

So, we want to search this dictionary to find all Oxygens
We want to then look at this Oxygen and all magnitude distances to all other atoms and we want to see if (any of them meet the required distance (2.75) && if they are of the correct species)
then we might want to see if (this atom has a specific number (2) of those neighboring atoms) and then
classify it as a water.

```
search dict. to find all H-O pairs. hoPairs = [] for k in atomdict: if "tags H and O" in k: hoPairs[k] = atomdict[k]
weed out ones greater than mag. distance of 1.00. for k in hoPairs: if hoPairs[k][2][0] > H2O_L: hoPairs.pop(k)
for all in the dict put them in a master dict if atomnum == atomnum with key equal to atomnum.
hopairsByAtom = {} HOnums = [] dictList = []; for k in hoPairs:
HOnums.append(hoPairs[k][3][0]); hoSet = set(HOnums);
for n in hoSet: hopairsByAtom[n]=[]; hopairsByAtom = {dictList: [] for dictList in range(len(hoSet))}
for n in hoSet: for k in hoPairs: if n == hoPairs[k][3][0]:
hopairsByAtom[n]=dictList.append(k)
H2ODict = {}
for k in hopairsByAtom: if len(hopairsByAtom[k]) >= 2: H2ODict[k]=True
```

Search dict for enties with (H(i)-O(j) && same atomnum): so, determine i's for H-O pairs and then create a set of those number's
to determine all unique H atom's in question. Then, with that set for each num in the set, run through the H-O pairs list again
and create a list for each unique H, of all pairs (H-O) of which that H is a part. Now, you will have a list of the distances
between that H atom and all oxygens, for each unique H atom. Now, you can apply the distance and quantity conditons and remove
all oxygens with mag dist. >= 1 away from the H and then count number of elements in the list to find out how many pairs of
there are satisfying those conditions.

```
atomdict = [[vecTo],[vecfrom],[mag],i,j,iTag,jTag]
```

now, I need to take all the H2O pairs from H2O dict and compare to atomdict and look fo O's from the slab

So, if the Oxygen index from the H2O pairs in H2O dict is used in any of the pairs in atom dict, create an new
list containing all pairs except those (those removed pairs will be the H2O pairs as in H2ODict)

I want to iterate over all values in a particulatr time step of the dataframe and for each value I want to compare
its index to the indexs of the O's from H2O's which are contained in the list OfromH2O returned from h2oSearch()