert(num,sentences3)
                        else:
                            flesch.insert(num,"n/a")
                            sentence.insert(num,"n/a")
                            word.insert(num,"n/a")
                            unique_w.insert(num,"n/a")
                        #Number of words
                        if keyf[7] != empty:
                            words = str(keyf[7])
                            spl1 = words.split('>')
                            words2 = str(spl1[4])
                            spl2 = words2.split('<')
                            words3 = str(spl2[0])
                            word.insert(num,words3)
                        else:
```

```
                                    flesch.insert(num,"n/a")
                                    sentence.insert(num,"n/a")
                                    word.insert(num,"n/a")
                                    unique_w.insert(num,"n/a")
                                #No. of complex words
                                if keyf[7] != empty:
                                    unique_ws = str(keyf[8])
                                    sp1 = unique_ws.split('>')
                                    unique_ws2 = str(sp1[4])
                                    sp2 = unique_ws2.split('<')
                                    unique_ws3 = str(sp2[0])
                                    unique_w.insert(num,unique_ws3)
                                else:
                                    flesch.insert(num,"n/a")
                                    sentence.insert(num,"n/a")
                                    word.insert(num,"n/a")
                                    unique_w.insert(num,"n/a")
                            else:
                                print("Site", str(num), "is not validated from che
                                flesch.insert(num,"n/a")
                                sentence.insert(num,"n/a")
                                word.insert(num,"n/a")
                                unique_w.insert(num,"n/a")
                time.sleep(2)
```

```
('Site', '0', 'is not validated from check 2')
('Site', '1', 'is not validated from check 2')
('Site', '11', 'is not validated from check 2')
('Site', '15', 'is not validated from sites')
('Site', '33', 'is not validated from check 2')
('Site', '37', 'is not validated from check 2')
('Site', '45', 'is not validated from check')
('Site', '51', 'is not validated from sites')
('Site', '58', 'is not validated from check 2')
('Site', '62', 'is not validated from sites')
('Site', '67', 'is not validated from check 2')
('Site', '70', 'is not validated from sites')
('Site', '82', 'is not validated from check 2')
('Site', '90', 'is not validated from sites')
('Site', '97', 'is not validated from sites')
('Site', '106', 'is not validated from check')
('Site', '107', 'is not validated from sites')
('Site', '118', 'is not validated from sites')
('Site', '125', 'is not validated from check 2')
('Site', '135', 'is not validated from sites')
('Site', '141', 'is not validated from sites')
('Site', '144', 'is not validated from sites')
('Site', '147', 'is not validated from check 2')
```

```
('Site', '155', 'is not validated from check 2')
('Site', '164', 'is not validated from sites')
('Site', '166', 'is not validated from check 2')
('Site', '171', 'is not validated from check 2')
('Site', '179', 'is not validated from check 2')
('Site', '195', 'is not validated from sites')
('Site', '209', 'is not validated from check 2')
('Site', '211', 'is not validated from check 2')
('Site', '216', 'is not validated from sites')
('Site', '225', 'is not validated from check 2')
('Site', '228', 'is not validated from sites')
('Site', '242', 'is not validated from sites')
('Site', '244', 'is not validated from sites')
('Site', '268', 'is not validated from sites')
('Site', '272', 'is not validated from check 2')
('Site', '275', 'is not validated from sites')
('Site', '282', 'is not validated from sites')
('Site', '297', 'is not validated from check 2')
('Site', '306', 'is not validated from sites')
('Site', '311', 'is not validated from sites')
('Site', '326', 'is not validated from sites')
('Site', '348', 'is not validated from check 2')
('Site', '363', 'is not validated from sites')
('Site', '366', 'is not validated from check 2')
('Site', '369', 'is not validated from check 2')
('Site', '375', 'is not validated from check 2')
('Site', '383', 'is not validated from check 2')
('Site', '384', 'is not validated from check 2')
('Site', '389', 'is not validated from check 2')
('Site', '390', 'is not validated from check')
('Site', '397', 'is not validated from sites')
('Site', '402', 'is not validated from check 2')
('Site', '404', 'is not validated from check 2')
('Site', '414', 'is not validated from sites')
('Site', '432', 'is not validated from check 2')
('Site', '438', 'is not validated from check')
('Site', '441', 'is not validated from sites')
('Site', '443', 'is not validated from check 2')
('Site', '447', 'is not validated from check 2')
('Site', '452', 'is not validated from sites')
('Site', '464', 'is not validated from sites')
('Site', '465', 'is not validated from sites')
('Site', '466', 'is not validated from check 2')
('Site', '468', 'is not validated from check 2')
('Site', '471', 'is not validated from sites')
```

