In [72]:

```
import csv
import json
```

In [73]:

```
def replace_dot(field, alt='_'):
    new_s = ''
    splitted = field.split(".")
    count = 0
    for s in splitted:
        count += 1
        if count == 1:
            new_s = s
        else:
            new_s = new_s + alt + s
    return new_s
```

In [74]:

```
def fields_update(k, v, fields, values, daddy_key):
    new_k = replace_dot(k)
    if daddy_key:
        curr = daddy_key + "." + new_k
    else:
        curr = new_k

    if type(v) is dict:
        dict_recur(v, fields, values, curr)
    elif type(v) is list:
        array_recur(v, fields, values, curr)
    else:
        fields.append(curr)
        values.append(v)
```

In [75]:

```
def array_recur(l, fields, values, daddy_key=''):
    if not isinstance(l, list):
        return
    for i in range(len(l)):
        fields_update(str(i), l[i], fields, values, daddy_key)
```

In [76]:

```
def dict_recur(d, fields, values, daddy_key=''):
    if not isinstance(d, dict):
        return
    for k,v in d.items():
        fields_update(k, v, fields, values, daddy_key)
```

In [77]:

```
def convert_to_metric(v):
    temp = v.split(')')
    if len(temp) > 1:
        temp2 = temp[0].split('(')
        return temp2[1]
    return v
```

In [78]:

```
def flatten_dict(data):
    fields = []
    values = []
    dict_recur(data, fields, values)
```

```
    new_dict = {}
    for k,v in zip(fields, values):
        if k == 'height' or k == 'weight':
            v = convert_to_metric(v)
        new_dict[k] = v
    return new_dict, fields
```

In [79]:

```
def add_fields(all_fields, fields):
    flag = False
    for f in fields:
        for af in all_fields:
            if f == af:
                flag = True
        if (flag == False):
            all_fields.append(f)
        flag = False
```

In [80]:

```
def add_values(row, header):
    rowie = {}
    for h in header:
        rowie[h] = ''
    for k,v in row.items():
        rowie[k] = v
    return rowie
```

In [81]:

```
def json_to_tsv(path):
    all_fields = []
    with open(path) as json_file:
        data = json.load(json_file)
        old_entries = []
        for r in data:
            old_row, fields = flatten_dict(r)
            add_fields(all_fields, fields)
            old_entries.append(old_row)
        entries_list = []
        for e in old_entries:
            row = add_values(e, all_fields)
            entries_list.append(row)
        entries_list.sort(key = lambda e: e['name'])

    with open('new.tsv', "w", newline='') as csv_file:
        writer = csv.DictWriter(csv_file, fieldnames=row.keys(), delimiter='\t', quotech
ar='|', quoting=csv.QUOTE_MINIMAL)
        writer.writeheader()
        for row in entries_list:
            writer.writerow(row)
```

In [82]:

```
if __name__ == '__main__':
    json_to_tsv("pokedex.json")
```