**'start'**: 0,  
 **'end'**: 29,  
 **'len'**: 30,  
 **'pageNum'**:10,  
 **'redisTopStart'**:0,  
 **'redisTopEnd'**:5  
}  
  
EXPIRE\_TIME = 3600  
MIN\_SCORE = 0.75  
HTTP = **'http://'**DEAL\_PARAMS = {  
 **'featureDS'**: **'featureDS'**,  
 **'imgPathDS'**: **'imgPathDS'**}  
  
UPLOAD\_TYPE = {  
 **"SUBMIT"**: **"SUBMIT"**,  
 **"DRAG"**: **"DRAG"**,  
 **"URL"**: **"URL"**,  
 **"LINK"**: **"LINK"**,  
 **"NONE"**: **"NONE"**}  
CONSTVALUE={  
 **'NULL\_DATE'**:**'0001-01-01'**,  
 **'SHOW\_CHAR\_NUM'**:100  
}  
URL\_PARAM={  
 **'differeNum'**:-5,  
 **'BACKLASH'**:**'+Z\_\*-'**,  
 **'QUEST'**:**'\*+Y\_-'**,  
 **"URLDOT"**:**"\_XTA\_"**}  
  
**def** GetMd5(url):  
 **if** isinstance(url, str):  
 url = url.encode(**"utf-8"**)  
 m = hashlib.md5()  
 m.update(url)  
 **return** m.hexdigest()  
  
**def** CreateImgNamePath(imgName, loadType):  
 **if** loadType != UPLOAD\_TYPE[**'LINK'**]:  
 imgPath = settings.UPLOAD\_DIR + imgName + **'.jpg'  
 return** imgPath, imgName  
  
 imgPath = os.path.join(settings.LINK\_DIR, imgName)  
 imgName = GetMd5(imgName)  
 **return** imgPath, imgName  
  
  
**def** DownloadImgData(imgInfo, loadType, imgName):  
 imgPath, imgName = CreateImgNamePath(imgName, loadType)  
 **if** loadType == UPLOAD\_TYPE[**'URL'**]:  
 urllib.request.urlretrieve(url=imgInfo, filename=imgPath)  
 **else**:  
 default\_storage.save(imgPath, ContentFile(imgInfo.read()))  
 **return** imgPath, imgName  
  
  
**def** ReturnTopn(start, end, conn):  
 topnSearch = conn.zrevrangebyscore(**"hotSearchSet"**, **"+inf"**, **"-inf"**, start=start, num=end)  
 strList = [(bytes.decode(byItem),urllib.parse.quote(byItem)) **for** byItem **in** topnSearch]  
 **return** strList  
  
  
**def** FetchImgPath(imgName, start, end, conn):  
 byteList = conn.zrangebyscore(imgName, start, end)  
 strList = [bytes.decode(byItem) **for** byItem **in** byteList]  
 **return** strList  
  
  
**def** OnlyImgPath(imgStr):  
 **return** os.path.splitext(imgStr)[0] + **'.jpg'  
  
  
def** CreateImgName():  
 **return** str(uuid.uuid4()).replace(**'-'**, **''**)  
  
  
**def** ChangeResImg(resImgList, client):  
 urlList = [HTTP + ChangeNameToUrl(item)[0] **for** item **in** resImgList]  
 urlTitle = FetchTitle(client, urlList)  
 targetList = []  
 **for** item **in** resImgList:  
 imgPath = OnlyImgPath(item)  
 imgUrl, imgSize = ChangeNameToUrl(item)  
 imgTitle = urlTitle[imgUrl] **if** imgUrl **in** urlTitle **else ''** targetList.append([imgPath, imgUrl, imgSize, imgTitle])  
 **return** targetList  
  
  
**def** ReadJsonData():  
 dic = {}  
 **with** codecs.open(settings.JSON\_DATA, **'r'**, encoding=**'utf-8'**) **as** f:  
 dic = json.load(f)  
 **return** dic  
  
  
**def** ChangeNameToUrl(imgName):  
 **try**:  
 imgPath = os.path.split(imgName)[1]  
 pathSplit = os.path.splitext(imgPath)  
 imgSize= pathSplit[1][4:]  
 imgUrl=pathSplit[0][:URL\_PARAM[**'differeNum'**]].replace(URL\_PARAM[**'BACKLASH'**], **'/'**)  
 imgUrl=imgUrl.replace(URL\_PARAM[**'QUEST'**], **'?'**)  
 **except**:  
 imgSize = **''** imgUrl=**''  
 return** imgUrl,imgSize  
  
  
**def** GetDicH5py():  
 **try**:  
 targetDics = [int(dic[len(settings.SEARCH\_DIR[:-1]):]) **for** dic **in** glob.glob(settings.SEARCH\_DIR)]  
 maxtargetDic = 0 **if not** len(targetDics) **else** max(targetDics) + 1  
 h5File = h5py.File(settings.H5PY\_FILE, **'r'**)  
 **except**:  
 maxtargetDic = 0  
 h5File = **None  
 return** maxtargetDic, h5File  
  
  
**def** FetchTitle(client, urlList):  
 **try**:  
 response = client.search(  
 index=**"hfut\_search"**,  
 body={  
 **"\_source"**: [**"url"**, **"title"**],  
 **"query"**: {  
 **"terms"**: {  
 **"url"**: urlList  
 }  
 },  
 **"size"**: 30  
 }  
 )  
 hitList = {}  
 **for** hit **in** response[**"hits"**][**"hits"**]:  
 **if "title" in** hit[**"\_source"**] **and "url" in** hit[**"\_source"**]:  
 hitList[hit[**"\_source"**][**"url"**][len(HTTP):]] = hit[**"\_source"**][**"title"**]  
 **except**:  
 hitList = {}  
  
 **return** hitList  
  
**def** GetStripLabelLen(str):  
 **return** len(re.sub(**'</?span.\*?>'**,**''**,str))

*# -\*- coding: utf-8 -\*-***from** elasticsearch\_dsl **import** Document, Completion, Keyword, Text,Date,Boolean  
**from** elasticsearch\_dsl.analysis **import** CustomAnalyzer **as** CustomAnaly  
**from** elasticsearch\_dsl.connections **import** connections  
  
  
**class** CustomAnalyzer(CustomAnaly):  
 **def** get\_analysis\_definition(self):  
 **return** {}  
  
  
**class** SearchType(Document):  
 url = Keyword()  
 url\_origin = Keyword()  
 title = Text(analyzer=**"ik\_max\_word"**)  
 content = Text(analyzer=**"ik\_max\_word"**)  
 img\_download = Keyword()  
 is\_index=Boolean()  
 create\_date = Date()  
 save\_data=Date()  
 suggest = Completion(analyzer=CustomAnalyzer(**"ik\_max\_word"**, filter=[**"lowercase"**]))  
 **class** Index:  
 name = **'hfut\_search'  
  
 class** Meta:  
 doc\_type = **'hfut\_type'  
  
  
class** SearchPicSave(Document):  
 img\_name=Keyword()  
 img\_path=Keyword()  
 save\_data=Date()  
 **class** Index:  
 name = **'hfut\_pic'  
  
 class** Meta:  
 doc\_type = **'hfut\_pic\_type'  
  
if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 params={  
 **"host"**: **"localhost"**,  
 **"port"**: 16110  
 }  
 connections.create\_connection(\*\*params)  
 SearchType.init()  
 SearchPicSave.init()

