

---

# **pyOBabel Documentation**

***Release 0.1.1***

**Rajendra Kumar**

**Aug 03, 2018**



---

## Contents

---

<b>1</b>	<b>Why another Python interface</b>	<b>3</b>
<b>2</b>	<b>Please cite the original publication of the openbabel:</b>	<b>5</b>
<b>3</b>	<b>Contents</b>	<b>7</b>
3.1	Summary of available classes . . . . .	7
3.2	Summary of available classes and functions . . . . .	7
<b>4</b>	<b>Indices and tables</b>	<b>33</b>
	<b>Python Module Index</b>	<b>35</b>



It is a Python interface for .

It is not the official Python interface of openbabel and may lack many functionality of openbabel. If you are looking for official Python, Please follow this .



# CHAPTER 1

---

## Why another Python interface

---

The official Python interface to openbabel was built using `SWIG`. However, I used `pybind11` to built the Python interface. It gives flexibility in building Python interface such that the obtained Python interface can be used in pythonic way.





## CHAPTER 2

---

Please cite the original publication of the openbabel:

---

- **O'Boyle et al (2011)** Open Babel: An open chemical toolbox Journal of Cheminformatics 2011 3:33 <https://doi.org/10.1186/1758-2946-3-33> .



### 3.1 Summary of available classes

<i>OBMol</i>	Molecule Class
<i>OBAtom</i>	
<i>OBBond</i>	
<i>OBResidue</i>	
<i>vector3</i>	Represents a vector in 3-dimensional real space.
<i>OBBitVec</i>	
<i>OBConversion</i>	
<i>OBBuilder</i>	
<i>OBForceField</i>	
<i>OBFFConstraints</i>	
<i>OBError</i>	
<i>OBMessageHandler</i>	

### 3.2 Summary of available classes and functions

#### 3.2.1 Classes

##### OBMol

<i>OBMol</i>	Molecule Class
<i>OBMol.atoms</i>	List of the atoms – <i>OBAtom</i> .
<i>OBMol.bonds</i>	List of the bonds – <i>OBBond</i> .
<i>OBMol.residues</i>	List of the residues - <i>OBResidue</i> .

## Documentation

### class OBMol

Molecule Class

The most important class in Open Babel is OBMol, or the molecule class. The OBMol class is designed to store all the basic information associated with a molecule, to make manipulations on the connection table of a molecule facile, and to provide member functions which automatically perceive information about a molecule. A guided tour of the OBMol class is a good place to start.

### Example

```
import pyOBabel as ob

# An OBMol class can be instantiated by
mol = OBMol()

conv = ob.OBConversion()
conv.SetInFormat("MOL2")
conv.ReadFile(mol, filename)

# Print index and atomic number
for atom in mol.atoms:
    print(atom.GetIndex(), atom.GetAtomicNum())

# Print bond order, first atom index and second atom index
for bond in mol.bonds:
    print(bond.GetBondOrder(), bond.GetBeginAtom().GetIndex(),
          bond.GetEndAtom().GetIndex())
```

**AddAtom** (self: pyOBabel.pyOBabel.OBMol, atom: OpenBabel::OBAtom, forceNewId: bool=False) → bool

**AddBond** (self: pyOBabel.pyOBabel.OBMol, arg0: OpenBabel::OBBond) → bool

**AddBondBetweenAtoms** (self: pyOBabel.pyOBabel.OBMol, beginIdx: int, endIdx: int, order: int, flags: int=0, insertpos: int=-1) → bool

**AddConformerCoords** (self: pyOBabel.pyOBabel.OBMol, arg0: list) → None

**AddHydrogens** (\*args, \*\*kwargs)

Overloaded function.