```
In [12]: readability = []
```

```
In [13]: def readable (flesch):
             for i in range (len(flesch)):
                     f_n = flesch[i]
                     if f_n == "n/a":
                         readability.insert(i,"n/a")
                     else:
                         a = int(float(f_n))
                         if a > 90:
                             readability.insert(i,"Very easy")
                         elif a > 80:
                             readability.insert(i,"Easy")
                         elif a > 70:
                             readability.insert(i,"Fairly easy")
                         elif a > 60:
                             readability.insert(i,"Standard")
                         elif a > 50:
                             readability.insert(i,"Fairly difficult")
                         elif a > 30:
                             readability.insert(i,"Difficult")
                         else:
                             readability.insert(i,"Very Confusing")
             print "The function is completed!"

In [14]: readable (flesch)

The function is completed!


In [15]: d1 = {'company' : pd.Series(list500_names, index=[list500_num]),
               'url' : pd.Series(list500_url, index=[list500_num]),
               'Readability' : pd.Series(readability, index=[list500_num]),
               'Flesh_Mesaure' : pd.Series(flesch,index=[list500_num]),
          'Sentences' : pd.Series(sentence, index=[list500_num]),
          'Words' : pd.Series(word, index=[list500_num]),
          'Unique words' : pd.Series(unique_w, index=[list500_num])}
          fre = pd.DataFrame(d1)
          fre.head(3) #we see the first 3 in the data frame
          fre.to_csv('total_500_words.csv', sep=';')

Out[15]:   Flesh_Mesaure       Readability Sentences Unique words Words      compar
          1           n/a               n/a       n/a          n/a   n/a      Walmar
          2           n/a               n/a       n/a          n/a   n/a  Exxon Mobi
          3          56.2  Fairly difficult       123           25   273        Appl

                           url
          1     www.walmart.com
          2  www.exxonmobil.com
          3       www.apple.com
```

```
In [16]: #Retreiving the social media from each site
         #First create empty lists for the ones that
         #we will need to calculate
         sm_f = []
         sm_t = []
         sm_i = []
         sm_p = []
         sm_y = []
         sm_l = []
         sm_nm = []
         nm = []
         sm_url = []

In [17]: #Then create a function that will feel in those
         #lists so as to make the data frame later on
         def socialmedia (list500_sites,list500_names,list500_url):
             from time import time
             # I used it to see how much time it does to run the function
             start = time ()
             for i in range(len(list500_names)):
                     myHTML = list500_sites[i]
                     sm = ['facebook.com','twitter.com',
                           'instagram.com','pinterest.com',
                           'youtube.com','linkedin.com']
                     if myHTML == 0:
                         sm_nm.insert(i,list500_names[i])
                         nm.insert(i,i)
                         sm_url.insert(i,list500_url[i])
                         sm_f.insert(i,'n/a')
                         sm_t.insert(i,'n/a')
                         sm_i.insert(i,'n/a')
                         sm_p.insert(i,'n/a')
                         sm_y.insert(i,'n/a')
                         sm_l.insert(i,'n/a')
                     else:
                         for index in range(len(sm)):
                             x = sm[index]
                             social = re.findall(x,myHTML)
                             if (len(social) > 0):
                                 if x == 'facebook.com':
                                     answerf = 'TRUE'
                                 if x == 'twitter.com':
                                     answert = 'TRUE'
                                 if x == 'instagram.com':
                                     answeri = 'TRUE'
                                 if x == 'pinterest.com':
                                     answerp = 'TRUE'
                                 if x == 'youtube.com':
```