**import** numpy **as** np  
**from** numpy **import** linalg  
**from** keras.applications.vgg16 **import** VGG16  
**from** keras.preprocessing **import** image  
**from** keras.applications.vgg16 **import** preprocess\_input  
**from** search.common **import** VGG\_PARAMS  
**from** PIL **import** Image **as** pilImage  
  
  
**class** VGGNet:  
 **def** \_\_init\_\_(self, \*\*kwargs):  
 self.inputShape = kwargs[**'inputShape'**]  
 self.weight = kwargs[**'weight'**]  
 self.pooling = kwargs[**'pooling'**]  
 self.includeTop = kwargs[**'includeTop'**]  
 self.modelVGG = VGG16(weights=self.weight, input\_shape=self.inputShape, pooling=self.pooling, include\_top=**False**)  
  
 **def** VGGExtractFeat(self, imgPath):  
 img = image.load\_img(imgPath)  
 imgSize=img.size  
 changeSize=(self.inputShape[0], self.inputShape[1])  
 **if** imgSize!=changeSize:  
 img=img.resize(changeSize,pilImage.NEAREST)  
 img = image.img\_to\_array(img)  
 img = np.expand\_dims(img, axis=0)  
 img = preprocess\_input(img)  
 feat = self.modelVGG.predict(img)  
 normFeat = feat[0] / linalg.norm(feat[0])  
 **return** normFeat,imgSize  
  
 @classmethod  
 **def** CreateVGGNet(cls):  
 **return** cls(\*\*VGG\_PARAMS)

*# \_\*\_ encoding:utf-8 \_\*\_***import** json  
**from** django.shortcuts **import** render  
**from** django.views.generic.base **import** View  
**from** django.http **import** HttpResponse  
**from** datetime **import** datetime  
**from** django\_redis **import** get\_redis\_connection  
**from** search.VGGNet **import** VGGNet  
**import** numpy **as** np  
**from** PIL **import** Image  
**from** search **import** common  
**from** elasticsearch\_dsl.search **import** Search  
**from** elasticsearch\_dsl.connections **import** connections  
  
model = VGGNet.CreateVGGNet()  
conn = get\_redis\_connection(**"default"**)  
  
client = connections.create\_connection(\*\*common.connStr)  
jsonData = common.ReadJsonData()  
  
  
**class** IndexView(View):  
 **def** get(self, request):  
 topnSearch = common.ReturnTopn(common.FETCH\_NUMBER\_DEF[**'redisTopStart'**],  
 common.FETCH\_NUMBER\_DEF[**'redisTopEnd'**], conn)  
 **return** render(request, **"index.html"**, {**"topnSearch"**: topnSearch})  
  
  
**class** UploadPictureMiddle(View):  
 **def** post(self, request, uploadType):  
 imgInfo = request.FILES.get(**'file'**)  
 imgPath, imgName = common.DownloadImgData(imgInfo, uploadType, common.CreateImgName())  
 **return** HttpResponse(**'{"imgName":"'** + imgName + **'","uploadType":"'** + uploadType + **'"}'**,  
 content\_type=**'application/json'**)  
  
  
**class** AjaxPictureMiddle(View):  
 **def** ChangeInfoToHtml(self, resImgList, pageIndex):  
 jsonObject = **'{"pageIndex":"'** + pageIndex + **'","imgList":'** + json.dumps(resImgList) + **'}'  
 return** jsonObject  
  
 **def** get(self, request):  
 imgToken = request.GET.get(**'imgToken'**)  
 start = request.GET.get(**'start'**)  
 end = request.GET.get(**'end'**)  
 imglist = common.FetchImgPath(imgToken, start, end, conn)  
 resImgList = common.ChangeResImg(imglist, client)  
 **try**:  
 pageIndex = int(int(start) / common.FETCH\_NUMBER\_DEF[**'len'**] + 1)  
 **except**:  
 pageIndex = 0  
 htmlCode = self.ChangeInfoToHtml(resImgList, str(pageIndex))  
 **return** HttpResponse(htmlCode, content\_type=**'application/json'**)  
  
  
**class** SearchPictureView(View):  
  
 **def** CreateZADD(self, imgName, sortTotalDict):  
 **for** index, item **in** enumerate(sortTotalDict):  
 conn.zadd(imgName, mapping={item[0]: index})  
 conn.expire(imgName, common.EXPIRE\_TIME)  
  
 **def** FetchUserInfo(self, predictImgUrl, imgPath, imgName, start, end, total, querySize):  
 userUpload = {}  
 **try**:  
 userUpload[**"imgUrl"**] = imgPath.split(**'static/'**)[1]  
 **except**:  
 userUpload[**"imgUrl"**] = imgPath  
 userUpload[**"infoGuess"**] = predictImgUrl  
 userUpload[**"imgName"**] = imgName  
 urlDic = common.FetchTitle(client, [common.HTTP + predictImgUrl])  
 userUpload[**"title"**] = **''  
 if** predictImgUrl **in** urlDic:  
 userUpload[**"title"**] = urlDic[predictImgUrl]  
 userUpload[**"height"**] = querySize[1]  
 userUpload[**"width"**] = querySize[0]  
 userUpload[**'start'**] = start  
 userUpload[**'end'**] = end  
 userUpload[**'total'**] = total  
 **return** userUpload  
  
 **def** ReturnImgsPath(self, imgPath):  
 sortTotalDict = {}  
 querySize = (0, 0)  
 maxtargetDic, h5File = common.GetDicH5py()  
 **if** h5File:  
 **try**:  
 totalDict = {}  
 queryVec, querySize = model.VGGExtractFeat(imgPath)  
 **for** item **in** range(maxtargetDic):  
 featStr = common.DEAL\_PARAMS[**'featureDS'**] + str(item)  
 imgStr = common.DEAL\_PARAMS[**'imgPathDS'**] + str(item)  
 **if** featStr **in** h5File **and** imgStr **in** h5File:  
 feats = h5File[featStr][:]  
 imgPaths = h5File[imgStr][:]  
 scores = np.dot(queryVec, feats.T)  
 validData = np.where(scores > common.MIN\_SCORE)[0]  
 **for** item **in** validData:  
 totalDict[imgPaths[item]] = scores[item]  
 sortTotalDict = sorted(totalDict.items(), key=**lambda** x: x[1], reverse=**True**)  
 h5File.close()  
 **except**:  
 h5File.close()  
 **return** sortTotalDict, querySize  
  