1. AddHydrogens(self: pyOBabel.pyOBabel.OBMol, polaronly: bool=False, correctForPH: bool=False, pH: float=7.4) -> bool
2. AddHydrogens(self: pyOBabel.pyOBabel.OBMol, arg0: OpenBabel::OBAtom) -> bool

**AddNewHydrogens** (self: pyOBabel.pyOBabel.OBMol, whichHydrogen: pyOBabel.pyOBabel.HydrogenType, correctForPH: bool=False, pH: float=7.4) → bool

**AddNonPolarHydrogens** (self: pyOBabel.pyOBabel.OBMol) → bool

**AddPolarHydrogens** (self: pyOBabel.pyOBabel.OBMol) → bool

**AddResidue** (self: pyOBabel.pyOBabel.OBMol, arg0: OpenBabel::OBResidue) → bool

**Align** (self: pyOBabel.pyOBabel.OBMol, arg0: OpenBabel::OBAtom, arg1: OpenBabel::OBAtom, arg2: pyOBabel.pyOBabel.vector3, arg3: pyOBabel.pyOBabel.vector3) → None

**AreInSameRing** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: OpenBabel::OBAtom, *arg1*: OpenBabel::OBAtom) → int

**AssignSpinMultiplicity** (*self*: pyOBabel.pyOBabel.OBMol, *NoImplicitH*: bool=False) → bool

**AssignTotalChargeToAtoms** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → bool

**AutomaticFormalCharge** (*self*: pyOBabel.pyOBabel.OBMol) → bool

**AutomaticPartialCharge** (*self*: pyOBabel.pyOBabel.OBMol) → bool

**BeginModify** (*self*: pyOBabel.pyOBabel.OBMol) → None

**Center** (*self*: pyOBabel.pyOBabel.OBMol) → None

**CenterOfConformer** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → pyOBabel.pyOBabel.vector3

**Clear** (*self*: pyOBabel.pyOBabel.OBMol) → bool

**ConnectTheDots** (*self*: pyOBabel.pyOBabel.OBMol) → None

**ContigFragList** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: List[List[int]]) → None

**ConvertDativeBonds** (*self*: pyOBabel.pyOBabel.OBMol) → bool

**ConvertZeroBonds** (*self*: pyOBabel.pyOBabel.OBMol) → bool

**CorrectForPH** (*self*: pyOBabel.pyOBabel.OBMol, *pH*: float=7.4) → bool

**DeleteAtom** (*self*: pyOBabel.pyOBabel.OBMol, *atom*: OpenBabel::OBAtom, *destroyAtom*: bool=False) → bool

**DeleteBond** (*self*: pyOBabel.pyOBabel.OBMol, *bond*: OpenBabel::OBBond, *destroyBond*: bool=False) → bool

**DeleteConformerByIndex** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → None

**DeleteHydrogen** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: OpenBabel::OBAtom) → bool

**DeleteHydrogensAll** (*self*: pyOBabel.pyOBabel.OBMol) → bool

**DeleteHydrogensFromAtom** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: OpenBabel::OBAtom) → bool

**DeleteNonPolarHydrogens** (*self*: pyOBabel.pyOBabel.OBMol) → bool

**DeletePolarHydrogens** (*self*: pyOBabel.pyOBabel.OBMol) → bool

**DeleteResidue** (*self*: pyOBabel.pyOBabel.OBMol, *residue*: OpenBabel::OBResidue, *destroyResidue*: bool=False) → bool

**Empty** (*self*: pyOBabel.pyOBabel.OBMol) → bool

**EndModify** (*self*: pyOBabel.pyOBabel.OBMol, *nukePerceivedData*: bool=True) → None

**FindAngles** (*self*: pyOBabel.pyOBabel.OBMol) → None

**FindChiralCenters** (*self*: pyOBabel.pyOBabel.OBMol) → None

**FindLSSR** (*self*: pyOBabel.pyOBabel.OBMol) → None

**FindRingAtomsAndBonds** (*self*: pyOBabel.pyOBabel.OBMol) → None

**FindSSSR** (*self*: pyOBabel.pyOBabel.OBMol) → None

**FindTorsions** (*self*: pyOBabel.pyOBabel.OBMol) → None

**GetAngle** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: OpenBabel::OBAtom, *arg1*: OpenBabel::OBAtom, *arg2*: OpenBabel::OBAtom) → float