```
                                    answery = 'TRUE'
                                if x =='linkedin.com':
                                    answerl = 'TRUE'
                            else:
                                 if x == 'facebook.com':
                                    answerf = 'FALSE'
                                 if x == 'twitter.com':
                                    answert = 'FALSE'
                                 if x == 'instagram.com':
                                    answeri = 'FALSE'
                                 if x == 'pinterest.com':
                                    answerp = 'FALSE'
                                 if x == 'youtube.com':
                                    answery = 'FALSE'
                                 if x =='linkedin.com':
                                    answerl = 'FALSE'
                            sm_nm.insert(i,list500_names[i])
                            nm.insert(i,i)
                            sm_url.insert(i,list500_url[i])
                            sm_f.insert(i,answerf)
                            sm_t.insert(i,answert)
                            sm_i.insert(i,answeri)
                            sm_p.insert(i,answerp)
                            sm_y.insert(i,answery)
                            sm_l.insert(i,answerl)
                    end = time ()
                    duration = round (end - start, 3)
                    minutes = round (duration /60, 1)
                    print 'The lists are completed in ', minutes, ' minutes'
                    print 'The lists are ready in ', duration, ' seconds'
```

```
In [18]: #Now we will run the function for the 25 first sites for starters
         socialmedia (list500_sites,list500_names,list500_url)
```

```
The lists are completed in  0.0  minutes
The lists are ready in  0.273  seconds
```

```
In [19]: #Finally we create the data frame with the elements we found
         d2 = {'company' : pd.Series(sm_nm, index=[nm]),
               'facebook' : pd.Series(sm_f, index=[nm]),
                'twitter' : pd.Series(sm_t, index=[nm]),
               'instagram' : pd.Series(sm_i, index=[nm]),
                'pinterest' : pd.Series(sm_p, index=[nm]),
               'youtube' : pd.Series(sm_y, index=[nm]),
                'linkedin' : pd.Series(sm_l, index=[nm]),}
         social_media = pd.DataFrame(d2)
         social_media.tail(3) #we see the first 3 in the data frame
         social_media.to_csv('total_500_sm.csv', sep=';')
```

Out[19]:
```
                    company facebook instagram linkedin pinterest twitter  \
497                     NVR     TRUE      TRUE    FALSE     FALSE   FALSE
498  Cincinnati Financial       TRUE     FALSE    FALSE     FALSE   FALSE
499     Burlington Stores       TRUE      TRUE    FALSE      TRUE    TRUE

    youtube
497    TRUE
498   FALSE
499    TRUE
```

In [20]:
```
#Create the lists we will need for the data frame
l_nm = []
l_ex = []
l_in = []
l_t = []
nm = []
l_url = []
```

In [21]:
```
#create the function that will calculate the different type of links
def links (list500_sites,list500_names,list500_url):
    from time import time
    # I used it to see how much time it does to run the function
    start = time ()
    for num in range(len(list500_names)):
            myHTML = list500_sites[num]
            if myHTML == 0:
                l_nm.insert(num,list500_names[num])
                l_ex.insert(num,'n/a')
                l_t.insert(num,'n/a')
                l_in.insert(num,'n/a')
                nm.insert(num,num)
            else:
                href = re.findall('href',myHTML)
                external = re.findall('href="https:',myHTML)
                ex = (len(external))
                alllinks = (len(href))
                internal =  (len(href) - len(external))
                l_nm.insert(num,list500_names[num])
                l_ex.insert(num,ex)
                l_t.insert(num,alllinks)
                l_in.insert(num,internal)
                nm.insert(num,num)
    end = time ()
    duration = round (end - start, 3)
    minutes = round (duration /60, 1)
    print 'The lists are ready in ', minutes, ' minutes'
    print 'The lists are ready in ', duration, ' seconds'
```

In [22]:
```
#Run the function in order to find the external,
```

```
        #internal and total links of each site
        #For now we are running for the first 25 sites only
        links (list500_sites,list500_names,list500_url)

The lists are ready in  0.0  minutes
The lists are ready in  0.086  seconds


In [23]: #Create a dataframe so as to be able to see
         #the results of the function we run
         d3 = {'company' : pd.Series(l_nm, index=[nm]),
               'external' : pd.Series(l_ex, index=[nm]),
               'internal' : pd.Series(l_in, index=[nm]),
              'total links' : pd.Series(l_t, index=[nm])}
         sites_links = pd.DataFrame(d3)
         sites_links.tail(3) #we see the first 3 in the data frame
         sites_links.to_csv('total_500_sites_links.csv', sep=';')

Out[23]:                    company external internal total links
         497                    NVR        9       17          26
         498  Cincinnati Financial        3       61          64
         499     Burlington Stores       16      147         163

In [24]: #The initial lists we will need in order
         #to calculate the loading time
         lt_nm = []
         lt_time = []
         nm = []
         lt_url = []