 **def** get(self, request, uploadType):  
 imgInfo = request.GET[**'query'**]  
 querySize = (0, 0)  
 imgPath, imgName = common.CreateImgNamePath(imgInfo, uploadType) \  
 **if** uploadType == common.UPLOAD\_TYPE[**'DRAG'**] **or** uploadType == common.UPLOAD\_TYPE[**'LINK'**] **or** uploadType == \  
 common.UPLOAD\_TYPE[**'SUBMIT'**] \  
 **else** common.DownloadImgData(imgInfo, uploadType, common.CreateImgName())  
 **if not** conn.exists(imgName):  
 sortTotalDict, querySize = self.ReturnImgsPath(imgPath)  
 totalNum = len(sortTotalDict)  
 self.CreateZADD(imgName, sortTotalDict)  
 **else**:  
 totalNum = conn.zcard(imgName)  
 fp = Image.open(imgPath)  
 **if** fp:  
 querySize = fp.size  
 resImgList = common.FetchImgPath(imgName, common.FETCH\_NUMBER\_DEF[**'start'**],  
 common.FETCH\_NUMBER\_DEF[**'end'**], conn)  
 upLoadInfo = self.FetchUserInfo(common.ChangeNameToUrl(resImgList[0])[0] **if** len(resImgList) **else ''**, imgPath,  
 imgName, common.FETCH\_NUMBER\_DEF[**'start'**],  
 common.FETCH\_NUMBER\_DEF[**'end'**], totalNum, querySize)  
  
 resImgList = common.ChangeResImg(resImgList, client)  
 **return** render(request, **"pictureResult.html"**, {**"userUpload"**: upLoadInfo, **"filesInfo"**: resImgList})  
  
  
**class** SearchSuggest(View):  
 **def** get(self, request):  
 keyWords = request.GET.get(**'s'**, **''**)  
 kw = {**'using'**: client,  
 **'index'**: **'hfut\_search'**,  
 **'doc\_type'**: **'hfut\_type'** }  
 sugg = Search(\*\*kw)  
 sugg = sugg.suggest(**'my\_suggest'**, keyWords, completion={  
 **"field"**: **"suggest"**,  
 **"size"**: common.FETCH\_NUMBER\_DEF[**'pageNum'**]  
 })  
 sugg = sugg.execute()  
 options = sugg.suggest[**'my\_suggest'**][0].options  
 reDatas = [match.\_source[**"title"**] **for** match **in** options]  
 **return** HttpResponse(json.dumps(reDatas), content\_type=**"application/json"**)  
  
**class** AjaxCollegeMiddle(View):  
 **def** get(self, request):  
 **return** HttpResponse(json.dumps(jsonData), content\_type=**'application/json'**)  
  
**class** SearchView(View):  
 **def** get(self, request):  
 keyWords = request.GET.get(**"q"**, **""**)  
 page = request.GET.get(**"p"**, **"1"**)  
 urlType=request.GET.get(**"t"**,**""**)  
 collType=urlType.replace(common.URL\_PARAM[**"URLDOT"**],**"."**)  
 conn.zincrby(**"hotSearchSet"**, 1, keyWords)  
 topnSearch = common.ReturnTopn(common.FETCH\_NUMBER\_DEF[**'redisTopStart'**],  
 common.FETCH\_NUMBER\_DEF[**'redisTopEnd'**], conn)  
 **try**:  
 page = int(page)  
 **except**:  
 page = 1  
 **if** page>50:  
 page=50  
 startTime = datetime.now()  
 originBody={  
 **"query"**: {  
 **"function\_score"**: {  
 **"query"**: {  
 **"bool"**:{  
 **"must"**:[  
 {  
 **"bool"**: {  
 **"should"**: [  
 {  
 **"multi\_match"**: {  
 **"analyzer"**: **"ik\_smart"**,  
 **"query"**: keyWords,  
 **"fields"**: **"title"**,  
 **"boost"**: 2  
 }  
 },  
 {  
 **"multi\_match"**: {  
 **"analyzer"**: **"ik\_smart"**,  
 **"query"**: keyWords,  
 **"fields"**: **"content"** }  
 }  
 ]  
 }  
 }  
 ]  
 }  
 },  
 **"script\_score"**: {  
 **"script"**: {  
 **"lang"**: **"painless"**,  
 **"source"**: **"int spanTime=params.keyword-doc.create\_date.value.year;"  
 "if(spanTime>0 && spanTime<params.keyword-1)return 1/spanTime;if( doc.is\_index.value)return 1/1.1;return 1;"**,  
 **"params"**: {  
 **"keyword"**: datetime.today().year  
 }  
 }  
 },  
 **"boost\_mode"**: **"multiply"** }  
 },  
 **"from"**: (page - 1) \* common.FETCH\_NUMBER\_DEF[**'pageNum'**],  
 **"size"**: common.FETCH\_NUMBER\_DEF[**'pageNum'**],  
 **"highlight"**: {  
 **"pre\_tags"**: [**'<span class="keyWord">'**],  
 **"post\_tags"**: [**'</span>'**],  
 **"fields"**: {  
 **"title"**: {},  
 **"content"**: {},  
 }  
 }  
 }  
  
 **if** collType:  
 originBody[**"query"**][**"function\_score"**][**"query"**][**"bool"**][**"must"**].append({**"term"**: {  
 **"url\_origin"**: {  
 **"value"**: collType  
 }  
 }})  
 response = client.search(  
 index=**"hfut\_search"**,  
 body=originBody  
 )  
  
 endTime = datetime.now()  
 lastSeconds = (endTime - startTime).total\_seconds()  
 totalNums = response[**"hits"**][**"total"**]  
 pageNums = int(totalNums / common.FETCH\_NUMBER\_DEF[**'pageNum'**])  
 **if** totalNums % common.FETCH\_NUMBER\_DEF[**'pageNum'**]:  
 pageNums += 1  
 hitList = []  
 **for** hit **in** response[**"hits"**][**"hits"**]:  
 hitDict = {}  
 **if "title" in** hit[**"highlight"**]:  
 hitDict[**"title"**] = **""**.join(hit[**"highlight"**][**"title"**])  
 **else**:  
 hitDict[**"title"**] = hit[**"\_source"**][**"title"**]  
 **if "content" in** hit[**"highlight"**]:  
 hitDict[**"content"**] = **""  
 for** item **in** hit[**"highlight"**][**"content"**]:  
 hitDict[**"content"**] += item  
 **if** common.GetStripLabelLen(hitDict[**"content"**]) > common.CONSTVALUE[**'SHOW\_CHAR\_NUM'**]:  
 hitDict[**"content"**] += **"..."  
 break  
 else**:  
 hitDict[**"content"**] = hit[**"\_source"**][**"content"**][:common.CONSTVALUE[**'SHOW\_CHAR\_NUM'**]]  
 **if** len(hit[**"\_source"**][**"content"**]) > common.CONSTVALUE[**'SHOW\_CHAR\_NUM'**]:  
 hitDict[**"content"**] += **'...'** hitDict[**"url"**] = hit[**"\_source"**][**"url"**]  
 dates = hit[**"\_source"**][**"create\_date"**].split(**'T'**)  
 hitDict[**"createDate"**] = dates[0] **if** len(dates) == 2 **and** dates[0] != common.CONSTVALUE[**'NULL\_DATE'**] **and not** hit[**"\_source"**][**"is\_index"**] **else ''** hitDict[**"urlOrigin"**] = jsonData[hit[**"\_source"**][**"url\_origin"**]] \  
 **if** hit[**"\_source"**][**"url\_origin"**] **in** jsonData **else** hit[**"\_source"**][**"url\_origin"**]  
 hitList.append(hitDict)  
 **return** render(request, **"result.html"**, {**"page"**: page,  
 **"hitList"**: hitList,  
 **"keyWords"**: keyWords,  
 **"totalNums"**: totalNums,  
 **"pageNums"**: pageNums,  
 **'topnSearch'**: topnSearch,  
 **"lastSeconds"**: lastSeconds  
 ,**"collList"**:json.dumps(jsonData)  
 ,**"collType"**:urlType})  
  