**GetConformerCoordsByIndex** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → list

**GetConformerEnergies** (*self*: pyOBabel.pyOBabel.OBMol) → List[float]  
**GetConformerEnergy** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → float  
**GetConformersAllCoords** (*self*: pyOBabel.pyOBabel.OBMol) → list  
**GetCoordinates** (*self*: pyOBabel.pyOBabel.OBMol) → list  
**GetDimension** (*self*: pyOBabel.pyOBabel.OBMol) → int  
**GetExactMass** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: bool) → float  
**GetFormula** (*self*: pyOBabel.pyOBabel.OBMol) → str  
**GetHeatFormation** (*self*: pyOBabel.pyOBabel.OBMol) → float  
**GetMolWt** (*self*: pyOBabel.pyOBabel.OBMol, *implicitH*: bool=True) → float  
**GetSpacedFormula** (*self*: pyOBabel.pyOBabel.OBMol, *ones*: int=0, *sp*: str=' ', *implicitH*: bool=True) → str  
**GetTitle** (*self*: pyOBabel.pyOBabel.OBMol, *replaceNewlines*: bool=True) → str  
**GetTorsionByAtoms** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: OpenBabel::OAtom, *arg1*: OpenBabel::OAtom, *arg2*: OpenBabel::OAtom, *arg3*: OpenBabel::OAtom) → float  
**GetTorsionByIndex** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int, *arg1*: int, *arg2*: int, *arg3*: int) → float  
**GetTotalCharge** (*self*: pyOBabel.pyOBabel.OBMol) → int  
**GetTotalSpinMultiplicity** (*self*: pyOBabel.pyOBabel.OBMol) → int  
**Has2D** (*self*: pyOBabel.pyOBabel.OBMol, *Not3D*: bool=False) → bool  
**Has3D** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasAromaticPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasAtomTypesPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasChainsPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasChiralityPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasClosureBondsPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasHybridizationPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasHydrogensAdded** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasLSSRPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasNonZeroCoords** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasPartialChargesPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasRingAtomsAndBondsPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasRingTypesPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasSSSRPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**HasSpinMultiplicityAssigned** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**InsertAtom** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: OpenBabel::OAtom) → bool  
**IsChiral** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**IsCorrectedForPH** (*self*: pyOBabel.pyOBabel.OBMol) → bool

**MakeDativeBonds** (*self*: pyOBabel.pyOBabel.OBMol) → bool  
**NewAtom** (*self*: pyOBabel.pyOBabel.OBMol) → OpenBabel::OAtom  
**NewAtomById** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → OpenBabel::OAtom  
**NewBond** (*self*: pyOBabel.pyOBabel.OBMol) → OpenBabel::OBond  
**NewBondById** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → OpenBabel::OBond  
**NewResidue** (*self*: pyOBabel.pyOBabel.OBMol) → OpenBabel::OBResidue  
**NumAtoms** (*self*: pyOBabel.pyOBabel.OBMol) → int  
**NumBonds** (*self*: pyOBabel.pyOBabel.OBMol) → int  
**NumConformers** (*self*: pyOBabel.pyOBabel.OBMol) → int  
**NumHvyAtoms** (*self*: pyOBabel.pyOBabel.OBMol) → int  
**NumResidues** (*self*: pyOBabel.pyOBabel.OBMol) → int  
**NumRotors** (*self*: pyOBabel.pyOBabel.OBMol, *sampleRingBonds*: bool=False) → int  
**PerceiveBondOrders** (*self*: pyOBabel.pyOBabel.OBMol) → None  
**RenumberAtomsByAtoms** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: List[OpenBabel::OAtom]) → None  
**RenumberAtomsByIndex** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: List[int]) → None  
**ReserveAtoms** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → None  
**Separate** (*self*: pyOBabel.pyOBabel.OBMol, *startIndex*: int=1) → List[pyOBabel.pyOBabel.OBMol]  
**SetAromaticPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None  
**SetAtomTypesPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None  
**SetAutomaticFormalCharge** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: bool) → None  
**SetAutomaticPartialCharge** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: bool) → None  
**SetChainsPerceived** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: bool) → None  
**SetChiralityPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None  
**SetClosureBondsPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None  
**SetConformerByIndex** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → None  
**SetConformerEnergies** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: List[float]) → None  
**SetConformersCoords** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: list) → None  
**SetCoordinates** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: list) → None  
**SetCorrectedForPH** (*self*: pyOBabel.pyOBabel.OBMol) → None  
**SetDimension** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → None  
**SetFlags** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → None  
**SetFormula** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: str) → None  
**SetHeatFormation** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: float) → None  
**SetHybridizationPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None  
**SetHydrogensAdded** (*self*: pyOBabel.pyOBabel.OBMol) → None  
**SetIsPatternStructure** (*self*: pyOBabel.pyOBabel.OBMol) → None

**SetLSSRPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None

**SetPartialChargesPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None

**SetRingAtomsAndBondsPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None

**SetRingTypesPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None

**SetSSSRPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None

**SetSpinMultiplicityAssigned** (*self*: pyOBabel.pyOBabel.OBMol) → None

**SetTitle** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: str) → None

**SetTorsion** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: OpenBabel::OAtom, *arg1*: OpenBabel::OAtom, *arg2*: OpenBabel::OAtom, *arg3*: OpenBabel::OAtom, *arg4*: float) → None

**SetTotalCharge** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → None

**SetTotalSpinMultiplicity** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → None

**StripSalts** (*self*: pyOBabel.pyOBabel.OBMol, *threshold*: int=0) → bool

**ToInertialFrameForAllConformers** (*self*: pyOBabel.pyOBabel.OBMol) → None

**ToInertialFrameForOneConformer** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int, *arg1*: float) → None

**TranslateAllConformers** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: pyOBabel.pyOBabel.vector3) → None

**TranslateOneConformer** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: pyOBabel.pyOBabel.vector3, *arg1*: int) → None

**UnsetAromaticPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None

**UnsetFlag** (*self*: pyOBabel.pyOBabel.OBMol, *arg0*: int) → None

**UnsetHydrogensAdded** (*self*: pyOBabel.pyOBabel.OBMol) → None

**UnsetLSSRPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None

**UnsetPartialChargesPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None

**UnsetRingTypesPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None

**UnsetSSSRPerceived** (*self*: pyOBabel.pyOBabel.OBMol) → None

#### **atoms**

*List of the atoms – OAtom. It yields a read-only list.*

#### **Example**

```
# Print index and atomic number
for atom in mol.atoms:
    print(atom.GetIndex(), atom.GetAtomicNum())
```

#### **bonds**

*List of the bonds – OBBond. It yields a read-only list.*



### Example

```
# Print bond order, first atom index and second atom index
for bond in mol.bonds:
    print(bond.GetBondOrder(), bond.GetBeginAtom().GetIndex(),
          bond.GetEndAtom().GetIndex())
```

**copy** (*self*: pyOBabel.pyOBabel.OBMol) → pyOBabel.pyOBabel.OBMol

**residues**

List of the residues - *OBResidue*.

### Example

```
# Print Residue name
for residue in mol.residues:
    print(residue.GetName())
```

## OBAtom

---

*OBAtom*

---

### Documentation

**class OBAtom**

**AddBond** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: OpenBabel::OBBond) → None

**AddResidue** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: OpenBabel::OBResidue) → None

**AverageBondAngle** (*self*: pyOBabel.pyOBabel.OBAtom) → float

**BOSum** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**Clear** (*self*: pyOBabel.pyOBabel.OBAtom) → bool

**ClearBond** (*self*: pyOBabel.pyOBabel.OBAtom) → None

**CountBondsOfOrder** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → int

**CountFreeOxygens** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**CountFreeSulfurs** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**CountRingBonds** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**DeleteResidue** (*self*: pyOBabel.pyOBabel.OBAtom) → None

**Duplicate** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: pyOBabel.pyOBabel.OBAtom) → None

**ExplicitHydrogenCount** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: bool) → int

**GetAngle** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: pyOBabel.pyOBabel.OBAtom, *arg1*: pyOBabel.pyOBabel.OBAtom) → float