In [ ]: #the function that will calculate the loading time
        def loadtime (list_company_website,list500_names,list500_url):
            from time import time
            browser2 = urllib2.build_opener()
            browser2.addheaders = [('User-agent', 'Mozilla/5.0')]
            for num in range(len(list500_names)):
                lc = str(list_company_website[num])
                lc = lc.replace("'","")
                lc = lc.replace("[","")
                lc = lc.replace("]","")
                url2 = 'http://' + lc
                if num == 118 or num == 464 or num == 70:
                    #The site 118(119) has a problem and the whole code
                    #is stacking when I run it so we will thing of this
                    #site as a not downloadable
                    lt_nm.insert(num,list500_names[num])
                    lt_time.insert(num,'n/a')
                    nm.insert(num,num)
                    lt_url.insert(num,list500_url[num])
```

```
            else:
                try:
                    response2 = browser2.open(url2)
                except Exception:
                    lt_time.insert(num,'n/a')
                    lt_nm.insert(num,list500_names[num])
                    nm.insert(num,num)
                    print ("The site " + str(num)+ " has NOT been loaded!")
                    continue
                start_time = time()
                myHTML2 = response2.read()
                end_time = time()
                response2.close()
                l_t = round(end_time-start_time, 3)
                #in order to be more readable we rounded the time
                loadt = str(l_t)
                lt_nm.insert(num,list500_names[num])
                lt_time.insert(num,loadt)
                nm.insert(num,num)
                lt_url.insert(num,list500_url[num])
                #print ("The site " + str(num) + " has been loaded!")
        print "The function is completed!"
```

In [ ]:
```
#running the function for the first 25 sites
loadtime (list_company_website,list500_names,list500_url)
```

In [ ]:
```
#creating the data frame with the loading times
d4 = {'company' : pd.Series(lt_nm, index=[nm]),
      'loading time' : pd.Series(lt_time, index=[nm])}
loading_time = pd.DataFrame(d4)
loading_time.head(3) #we see the first 3 in the data frame
loading_time.to_csv('loading_time.csv', sep=';')
```

In [ ]:
```
#Find out how many and what type of images each site has
#first we create the initially empty lists
p_p = []
p_d = []
p_jpg = []
p_jpeg = []
p_gif = []
p_tif = []
p_tiff = []
p_bmp = []
p_jpe = []
p_nm = []
p_tt =[]
nm = []
p_url = []
```

```
In [ ]: #Then we create the function that will explore
        #the html pages and search for the images
        def images (list500_sites,list500_names,list500_url):
            from time import time # I used it to see
            #how much time it does to run the function
            start = time ()
            for num in range(len(list500_names)):
                    myHTML = list500_sites[num]
                    image = ['.png','.dib','.jpg','.jpeg',
                             '.bmp','.jpe','.gif','.tif','.tiff']
                    totalnumber = 0
                    if myHTML == 0:
                        p_nm.insert(num,list500_names[num])
                        p_p.insert(num,'n/a')
                        p_d.insert(num,'n/a')
                        p_jpg.insert(num,'n/a')
                        p_jpeg.insert(num,'n/a')
                        p_gif.insert(num,'n/a')
                        p_tif.insert(num,'n/a')
                        p_tiff.insert(num,'n/a')
                        p_bmp.insert(num,'n/a')
                        p_jpe.insert(num,'n/a')
                        p_tt.insert(num,'n/a')
                        nm.insert(num,num)
                        p_url.insert(num,list500_url[num])
                    else:
                        for index in range(len(image)):
                            x = image[index]
                            photo = re.findall(x,myHTML)
                            if x == '.png':
                                p = str (len(photo))
                            if x == '.dib':
                                d = str (len(photo))
                            if x == '.jpg':
                                jpg = str (len(photo))
                            if x == '.jpeg':
                                jpeg = str (len(photo))
                            if x == '.gif':
                                gif = str (len(photo))
                            if x == '.tif':
                                tif = str (len(photo))
                            if x == '.tiff':
                                tiff = str (len(photo))
                            if x == '.bmp':
                                bmp = str (len(photo))
                            if x == '.jpe':
                                jpe = str (len(photo))
                            totalnumber = len(photo) + totalnumber
```