  
**class** TxtPicSearch(View):  
 **def** CreateZADD(self, keyWords, imgPathList):  
 **for** index, item **in** enumerate(imgPathList):  
 conn.zadd(keyWords, mapping={item: index})  
 conn.expire(keyWords, common.EXPIRE\_TIME)  
  
 **def** get(self, request):  
 keyWords = request.GET.get(**"q"**, **""**)  
 redisKeys = common.GetMd5(keyWords)  
 **if not** conn.exists(redisKeys):  
 response = client.search(  
 index=**"hfut\_search"**,  
 body={  
 **"query"**: {  
 **"bool"**: {  
 **"must"**: [  
 {  
 **"multi\_match"**: {  
 **"analyzer"**: **"ik\_smart"**,  
 **"query"**: keyWords,  
 **"fields"**: **"title"** }  
 }  
 ],  
 **"must\_not"**: [  
 {  
 **"term"**: {  
 **"img\_download"**: {  
 **"value"**: **""** }  
 }  
 }  
 ],  
 **"filter"**: [  
 {**"term"**: {**"is\_index"**: **"false"**}}  
 ]  
 }  
 },  
 **"\_source"**: **"img\_download"**,  
 **"from"**: 0,  
 **"size"**: 100  
 }  
 )  
 imgNameList = **","**.join([item[**'\_source'**][**'img\_download'**] **for** item **in** response[**"hits"**][**"hits"**]]).split(**","**)  
 response = client.search(  
 index=**"hfut\_pic"**,  
 body={  
 **"query"**: {  
 **"terms"**: {  
 **"img\_name"**: imgNameList  
 }  
 },  
 **"from"**: 0,  
 **"size"**: len(imgNameList)  
 }  
 )  
 imgNameList = [item[**'\_source'**][**'img\_path'**] **for** item **in** response[**"hits"**][**"hits"**]]  
 totalNum = len(imgNameList)  
 self.CreateZADD(redisKeys, imgNameList)  
 **else**:  
 totalNum = conn.zcard(redisKeys)  
 resImgList = common.FetchImgPath(redisKeys, common.FETCH\_NUMBER\_DEF[**'start'**],  
 common.FETCH\_NUMBER\_DEF[**'end'**], conn)  
 resImgList = common.ChangeResImg(resImgList, client)  
 pageInfo = {  
 **'start'**: common.FETCH\_NUMBER\_DEF[**'start'**],  
 **'end'**: common.FETCH\_NUMBER\_DEF[**'end'**],  
 **'total'**: totalNum,  
 **"keysToken"**: redisKeys  
 }  
 **return** render(request, **"textPicture.html"**,  
 {**"pageInfo"**: pageInfo,  
 **"filesInfo"**: resImgList,  
 **"keyWords"**: keyWords})

**import** os  
**import** sys  
**import** tensorflow  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 os.environ.setdefault(**"DJANGO\_SETTINGS\_MODULE"**, **"HfutSearch.settings"**)  
 **try**:  
 **from** django.core.management **import** execute\_from\_command\_line  
 **except** ImportError:  
 *# The above import may fail for some other reason. Ensure that the  
 # issue is really that Django is missing to avoid masking other  
 # exceptions on Python 2.* **try**:  
 **import** django  
 **except** ImportError:  
 **raise** ImportError(  
 **"Couldn't import Django. Are you sure it's installed and "  
 "available on your PYTHONPATH environment variable? Did you "  
 "forget to activate a virtual environment?"** )  
 **raise** execute\_from\_command\_line(sys.argv)

**@charset "utf-8"**;  
**html** {  
 \***overflow**: **auto**;  
}  
  
.**logo** {  
 **float**: **left**;  
 **margin-right**: 30**px**;  
 **height**: 33**px**;  
}  
  
*/\*input搜索区域\*/*.**inputArea** {  
 **float**: **left**;  
 **position**: **relative**;  
}  
  
.**inputArea** .**searchInput** {  
 **border**: 1**px solid #bfbfbf**;  
 **padding**: 0 15**px**;  
 **outline**: **none**;  
 **height**: 35**px**;  
 \***line-height**: 35**px**;  
 **width**: 355**px**;  
 **background**: **url**(**../img/inputbg.png**);  
 **font-size**: 14**px**;  
}  
  
.**inputArea** .**searchButton** {  
 **position**: **absolute**;  
 **left**: 382**px**;  
 **top**: 0;  
 \***top**: 1**px**;  
 \***left**: 381**px**;  
 **width**: 106**px**;  
 **height**: 38**px**;  
 **background**: **url**(**../img/btn\_min.png**) **no-repeat**;  
 **border**: **none**;  
 **cursor**: **pointer**;  
}  
  
  
*/\*新增\*/***#header** {  
 **position**: **fixed**;  
 **width**: 100%;  
 **z-index**: 5  
}  
  
**#header** .**inner** {  
 **background**: **#fff**;  
 **border-bottom**: 1**px solid #ebebeb**;  
 **min-width**: 964**px**;  
 **height**: 70**px**;  
 **padding-top**: 30**px**;  
 **padding-left**: 100**px**;  
}  
  
**#bd** {  
 **clear**: **both**;  
 **height**: **auto**;  
 **margin-left**: 120**px**;  
 **min-height**: 500**px**;  
 **position**: **relative**;  
 **width**: 1200**px**}  
  
.**result-upload-wrap** {  
 **position**: **absolute**;  
 **width**: 86**px**;  
 **height**: 35**px**;  
 **font-size**: 18**px**;  
 **border**: 1**px solid #82ac01**;  
 **line-height**: 33**px**;  
 **margin**: 0 **auto**;  
 **text-align**: **center**;  
 **left**: 479**px**;  
 **top**: 0**px**;  
 **color**: **#82ac01**;  
}  
  
.**result-upload-pic** {  
 **position**: **absolute**;  
 **font-size**: 0;  
 **width**: 100%;  
 **height**: 100%;  
 **outline**: 0;  
 **opacity**: 0;  
 **z-index**: 2;  
 **cursor**: **pointer**;  
 **left**: 0**px**;  
}  
  
.**st\_box** {  
 **border**: 1**px solid #e5e5e5**;  
 **box-shadow**: 0 4**px** 16**px rgba**(0, 0, 0, .2);  
 **height**: 130**px**;  
 **padding**: 20**px** 30**px**;  
 **position**: **relative**;  
 **width**: 596**px**;  
 **z-index**: 2;  
 **background**: **white**;  
 **margin-left**: 190**px**;  
}  
  
.**st\_box** .**stBox-drop** {  
 **text-align**: **center**;  
 **height**: 75**px**;  
 **padding**: 20**px** 30**px** 10**px**;  
 **margin-bottom**: 16**px**;  
}  
  
.**stBox-drop-tip** {  
 **display**: **block**;  
 **font-size**: 20**px**;  
 **color**: **#999**}  
  