**GetAngleBetweenAtoms** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: pyOBabel.pyOBabel.OBAtom, *arg1*: pyOBabel.pyOBabel.OBAtom) → float

**GetAtomicMass** (*self*: pyOBabel.pyOBabel.OBAtom) → float

**GetAtomicNum** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**GetBond** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: pyOBabel.pyOBabel.OBAtom) → OpenBabel::OBBond

**GetDistanceFromAtom** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: pyOBabel.pyOBabel.OBAtom) → float

**GetDistanceFromIndex** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → float

**GetDistanceFromVector** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: pyOBabel.pyOBabel.vector3) → float

**GetExactMass** (*self*: pyOBabel.pyOBabel.OBAtom) → float

**GetFormalCharge** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**GetHeteroValence** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**GetHvyValence** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**GetHyb** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**GetId** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**GetIdx** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**GetImplicitHCount** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**GetIndex** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**GetIsotope** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**GetNewBondVector** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: float) → pyOBabel.pyOBabel.vector3

**GetNextAtom** (*self*: pyOBabel.pyOBabel.OBAtom) → pyOBabel.pyOBabel.OBAtom

**GetParent** (*self*: pyOBabel.pyOBabel.OBAtom) → pyOBabel.pyOBabel.OBMol

**GetPartialCharge** (*self*: pyOBabel.pyOBabel.OBAtom) → float

**GetResidue** (*self*: pyOBabel.pyOBabel.OBAtom) → OpenBabel::OBResidue

**GetSpinMultiplicity** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**GetType** (*self*: pyOBabel.pyOBabel.OBAtom) → str

**GetValence** (*self*: pyOBabel.pyOBabel.OBAtom) → int

**GetVector** (*self*: pyOBabel.pyOBabel.OBAtom) → pyOBabel.pyOBabel.vector3

**GetX** (*self*: pyOBabel.pyOBabel.OBAtom) → float

**GetY** (*self*: pyOBabel.pyOBabel.OBAtom) → float

**GetZ** (*self*: pyOBabel.pyOBabel.OBAtom) → float

**HasAlphaBetaUnsat** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: bool) → bool

**HasAromaticBond** (*self*: pyOBabel.pyOBabel.OBAtom) → bool

**HasBondOfOrder** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → bool

**HasChiralVolume** (*self*: pyOBabel.pyOBabel.OBAtom) → bool

**HasChiralitySpecified** (*self*: pyOBabel.pyOBabel.OBAtom) → bool

**HasDoubleBond** (*self*: pyOBabel.pyOBabel.OBAtom) → bool

**HasNonSingleBond** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**HasResidue** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**HasSingleBond** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**HighestBondOrder** (*self*: pyOBabel.pyOBabel.OBAtom) → int  
**HtoMethyl** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsAmideNitrogen** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsAntiClockwise** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsAromatic** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsAromaticNOxide** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsAxial** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsCarboxylOxygen** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsChiral** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsClockwise** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsConnected** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsHbondAcceptor** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsHbondAcceptorSimple** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsHbondDonor** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsHbondDonorH** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsHetAtom** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsHeteroatom** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsInRing** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsInRingSize** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → bool  
**IsNegativeStereo** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsNitroOxygen** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsNonPolarHydrogen** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsOneFour** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsOneThree** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsPhosphateOxygen** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsPolarHydrogen** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsPositiveStereo** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**IsSulfateOxygen** (*self*: pyOBabel.pyOBabel.OBAtom) → bool  
**LewisAcidBaseCounts** (*self*: pyOBabel.pyOBabel.OBAtom) → Tuple[int, int]  
**MatchesSMARTS** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: str) → bool  
**MemberOfRingCount** (*self*: pyOBabel.pyOBabel.OBAtom) → int  
**MemberOfRingSize** (*self*: pyOBabel.pyOBabel.OBAtom) → int  
**NewResidue** (*self*: pyOBabel.pyOBabel.OBAtom) → None