```
                        total = str (totalnumber)
                        p_nm.insert(num,list500_names[num])
                        p_p.insert(num,p)
                        p_d.insert(num,d)
                        p_jpg.insert(num,jpg)
                        p_jpeg.insert(num,jpeg)
                        p_gif.insert(num,gif)
                        p_tif.insert(num,tif)
                        p_tiff.insert(num,tiff)
                        p_bmp.insert(num,bmp)
                        p_jpe.insert(num,jpe)
                        p_tt.insert(num,total)
                        nm.insert(num,num)
                        p_url.insert(num,list500_url[num])
            end = time ()
            duration = round (end - start, 3)
            minutes = round (duration /60, 1)
            print 'The lists are ready in ', minutes, ' minutes'
            print 'The lists are ready in ', duration, ' seconds'
```

In [ ]:
```
images (list500_sites,list500_names,list500_url)
```

In [ ]:
```
d5 = {'company' : pd.Series(p_nm, index=[nm]),
      '.png' : pd.Series(p_p, index=[nm]),
      '.dib' : pd.Series(p_d, index=[nm]),
      '.jpg' : pd.Series(p_jpg, index=[nm]),
      '.jpeg' : pd.Series(p_jpeg, index=[nm]),
      '.bmp' : pd.Series(p_bmp, index=[nm]),
      '.jpe' : pd.Series(p_jpe, index=[nm]),
      '.gif' : pd.Series(p_gif, index=[nm]),
      '.tif' : pd.Series(p_tif, index=[nm]),
      '.tiff' : pd.Series(p_tiff, index=[nm]),
      'total images' : pd.Series(p_tt, index=[nm])}
images_types = pd.DataFrame(d5)
images_types.head(3) #we see the first 3 in the data frame
images_types.to_csv('images_types.csv', sep=';')
```

In [ ]:
```
#Now we will gather the total number of pixels that exist
#in each web page under examination
#with this we will have a grasp of the images that each site is using
im_nm = []
im_pix = []
import time
def url_to_image(list500_names, list500_url):
    import numpy as np
    import urllib
```

```
        for num in range(len(list500_names)):
            url= 'http://' + list500_url[num]
            if num == 45 or num == 118 or num == 390:
                print (str(num), "Exception")
                im_pix.insert(num,"n/a")
                im_nm.insert(num,list500_names[num])
            else:
                try: # download the image, convert it to a NumPy array
                    resp = urllib.urlopen(url)
                except Exception:
                    print (str(num), "Exception")
                    im_pix.insert(num,"n/a")
                    im_nm.insert(num,list500_names[num])
                    continue
                image = np.asarray(bytearray(resp.read()), dtype="uint8")
                #print(image)
                pixels = 0
                #add up all the pixels of the page
                for n in range (len(image)):
                    pixels = pixels + image[n]
                print (str(num), pixels)
                im_nm.insert(num,list500_names[num])
                im_pix.insert(num,pixels)
            time.sleep(2)
```

In [ ]:
```
#Run the function
url_to_image(list500_names, list500_url)
```

In [ ]:
```
#Finally we create a dataframe in order to see the results of the function
dpix = {'company' : pd.Series(im_nm, index=[list500_num]),
      'pixels' : pd.Series(im_pix, index=[list500_num])}
images_pixels = pd.DataFrame(dpix)
images_pixels.head(3) #we see the first 3 in the data frame
images_pixels.to_csv('images_pixels.csv', sep=';')
```

In [ ]:
```
#In order to validate the html code we will use the w3 validator
#We will validate each url and then we will open the url of the validation
#so as to extract the errors, the info warnings and the non-document-error
#First we create the empty lists we would use later on
num_errors = []
num_info_warnings = []
num_non_doc = []
nm = []
num_open_page = []
empty = ""
```

In [ ]:
```
#Then we create the function that will pull the informations we want
def html_validation (list500_url,list500_names):
    from time import time # I used it to see how much time it does to run t
```