.**stBox-drop-icon** {  
 **display**: **block**;  
 **background**: **url**(**"../img/addphoto.png"**);  
 **width**: 28**px**;  
 **height**: 28**px**;  
 **position**: **relative**;  
 **margin**: 0 **auto**;  
 **background-position**: -70**px** -2**px**}  
  
.**stBox-close** {  
 **display**: **inline-block**;  
 **background**: **url**(**"../img/addphoto.png"**);  
 **position**: **absolute**;  
 **right**: 18**px**;  
 **bottom**: 13**px**;  
 **z-index**: 2;  
 **background-position**: -20**px** 0;  
 **width**: 18**px**;  
 **height**: 18**px**}  
  
**#tabs-wrap** {  
 **background**: **#fff**;  
 **float**: **none**;  
 **height**: 55**px**;  
 **line-height**: 36**px**;  
 **padding**: 60**px** 0 0 120**px**;  
 **zoom**: 1  
}  
  
.**user\_upload**:**after** {  
 **clear**: **both**;  
 **content**: **''**;  
 **display**: **block**}  
  
.**user\_upload** {  
 **padding-bottom**: 20**px**;  
 \***zoom**: 1  
}  
  
.**user\_upload** .**img\_box** {  
 **float**: **left**;  
 **max-height**: 115**px**;  
 **min-width**: 100**px**;  
 **overflow**: **hidden**;  
 **position**: **relative**;  
 **text-align**: **center**}  
  
.**user\_upload** .**img\_info** {  
 **overflow**: **hidden**;  
 **padding-left**: 20**px**;  
 \***zoom**: 1  
}  
  
.**user\_upload** .**img\_info dt**,  
.**user\_upload** .**img\_info dd** {  
 **color**: **#666**;  
 **line-height**: 18**px**;  
 **padding-bottom**: 13**px**}  
  
.**multiply** {  
 **font-family**: **arial**;  
 **font-size**: 11**px**}  
  
.**like-img-shopping** {  
 **border-top**: 1**px solid #eee**;  
 **overflow**: **hidden**;  
 **padding**: 20**px** 0 25**px**}  
  
.**like-nav** {  
 **padding-bottom**: 15**px**}  
  
**#similar** {  
 **display**: **inline-block**;  
 **font-family**: **'Microsoft YaHei'**, **arial**;  
 **font-size**: 18**px**;  
 **text-align**: **center**;  
 **width**: 90**px**;  
}  
  
.**like-img-list ul** {  
 **overflow**: **hidden**}  
  
.**like-content ul** {  
 **margin-top**: 5**px**}  
  
**ul** {  
 **padding**: 0;  
}  
  
.**like-img-list li** {  
 **float**: **left**;  
 **height**: 188**px**;  
 **margin-right**: 5**px**;  
 **margin-bottom**: 5**px**;  
 **position**: **relative**}  
  
**li** {  
 **list-style**: **none**;  
}  
  
.**like-img-list a** {  
 **display**: **block**;  
 **height**: 100%;  
 **position**: **relative**;  
 **width**: 100%;  
}  
  
.**wfx\_row li img** {  
 **-webkit-transition**: **-webkit-transform** .4**s linear**;  
 **transition**: **transform** .4**s linear**, **-webkit-transform** .4**s linear**;  
}  
  
.**wfx\_row** .**img\_info** {  
 **bottom**: 0;  
 **display**: **none**;  
 **height**: 44**px**;  
 **left**: 0;  
 **overflow**: **hidden**;  
 **position**: **absolute**;  
 **text-overflow**: **ellipsis**;  
 **white-space**: **nowrap**;  
 **width**: 100%;  
}  
  
.**wfx\_row** .**img\_info** .**info\_bg** {  
 **background**: **url**(**../img/stu\_result\_list.png**) **repeat-x**;  
 **\_background-color**: **#000**;  
 **\_filter**: **alpha**(**opacity**=50);  
 **height**: 100%;  
 **width**: 100%;  
 **color**: **#fff**}  
  
.**wfx\_row** .**img\_info** .**info\_desc** {  
 **overflow**: **hidden**;  
 **padding-right**: 30**px**;  
 **text-overflow**: **ellipsis**;  
 **text-shadow**: 0 1**px** 0 **rgba**(0, 0, 0, .5);  
 **white-space**: **nowrap**;  
 **padding-left**: 5**px**;  
 **padding-top**: 5**px**;  
}  
  
.**wfx\_row** .**img\_info** .**info\_size** {  
 **bottom**: 5**px**;  
 **padding-left**: 5**px**;  
 **text-shadow**: 0 1**px** 0 **rgba**(0, 0, 0, .5)  
}  
  
.**wfx\_row** .**img\_info** .**info\_st** {  
 **background**: **url**(**../img/stu\_result\_search.png**) **no-repeat**;  
 **background-size**: 16**px** 16**px**;  
 **bottom**: 5**px**;  
 **display**: **block**;  
 **height**: 16**px**;  
 **position**: **absolute**;  
 **right**: 8**px**;  
 \***right**: 14**px**;  
 **width**: 16**px**}  
  
.**high-quality-load** {  
 **display**: **block**;  
 **position**: **relative**;  
 **top**: 26%;  
 **color**: **#333**;  
 **margin-top**: -20**px**;  
 **text-align**: **center**;  
 **padding**: 8**px** 0 32**px**;  
 **font-size**: 16**px**;  
 **width**: 230**px**;  
 **left**: 30%;  
}  
  
**#gotop** {  
 **display**: **none**;  
 **z-index**: 1000  
}  
  
.**gotop a** {  
 **background**: **url**(**../img/return\_top.png**) **no-repeat** 0 0;  
 **background-image**: **url**(**../img/return\_top.png**);  
 **background-size**: 36**px** 36**px**;  
 **display**: **block**;  
 **height**: 36**px**;  
 **overflow**: **hidden**;  
 **text-indent**: -999**em**;  
 **width**: 36**px**}

**var *picSearchVar*** = {  
 **camera**: $(**"#camera"**),  
 **searchInput**: $(**".searchInput"**),  
 *//share* **inputText**: $(**"#input\_text"**),  
 **searchButton**: $(**'.searchButton'**),  
 **allPos**: $(**"html"**),  
 **stBoxChildElem**: $(**"#stBox \*"**),  
 **stBoxColse**: $(**".stBox-close"**),  
 **stBox**: $(**"#stBox"**),  
 **stBoxChild**: $(**"#stBoxChild"**),  
 **uploadTip**: $(**".st\_box .high-quality-load"**),  
 **picHide**: $(**"#pic\_hide"**)*//result*};  
**var *PAGE\_TYPE*** = {  
 **INDEX**: **"index"**,  
 **RESULT**: **"result"**};  
**var *UPLOAD\_TYPE*** = {  
 **SUBMIT**: **"SUBMIT"**,  
 **DRAG**: **"DRAG"**,  
 **URL**: **"URL"**,  
 **LINK**: **"LINK"**,  
 **NONE**: **"NONE"**};  
**var *dragTimer*** = **null**;  
  
  
$(***document***).keydown(**function** (event) {  
 event.**keyCode** == 13 && ***picSearchVar***.**searchButton**.click();  
})  
  