**SetAntiClockwiseStereo** (*self*: pyOBabel.pyOBabel.OBAtom) → None

**SetAromatic** (*self*: pyOBabel.pyOBabel.OBAtom) → None

**SetAtomicNum** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → None

**SetChiral** (*self*: pyOBabel.pyOBabel.OBAtom) → None

**SetClockwiseStereo** (*self*: pyOBabel.pyOBabel.OBAtom) → None

**SetFormalCharge** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → None

**SetHyb** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → None

**SetHybAndGeom** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → bool

**SetId** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → None

**SetIdx** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → None

**SetImplicitHCount** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → None

**SetInRing** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: bool) → None

**SetIsotope** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → None

**SetNegativeStereo** (*self*: pyOBabel.pyOBabel.OBAtom) → None

**SetParent** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: pyOBabel.pyOBabel.OBMol) → None

**SetPartialCharge** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: float) → None

**SetPositiveStereo** (*self*: pyOBabel.pyOBabel.OBAtom) → None

**SetResidue** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: OpenBabel::OBResidue) → None

**SetSpinMultiplicity** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: int) → None

**SetType** (*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: str) → None

**SetVector** (\*args, \*\*kwargs)  
Overloaded function.

1. SetVector(*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: pyOBabel.pyOBabel.vector3) → None
2. SetVector(*self*: pyOBabel.pyOBabel.OBAtom, *arg0*: float, *arg1*: float, *arg2*: float) → None

**SmallestBondAngle** (*self*: pyOBabel.pyOBabel.OBAtom) → float

**UnsetAromatic** (*self*: pyOBabel.pyOBabel.OBAtom) → None

**UnsetStereo** (*self*: pyOBabel.pyOBabel.OBAtom) → None

## OBBond

---

*OBBond*

---

## Documentation

### class OBBond

**GetBO** (*self*: pyOBabel.pyOBabel.OBBond) → int

**GetBeginAtom** (*self*: pyOBabel.pyOBabel.OBBond) → pyOBabel.pyOBabel.OBAtom

---

```

GetBeginAtomIdx (self: pyOBabel.pyOBabel.OBBond) → int
GetBondOrder (self: pyOBabel.pyOBabel.OBBond) → int
GetEndAtom (self: pyOBabel.pyOBabel.OBBond) → pyOBabel.pyOBabel.OBAtom
GetEndAtomIdx (self: pyOBabel.pyOBabel.OBBond) → int
GetEquibLength (self: pyOBabel.pyOBabel.OBBond) → float
GetFlags (self: pyOBabel.pyOBabel.OBBond) → int
GetId (self: pyOBabel.pyOBabel.OBBond) → int
GetIdx (self: pyOBabel.pyOBabel.OBBond) → int
GetLength (self: pyOBabel.pyOBabel.OBBond) → float
GetNbrAtom (self: pyOBabel.pyOBabel.OBBond, arg0: pyOBabel.pyOBabel.OBAtom) → pyOBabel.pyOBabel.OBAtom
GetNbrAtomIdx (self: pyOBabel.pyOBabel.OBBond, arg0: pyOBabel.pyOBabel.OBAtom) → int
GetParent (self: pyOBabel.pyOBabel.OBBond) → pyOBabel.pyOBabel.OBMol
IsAmide (self: pyOBabel.pyOBabel.OBBond) → bool
IsAromatic (self: pyOBabel.pyOBabel.OBBond) → bool
IsCarbonyl (self: pyOBabel.pyOBabel.OBBond) → bool
IsCisOrTrans (self: pyOBabel.pyOBabel.OBBond) → bool
IsClosure (self: pyOBabel.pyOBabel.OBBond) → bool
IsDoubleBondGeometry (self: pyOBabel.pyOBabel.OBBond) → bool
IsDown (self: pyOBabel.pyOBabel.OBBond) → bool
IsEster (self: pyOBabel.pyOBabel.OBBond) → bool
IsHash (self: pyOBabel.pyOBabel.OBBond) → bool
IsInRing (self: pyOBabel.pyOBabel.OBBond) → bool
IsPrimaryAmide (self: pyOBabel.pyOBabel.OBBond) → bool
IsRotor (self: pyOBabel.pyOBabel.OBBond, arg0: bool) → bool
IsSecondaryAmide (self: pyOBabel.pyOBabel.OBBond) → bool
IsTertiaryAmide (self: pyOBabel.pyOBabel.OBBond) → bool
IsUp (self: pyOBabel.pyOBabel.OBBond) → bool
IsWedge (self: pyOBabel.pyOBabel.OBBond) → bool
IsWedgeOrHash (self: pyOBabel.pyOBabel.OBBond) → bool
SetAromatic (self: pyOBabel.pyOBabel.OBBond) → None
SetBO (self: pyOBabel.pyOBabel.OBBond, arg0: int) → None
SetBegin (self: pyOBabel.pyOBabel.OBBond, arg0: pyOBabel.pyOBabel.OBAtom) → None
SetBondOrder (self: pyOBabel.pyOBabel.OBBond, arg0: int) → None
SetClosure (self: pyOBabel.pyOBabel.OBBond, arg0: bool) → None
SetDown (self: pyOBabel.pyOBabel.OBBond) → None
SetEnd (self: pyOBabel.pyOBabel.OBBond, arg0: pyOBabel.pyOBabel.OBAtom) → None