```
start = time ()
for num in range(len(list500_names)):
    line = list500_url[num]
    url_check = "https://validator.w3.org/nu/?doc=https://" + line
    browser = urllib2.build_opener()
    browser.addheaders = [('User-agent', 'Mozilla/5.0')]
    response = browser.open(url_check)
    html_check = response.read()
    html_check
    check = str(html_check)
    er = 0
    err = 0
    errr = 0
    e = False
    if check != empty:
        e = True
        soup = BeautifulSoup(check,"lxml")
        o = 0
        keyf = []
        for row in soup.html.body.findAll('div'):
            keyf.insert(o,row)
            o = o + 1
        #print(len(keyf),list500_url[num], "site number: ", str(num))
        if len(keyf) != 0:
                keyfin = str(keyf[2])
                #the elements we need is in the 2nd div of the code
                dol= re.findall('class="error"',keyfin)
                er = er + len(dol)
                doll= re.findall('class="info warning"'
                                 ,keyfin)
                err = err + len(doll)
                dolll= re.findall('class="non-document-error io"'
                                  ,keyfin)
                errr = errr + len(dolll)
    num_errors.insert(num,er)
    num_info_warnings.insert(num,err)
    num_non_doc.insert(num,errr)
    nm.insert(num,num)
    num_open_page.insert(num,e)
end = time ()
duration = round (end - start, 3)
minutes = round (duration /60, 1)
print 'The lists are ready in ', minutes, ' minutes'
```

In [ ]:
```
#Now we will run the function we created
html_validation (list500_url,list500_names)
```

In [ ]:
```
#After the checks we will create the dataframe with the informations we wa
d8 = {'company' : pd.Series(list500_names, index=[nm]),
```

```
            'The_page_opened' : pd.Series(num_open_page, index=[nm])
            ,'number_of_errors' : pd.Series(num_errors, index=[nm]),
            'number_of_warning' : pd.Series(num_info_warnings, index=[nm])
            ,'non-document-error' : pd.Series(num_non_doc, index=[nm])}
        html_val = pd.DataFrame(d8)
        html_val.head(3)
        html_val.to_csv('html_val.csv', sep=';')
```

In [ ]:
```
#The next step is to take some informations from the fortune 500 site for e
#In order to achieve that we should open the pages for each one of the site
#Since there is a pattern in the way the pages are named it shouldn't be di
#Firstly we should create the pattern with which we will download the pages
#By running the code we can see that the names of each comany are not
#written exactly as we have saved them
#So we do need to alter the names first in order for the below function to
```

In [ ]:
```
#creating a new list with alterations in order for the names
#to match the ones that fortune 500 uses so that we can download the html p
list_company_name_new = []
for num in range (0,500):
    cn = list_company_name[num]
    cn = cn.replace(" ", "-")
    cn = cn.replace("&", "")
    cn = cn.replace("'", "")
    cn = cn.replace(".", "-")
    cn = cn.replace("amp;", "")
    company = cn.lower()
    list_company_name_new.insert(num,cn)
```

In [ ]:
```
fortune_pages = []
def fortune500 (list_company_name_new):
    from time import time # I used it to see how much time it does to run t
    start = time ()
    for num3 in range (0,500):
        i = str (num3 +1)
        companyname =  list_company_name_new[num3]
        browser = urllib2.build_opener()
        #because i work from different computers with different
        #pyhton version some commands are not recognizable in each version
        browser.addheaders = [('User-agent', 'Mozilla/5.0')]
        site_fortune = "http://beta.fortune.com/fortune500/"+companyname+"-
        page_fortune = browser.open(site_fortune)
        html_fortune = page_fortune.read()
        #print("fortune page for company: ", list_company_name_new[num3],i)
        fortune_pages.insert(num3, html_fortune)
    end = time ()
    duration = round (end - start, 3)
    minutes = round (duration /60, 1)
```

```
            print 'The lists are ready in ', minutes, ' minutes'
            print 'The lists are ready in ', duration, ' seconds'

In [ ]: #Run the function we created
        fortune500 (list_company_name_new)

In [ ]: #Now that we have opened the url we are going to extract
        #some informations that we need from them
        #In order to do that initially we have to create
        #the variables we will need
        keyf =[]
        keyfaw = []
        per =[]
        rev_dol = []
        prof_dol = []
        assets_dol = []
        tse_dol = []
        tse_per = []
        mar_dol = []
        market = []
        nm = []
        ln = []
        urln = []
        empty = []
        pef = []
        pe_rat =[]
        pr_r = []
        peratio = []
        empty = []
        keyfawin = 0

In [ ]: def fortune_metrics (list_company_name,list_company_website):
            for n in range (0,500):
                print("site",str(n), list_company_website[n])
                x = 0
                nm.insert(x,n)
                ln.insert(x,list_company_name[n])
                urln.insert(x,list_company_website[n])
                files = fortune_pages[n]
                soup = BeautifulSoup(files, 'lxml')
                ##############################################
                o=0
                for row in soup.html.body.findAll('tbody'):
                    keyf.insert(o,row)
                    o = o + 1
                keyfin = keyf[0]  #the elements we need is in the first tbody of th
                keyfon = keyf[1]
                data = keyfin.findAll('td')
```