**function** *Check*() {  
 **var** filePath = ***picSearchVar***.**inputText**.val();  
 filePath = filePath.replace(/(^\s\*)|(\s\*$)/g, **''**);  
 **if** (filePath === 0)  
 **return false**;  
 **if** (filePath.length < 7 || (filePath.substring(0, 7) != **'http://'** && filePath.substring(0, 8) != **'https://'**)) {  
 alert(**"请输入以http://开头的网络图片地址！"**);  
 **return false**;  
 }  
 **var** filePath = filePath.substring(8);  
 **if** (filePath.indexOf(**'/'**) < 0) {  
 alert(**"请输入正确的网络图片地址！"**);  
 **return false**;  
 }  
 filePath = filePath.substring(filePath.lastIndexOf(**'.'**) + 1, filePath.length);  
 filePath = filePath.toLowerCase();  
 **if** (filePath.length < 5 && filePath != **'jpg'** && filePath != **'png'** && filePath != **'jpeg'**) {  
 alert(**"抱歉，目前不支持此文件格式，请确认图片是JPG、PNG、JPEG格式。"**);  
 **return false**;  
 }  
 **return true**;  
}  
  
**function** CheckUploadFile(target) {  
  
 **try** {  
 **var** fileSize = 0;  
 **var** fileType = **""**;  
 fileSize = target.files[0].size;  
 fileType = target.files[0].type;  
  
 **if** (fileSize > 5 \* 1000 \* 1000) {  
 alert(**"您上传的文件过大，请选择小于5M的文件上传。"**);  
 **return false**;  
 }  
 **if** (fileType.indexOf(**'image'**) < 0) {  
 alert(**"抱歉，目前不支持此文件格式，请确认图片是JPG、PNG、JPEG格式。"**);  
 **return false**;  
 }  
 } **catch** (e) {  
 }  
 **return true**;  
  
}  
  
**function** UploadFileChange(target, pageType) {  
  
 **if** (CheckUploadFile(target)) {  
 **if** (target.files.length) {  
 UploadBackgroud(pageType);  
 UploadImage(target.files[0], UPLOAD\_TYPE.SUBMIT);  
 }  
 }  
}  
  
**function** UploadImage(file, uploadType) {  
 **var** fd = **new** FormData();  
 fd.append(**'file'**, file);  
 **var** csrfToken = $(**"[name='csrfmiddlewaretoken']"**).val();  
 $.ajax({  
 url: **"/uploadPicture/"** + uploadType + **'/'**,  
 type: **"POST"**,  
 data: fd,  
 processData: **false**,  
 contentType: **false**,  
 beforeSend: **function** (xhr, settings) {  
 xhr.setRequestHeader(**"X-CSRFToken"**, csrfToken);  
 },  
 success: **function** (imgInfo) {  
 location.href = **"/searchPicture/"** + imgInfo.uploadType + **"/?"** + **"query="** + imgInfo.imgName;  
 }  
 });  
};  
  
  
**function** DragWindow(isShow, showTime, pageType) {  
  
 pageType == PAGE\_TYPE.INDEX && (isShow ? (picSearchVar.camera.hide(),  
 picSearchVar.searchInput.attr(**'placeholder'**, **'在此处粘贴图片网址'**)) : (picSearchVar.camera.show(),  
 picSearchVar.searchInput.attr(**'placeholder'**, **'进行文本搜索'**)));  
 isShow ? picSearchVar.stBox.show(showTime) : picSearchVar.stBox.hide(showTime);  
}  
  
**function** DragClass(isShow) {  
 picSearchVar.stBoxChild.removeClass(isShow ? **"color\_less"** : **"color\_more"**);  
 picSearchVar.stBoxChild.addClass(isShow ? **"color\_more"** : **"color\_less"**);  
}  
  
**function** ClearTimer() {  
 dragTimer && (clearTimeout(**this**.dragTimer), dragTimer = **null**);  
}  
  
**function** UploadBackgroud(pageType) {  
 pageType == PAGE\_TYPE.RESULT && picSearchVar.stBox.show();  
 picSearchVar.picHide.hide();  
 picSearchVar.uploadTip.show();  
}  
  
**function** DragStrategy() {  
 **try** {  
  
 **var** isDrag = **false**;  
 **if** (**typeof** DragStrategy.\_initialized == **"undefined"**) {  
  
 $.extend(DragStrategy.prototype, {  
 BindContainEvent: **function** (pageType) {  
  
 picSearchVar.stBoxChild.bind(**"dragover"**, **function** (event) {  
  
 pageType == PAGE\_TYPE.INDEX && DragClass(**true**);  
  
 }).bind(**"dragleave"**, **function** (event) {  
  
 pageType == PAGE\_TYPE.INDEX && DragClass(**false**);  
  
 }).bind(**"drop"**, **function** (event) {  
 event.stopPropagation();  
 event.preventDefault();  
 **var** dtf = event.originalEvent.dataTransfer;  
 **if** (!CheckUploadFile(dtf)) {  
 picSearchVar.stBoxChild.addClass(**"color\_less"**);  
 pageType == PAGE\_TYPE.RESULT && DragWindow(**false**, 0, pageType);  
 **return false**;  
 }  
 isDrag = **true**;  
 **var** files = dtf.files;  
 **if** (files.length > 0) {  
 UploadImage(files[0], UPLOAD\_TYPE.DRAG);  
 UploadBackgroud(pageType);  
 }  
 isDrag = **false**;  
 });  
 },  
  
  
 BindPicHtmlEvent: **function** (pageType) {  
 $(picSearchVar.allPos).bind(**"dragover"**, **function** (event) {  
  
 pageType == PAGE\_TYPE.RESULT && ClearTimer(),  
 DragWindow(**true**, 0, pageType), event.preventDefault();  
  
 }).bind(**'dragenter'**, **function** (event) {  
  
 pageType == PAGE\_TYPE.RESULT &&  
 (ClearTimer(), DragWindow(**true**, 0, pageType)), event.preventDefault();  
  
 }).bind(**"dragleave"**, **function** (event) {  
  
 pageType == PAGE\_TYPE.RESULT && (dragTimer = setTimeout(**function** () {  
 DragWindow(**false**, 0, pageType);  
 **this**.dragTimer = **null**;  
 }, 100)), event.preventDefault();  
  
 }).bind(**"drop"**, **function** (event) {  
  
 !isDrag && pageType == PAGE\_TYPE.RESULT &&  
 DragWindow(**false**, 0, pageType), event.preventDefault();  
  
 });  
 },  
  
 BindEvent: **function** (pageType) {  
 picSearchVar.allPos.click(**function** (event) {  
 **var** target = $(event.target);  
 (target.is(picSearchVar.camera) && !picSearchVar.stBox.is(**':visible'**) &&  
 (DragWindow(**true**, 500, pageType), **true**)) ||  
 ((picSearchVar.stBox.is(**':visible'**) && ((!target.is(picSearchVar.stBoxChildElem) &&  
 !target.is(picSearchVar.stBox) && !target.is(picSearchVar.searchInput) && !target.is(picSearchVar.searchButton))  
 || target.is(picSearchVar.stBoxColse))) &&  
 DragWindow(**false**, 500, pageType))  
 });  
 },  
 init: **function** (pageType) {  
 pageType == PAGE\_TYPE.INDEX && **this**.BindEvent(pageType);  
 **this**.BindPicHtmlEvent(pageType);  
 **this**.BindContainEvent(pageType);  
 }  
 });  
 DragStrategy.\_initialized = **true**;  
 }  
 } **catch** (e) {  
  