```

**SetHash** (*self*: pyOBabel.pyOBabel.OBBond) → None

**SetId** (*self*: pyOBabel.pyOBabel.OBBond, *arg0*: int) → None

**SetInRing** (*self*: pyOBabel.pyOBabel.OBBond, *arg0*: bool) → None

**SetLength** (\*args, \*\*kwargs)

Overloaded function.

1. SetLength(*self*: pyOBabel.pyOBabel.OBBond, *arg0*: pyOBabel.pyOBabel.OBAtom, *arg1*: float) -> None

2. SetLength(*self*: pyOBabel.pyOBabel.OBBond, *arg0*: float) -> None

**SetParent** (*self*: pyOBabel.pyOBabel.OBBond, *arg0*: pyOBabel.pyOBabel.OBMol) → None

**SetUp** (*self*: pyOBabel.pyOBabel.OBBond) → None

**SetWedge** (*self*: pyOBabel.pyOBabel.OBBond) → None

**SetWedgeOrHash** (*self*: pyOBabel.pyOBabel.OBBond) → None

**UnsetAromatic** (*self*: pyOBabel.pyOBabel.OBBond) → None

**UnsetDown** (*self*: pyOBabel.pyOBabel.OBBond) → None

**UnsetHash** (*self*: pyOBabel.pyOBabel.OBBond) → None

**UnsetUp** (*self*: pyOBabel.pyOBabel.OBBond) → None

**UnsetWedge** (*self*: pyOBabel.pyOBabel.OBBond) → None

## OBResidue

---

*OBResidue*

---

## Documentation

### class OBResidue

**GetName** (*self*: pyOBabel.pyOBabel.OBResidue) → str

**SetName** (*self*: pyOBabel.pyOBabel.OBResidue, *arg0*: str) → None

### vector3

<i>vector3</i>	Represents a vector in 3-dimensional real space.
<i>vector3.Set</i> ( <i>self</i> , inX, inY, inZ)	Set x,y and z-component of a vector :param inX: Input X-coordinate :type inX: float :param inY: Input Y-coordinate :type inY: float :param inZ: Input Z-coordinate :type inZ: float
<i>vector3.SetX</i> ( <i>self</i> , inX)	To set the x-coordinate of the vector :param inX: Input X-coordinate :type inX: float
<i>vector3.SetY</i> ( <i>self</i> , inY)	To set the y-coordinate of the vector :param inX: Input Y-coordinate :type inX: float