```
        ##################################
        # revenue
        two = str(data[1])
        # revenue in dollars we need to extract this
        revdol= re.findall('>\$(.+?)</td>',two)
        #we keep only the numbers
        if revdol[0] != empty:
            w = revdol[0]
            a = w.replace("[", "")
            r = a.replace("]","")
            rev_dol.insert(x,r)
        else:
            rev_dol.insert(x,'not available')
        five = str(data[5])
        ##################################
        # profit in dollars we need to extract this
        profdol= re.findall('>\$(.+?)</td>',five)
        #we keep only the numbers
        if profdol != empty:
            w = profdol[0]
            a = w.replace("[", "")
            p = a.replace("]","")
            prof_dol.insert(x,p)
        else:
            prof_dol.insert(x,'not available')
        ##################################
        #assets in dollard
        eight = str(data[7]) #assets in dollars we need to extract this
        assetsdol= re.findall('>\$(.+?)</td>',eight)
        #we keep only the numbers
        if assetsdol != empty:
            w = assetsdol[0]
            a = w.replace("[", "")
            ass = a.replace("]","")
            assets_dol.insert(x,ass)
        else:
            assets_dol.insert(x,'not available')
        ##################################
       #Total Stockholder Equity ($M)
        eleven = str(data[10])
        #Total Stockholder Equity ($M) in dollars we need to extract this
        tsedol= re.findall('>\$(.+?)</td>',eleven)
        #we keep only the numbers
        if tsedol != empty:
            w = tsedol[0]
            a = w.replace("[", "")
            ts = a.replace("]","")
            tse_dol.insert(x,ts)
```

```
                else:
                    tse_dol.insert(x,'not available')
                ######################################
                # market value
                fourteen = str(data[13])
                # market value in dollars we need to extract this
                mardol= re.findall('>\$(.+?)</td>',fourteen)
                #we keep only the numbers
                if mardol != empty:
                    w = mardol[0]
                    a = w.replace("[", "")
                    mar = a.replace("]","")
                    mar_dol.insert(x,mar)
                else:
                    mar_dol.insert(x,'not available')
                ############################################################
                datao = keyfon.findAll('td')
                # Profit as % of Revenues
                twoo = str(datao[2])
                prof_rev= re.findall('>(.+?)%</td>',twoo)
                #we keep only the numbers
                if  prof_rev[0] != empty:
                    w = prof_rev[0]
                    a = w.replace("[", "")
                    r = a.replace("]","")
                    pr_r.insert(x,r)
                else:
                    pr_r.insert(x,'not available')
                x = x + 1
            print "The function is complete!"

In [ ]: fortune_metrics (list_company_name,list_company_website)

In [ ]: d9 = {'company' : pd.Series(ln, index=[nm]),
              'Revenues $' : pd.Series(rev_dol, index=[nm]),
              'Profits $' : pd.Series(prof_dol, index=[nm]),
              'Assets $' : pd.Series(assets_dol, index=[nm]),
              'Total Stockholder Equity $' : pd.Series(tse_dol, index=[nm]),
        'Proft_%_Revenues' : pd.Series(pr_r, index=[nm]),
              'Market value $' : pd.Series(mar_dol, index=[nm])}
        fort500 = pd.DataFrame(d9)
        fort500.head(3)
        fort500.to_csv('fort500.csv', sep=';')

In [ ]: result = pd.merge(fort500, html_val, how='inner', on=['company', 'company']
        result2 = pd.merge(social_media, fre, how='inner', on=['company', 'company'
        result3 = pd.merge(sites_links, images_pixels, how='inner', on=['company',
        result4 = pd.merge(images_types, loading_time, how='inner', on=['company',
```

```
        result5 = pd.merge(result,result2 , how='inner', on=['company', 'company'])
        result6 = pd.merge(result3, result4, how='inner', on=['company', 'company']
        final = pd.merge(result5, result6, how='inner', on=['company', 'company'])
        final.head(3)

In [ ]: final.to_csv('total_500_final.csv', sep=';')

In [ ]: data500 = pd.read_csv("total_500_final.csv", sep=';')

In [ ]: data500.head(3)

In [ ]:
```