 }  
}

<!DOCTYPE **html** >  
<**html xmlns="http://www.w3.org/1999/xhtml"**>  
{% **load staticfiles** %}  
<**head**>  
 <**meta http-equiv="X-UA-Compatible" content="IE=emulateIE11"**/>  
 <**meta http-equiv="Content-Type" content="text/html; charset=UTF-8"**/>  
 <**title**>合肥工业大学 搜索引擎</**title**>  
 <**link href="**{% **static 'css/style.css'** %}**" rel="stylesheet" type="text/css"**/>  
 <**link href="**{% **static 'css/searchPicture.css'** %}**" rel="stylesheet" type="text/css"**/>  
</**head**>  
<**body**>  
<**div id="header"**>  
 <**div id="hd" class="inner"**>  
  
 <**a href="/"**>  
 <**div class="logo"**></**div**>  
 </**a**>  
 <**div class="inputArea"**>  
 <**input type="text" id="input\_text" class="searchInput" placeholder="拖拽图片到此或在此处粘贴图片网址"**/>  
 <**input type="button" class="searchButton" onclick="***AddSearch*()**"**/>  
 <**div class="result-upload-wrap"**>  
  
 <**input type="file" id="uploadBt" name='imgFile' class="result-upload-pic" value="上传图片"  
 accept="image/\*" onchange="***UploadFileChange*(**this**,***PAGE\_TYPE***.**RESULT**)**"**>  
 {% **csrf\_token** %}  
 <**span**>本地上传</**span**>  
 </**div**>  
  
 </**div**>  
 <**div class="st\_box" id="stBox" style="display**: **none"**>  
 <**div class="high-quality-load" style="display**:**none"**>  
 <**span**><**i**></**i**><**i**></**i**></**span**>  
 正在上传文件...  
 </**div**>  
 <**div id="pic\_hide"**>  
 <**div id="stBoxChild" class="stBox-drop color\_less"**>  
 <**span class="stBox-drop-tip"**>拖拽图片到这里</**span**>  
 <**i class="stBox-icon stBox-drop-icon"**></**i**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
  
</**div**>  
<**div id="tabs-wrap"**></**div**>  
<**div id="bd"**>  
 <**div class="user\_upload" id="userUpload"**>  
 <**div class="img\_box"**><**img src="**{% **static ''** %}{{ **userUpload**.**imgUrl** }}**" alt="您上传的图"  
 style="width**:180**px**;**height**:120**px**; **background-color**: **#eeeeee"**></**div**>  
 <**dl class="img\_info"**>  
 <**dt class="img\_title" id="bestGuess"**>最佳猜测：  
 <**a target="\_blank" href="http://**{{ **userUpload**.**infoGuess** }}**"**>  
 {% **if userUpload**.**title** %}{{ **userUpload**.**title** }}{% **else** %}无标题{% **endif** %}</**a**>  
 </**dt**>  
 <**dd**>图片尺寸：{{ **userUpload**.**width** }}**&nbsp;**<**span class="multiply"**>X</**span**>**&nbsp;**{{ **userUpload**.**height** }}</**dd**>  
 </**dl**>  
 </**div**>  
 <**div class="like-img-shopping"**>  
 <**div class="like-nav"**>  
 <**span id="similar"**>相似图片</**span**>  
 </**div**>  
 <**div class="like-content"**>  
 <**div class="like-wrap" id="likeImg" style="display**: **block**;**"**>  
 <**div class="like-img-list"**>  
 <**div class="wfx\_group"**>  
  
 {% **if userUpload**.**total** %}<**span style="font-size**: **medium"**>第1页</**span**>{% **endif** %}  
 <**ul class="wfx\_row" style="width**:1200**px**;**"**>  
 {% **for imgPath**, **imgUrl**, **imgSize**,**imgTitle in filesInfo** %}  
 <**li style="width**:230**px**;**height**:150**px**; **background-color**: **#eeeeee"**>  
 <**a target="\_blank" href="http://**{{ **imgUrl** }}**" class="img\_box" data-copr="1"  
 data-pos="0"**>  
 <**img class="main\_img" src="**{% **static ''** %}{{ **imgPath** }}**"  
 style="width**:100%;**height**:100%**"**>  
 </**a**>  
 <**div class="img\_info"**>  
 <**div class="info\_bg"**>  
 <**div class="info\_desc"**>{{ **imgTitle** }}</**div**>  
 <**div class="info\_size"**><**span**>{{ **imgSize** }}</**span**>  
 <**a href="javascript:;" data-href="**{{ **imgPath** }}**" class="info\_st"**></**a**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**li**>  
 {% **endfor** %}  
  
 </**ul**>  
 </**div**>  
 </**div**>  
  
 <**div id="footLoad" class="high-quality-load" style="display**:**none**;**margin-top**: 1**px**;**left**: 40%;**"**>  
 <**span**><**i**></**i**><**i**></**i**></**span**>  
 加载中...  
 </**div**>  
 <**div id="footLoadOver" class="high-quality-load" style="display**:**none**;**margin-top**: 1**px**;**left**: 40%;**"**>  
 已经加载完毕  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 <**div class="gotop" id="gotop" title="返回顶部" style="left**: 1300**px**; **position**: **fixed**; **bottom**: 60**px**; **display**: **none**;**"**>  
 <**a href="javascript:;" class="btn\_gotop"**></**a**></**div**>  
</**div**>  
  
<**div id="foot"**>合肥工业大学</**div**>  
</**body**>  
<**script type="text/javascript" src="**{% **static 'js/jquery.js'** %}**"**></**script**>  
<**script type="text/javascript" src="**{% **static 'js/searchPicture.js'** %}**"**></**script**>  
<**script type="text/javascript" src="**{% **static 'js/global.js'** %}**"**></**script**>  
<**script type="text/javascript"**>  
  
 **var *searchPhotoUrl*** = **"**{% **url 'searchPicture' 'URL'** %}**"**, ***isAchieve*** = **true**, ***isInit*** = **true**,  
 ***initStart*** ={{ **userUpload**.**start** }}, ***initEnd*** ={{ **userUpload**.**end** }}, ***total*** = {{ **userUpload**.**total** }};  
  
 **function** *AddSearch*() {  
 *Check*() && (***location***.**href** = ***searchPhotoUrl*** + **"?query="** + *encodeURIComponent*(***picSearchVar***.**inputText**.val()));  
 }  
  
 $(**"a.btn\_gotop"**).bind(**"click"**, **function** (e) {  
 $(***window***).**scrollTop**(0);  
 $(**'#gotop'**).**fadeOut**(300);  
 });  
  
 **try** {  
 (**new** *DragStrategy*()).init(***PAGE\_TYPE***.**RESULT**);  
  
 $(**'#likeImg .wfx\_group'**).on(**'mouseover'**, **'li'**, **function** (e) {  
 $(**this**).children(**'.img\_info'**).**fadeIn**(0);  
 }).on(**'mouseout'**, **'li'**, **function** (e) {  
 $(**this**).children(**'.img\_info'**).**fadeOut**(0);  
 });  
  
 $(**'#likeImg .wfx\_group'**).on(**'click'**, **'.img\_info a'**, **function** (e) {  
 ***location***.**href** = **"/searchPicture/"** + ***UPLOAD\_TYPE***.**LINK** + **"/?"** + **"query="** + *encodeURIComponent*($(**this**).attr(**'data-href'**));  
 })  
  
 $(***window***).scroll(**function** () {  
 (***isInit*** && $(***window***).**scrollTop**(0)) || ((($(***window***).**scrollTop**() + $(***window***).height() >= $(***document***).height() - 50)  
 && (***initEnd*** + 1 < ***total***) && ***isAchieve***) ? ($(**'#footLoad'**).show(), ***isAchieve*** = **false**,  
 *ajaxLoad*(***initEnd*** + 1, ***initEnd*** + ***initEnd*** - ***initStart*** + 1)) : (!(***initEnd*** + 1 < ***total***) && $(**'#footLoadOver'**).show()));  
 $(***window***).**scrollTop**() > 0.5 \* $(***window***).height() ? $(**'#gotop'**).**fadeIn**(300) : $(**'#gotop'**).**fadeOut**(300);  
 ***isInit*** = **false**;  
 });  
  
 } **catch** (e) {  
 }  
  
 **function** *addElemHtm*(pageIndex, imgList) {  
 **var** htmlCode = **'<span style="font-size: medium">第'** + pageIndex + **'页</span>'** +  
 **'<ul class="wfx\_row" style="width:1200px;">'  
 for** (**itemTotal in** imgList) {  
 htmlCode += **'<li style="width:230px;height:150px; background-color: #eeeeee">\n'** +  
 **'<a target="\_blank" href="http://'** + imgList[**itemTotal**][1] + **'"class="img\_box" data-copr="1" data-pos="0">'** +  
 **'<img class="main\_img" src="**{% **static ''** %}**'** + imgList[**itemTotal**][0] + **'"style="width:100%;height:100%">'** +  
 **'</a><div class="img\_info">\n'** +  
 **'<div class="info\_bg">\n'** +  
 **'<div class="info\_desc">'** + imgList[**itemTotal**][3] + **'</div>\n'** +  
 **'<div class="info\_size"><span>'** + imgList[**itemTotal**][2] + **'</span>\n'** +  
 **'<a href="javascript:;"data-href="'** + imgList[**itemTotal**][0] + **'"class="info\_st"></a>\n'** +  
 **' </div></div></div></li>'** }  
 **return** htmlCode += **'</ul>'** };  
  
 **function** *ajaxLoad*(start, end) {  
 **urlParams** = **"?imgToken="** + **"**{{ **userUpload**.**imgName** }}**"** +  
 **"&"** + **'start='** + start + **'&end='** + end;  
 $.**ajax**({  
 **url**: **"/ajaxQuest/"** + **urlParams**,  
 **processData**: **false**,  
 **contentType**: **"application/json; charset=utf-8"**,  
 **dataType**: **"json"**,  
 success: **function** (jsonInfo) {  
 **if** (jsonInfo.imgList.**length**) {  
 **var** htmlCode = *addElemHtm*(jsonInfo.pageIndex, jsonInfo.imgList)  
 $(**".wfx\_group"**).append(htmlCode)  
 ***initStart*** = start;  
 ***initEnd*** = end;  
 } **else** {  
 ***total*** = start;  
 $(**'#footLoadOver'**).show()  
 }  
 $(**'#footLoad'**).hide()  
 ***isAchieve*** = **true**;  
 }  
 });  
 };  
</**script**>  
</**html**>

{% **extends 'searchBase.html'** %}  
{% **load staticfiles** %}  
{% **block custom\_css** %}  
 <**link href="**{% **static 'css/searchPicture.css'** %}**" rel="stylesheet" type="text/css"**/>  
{% **endblock** %}  
{% **block searchFunction** %}  
 <**input type="button" class="searchButton" onclick="***txtPicSearch*()**"**/>  
{% **endblock** %}  
{% **block chooseType** %}  
 <**li**><**a onclick="***txtSearch*()**"**>网页</**a**></**li**>  
 <**li class="txt-g-hd-cur"**><**a**>图片</**a**></**li**>  
{% **endblock** %}  
{% **block bodyConent** %}  
 <**div id="txt-bd"**>  
 <**div class="like-img-shopping"**>  
 <**div class="like-content"**>  
 <**div class="like-wrap" id="likeImg" style="display**: **block**;**"**>  
 <**div class="like-img-list"**>  
 <**div class="wfx\_group"**>  
 <**ul class="wfx\_row" style="width**:1200**px**;**"**>  
 {% **for imgPath**, **imgUrl**, **imgSize**,**imgTitle in filesInfo** %}  
 <**li style="width**:230**px**;**height**:150**px**; **background-color**: **#eeeeee"**>  
 <**a target="\_blank" href="http://**{{ **imgUrl** }}**" class="img\_box" data-copr="1"  
 data-pos="0"**>  
 <**img class="main\_img" src="**{% **static ''** %}{{ **imgPath** }}**"  
 style="width**:100%;**height**:100%**"**>  
 </**a**>  
 <**div class="img\_info"**>  
 <**div class="info\_bg"**>  
 <**div class="info\_desc"**>{{ **imgTitle** }}</**div**>  
 <**div class="info\_size"**><**span**>{{ **imgSize** }}</**span**>  
 <**a href="javascript:;" data-href="**{{ **imgPath** }}**"  
 class="info\_st"**></**a**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**li**>  
 {% **endfor** %}  
 </**ul**>  
 </**div**>  
 </**div**>  
  
 <**div id="footLoad" class="high-quality-load" style="display**:**none**;**margin-top**: 1**px**;**left**: 40%;**"**>  
 <**span**><**i**></**i**><**i**></**i**></**span**>  
 加载中...  
 </**div**>  
 <**div id="footLoadOver" class="high-quality-load" style="display**:**none**;**margin-top**: 1**px**;**left**: 40%;**"**>  
 已经加载完毕  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 <**div class="gotop" id="gotop" title="返回顶部" style="left**: 1300**px**; **position**: **fixed**; **bottom**: 60**px**; **display**: **none**;**"**>  
 <**a href="javascript:;" class="btn\_gotop"**></**a**></**div**>  
 </**div**>  
{% **endblock** %}  
{% **block custom\_js** %}  
 <**script type="text/javascript" src="**{% **static 'js/searchPicture.js'** %}**"**></**script**>  
 <**script type="text/javascript"**>  
 **var *searchUrl*** = **"**{% **url 'search'** %}**"**, ***keyWords*** = **"**{{ **keyWords** }}**"**,  
 ***searchPhotoUrl*** = **"**{% **url 'searchPicture' 'URL'** %}**"** , ***isAchieve*** = **true**, ***isInit*** = **true**, ***txtPicSear*** = **"**{% **url 'txtPicSearch'** %}**"**, ***initStart*** ={{ **pageInfo**.**start** }},