Continued on next page

Table 6 – continued from previous page

<code>vector3.SetZ(self, inZ)</code>	To set the z-coordinate of the vector :param inX: Input Z-coordinate :type inX: float
<code>vector3.GetX(self)</code>	to Get the x-coordinate of the vector
<code>vector3.GetY(self)</code>	to Get the y-coordinate of the vector
<code>vector3.GetZ(self)</code>	to Get the z-coordinate of the vector
<code>vector3.Get(self)</code>	To Get coordinates as a list
<code>vector3.copy(self)</code>	To copy this vector
<code>vector3.normalize(self)</code>	Normalize the vector
<code>vector3.CanBeNormalized(self)</code>	Whether a vector can be normalized
<code>vector3.length_2(self)</code>	The length of the vector squared
<code>vector3.length(self)</code>	The length of the vector
<code>vector3.IsApprox(self, arg0, arg1)</code>	Safe comparison for floating-point vector3
<code>vector3.distSq(self, v)</code>	Square of distance between self and input vector.
<code>vector3.createOrthoVector(self)</code>	Creates a vector of length one, orthogonal to this
<code>dot(v1, v2)</code>	Dot product of two vectors
<code>cross(v1, v2)</code>	Cross product of two vectors
<code>vectorAngle(v1, v2)</code>	Calculate the angle between vectors (in degrees)
<code>CalcTorsionAngle(a, b, c, d)</code>	Calculate the torsion angle between vectors (in degrees)
<code>Point2PlaneSigned(a, b, c, d)</code>	Calculate the signed distance of point a to the plane determined by b,c,d
<code>Point2Plane(a, b, c, d)</code>	Calculate the distance of point a to the plane determined by b,c,d
<code>Point2PlaneAngle(a, b, c, d)</code>	Calculate the angle between point a and the plane determined by b,c,d
<code>Point2Line(a, b, c)</code>	Calculate the distance of a point a to a line determined by b and c

## Documentation

### class vector3

Represents a vector in 3-dimensional real space.

The vector3 class was designed to simplify operations with floating point coordinates. To this end many of the common operations have been overloaded for simplicity. Vector addition, subtraction, scalar multiplication, dot product, cross product, magnitude and a number of other utility functions are built in to the vector class. For a full description of the class member functions please consult the documentation. The following code demonstrates several of the functions of the vector class:

### Example

```
v1, v2, v3 = vector3(), vector3(), vector3()
v1.Set([1, 2, 3])
v2.Set([4, 5, 6])
v3 = cross(v1,v2)
v3 *= 2.5
v3.normalize()
```

**CanBeNormalized** (*self*: pyOBabel.pyOBabel.vector3) → bool

Whether a vector can be normalized

**Returns** CanBeNormalize – True or False

**Return type** bool

**Get** (*self*: *pyOBabel.pyOBabel.vector3*) → list

To Get coordinates as a list

**Returns** list – coordinates as a list

**Return type** list

**GetX** (*self*: *pyOBabel.pyOBabel.vector3*) → float

to Get the x-coordinate of the vector

**Returns** outX – x-coordinate of the vector

**Return type** float

**GetY** (*self*: *pyOBabel.pyOBabel.vector3*) → float

to Get the y-coordinate of the vector

**Returns** outY – y-coordinate of the vector

**Return type** float

**GetZ** (*self*: *pyOBabel.pyOBabel.vector3*) → float

to Get the z-coordinate of the vector

**Returns** outZ – z-coordinate of the vector

**Return type** float

**IsApprox** (*self*: *pyOBabel.pyOBabel.vector3*, *arg0*: *pyOBabel.pyOBabel.vector3*, *arg1*: *float*) → bool

Safe comparison for floating-point vector3

True if this vector is approximately equal to the other vector to the input precision. More specifically, this method works exactly like the `pyOBabel.IsApprox()` function, replacing the absolute value for doubles by the norm for vectors.

**Parameters**

- **other** (*vector3*) – The vector for comparison
- **precision** (*float*) – This parameter plays the same role as in `pyOBabel.IsApprox()`

**Returns** IsApprox – True or False

**Return type** bool

**Set** (*self*: *pyOBabel.pyOBabel.vector3*, *inX*: *float*, *inY*: *float*, *inZ*: *float*) → None

Set x,y and z-component of a vector :param inX: Input X-coordinate :type inX: float :param inY: Input Y-coordinate :type inY: float :param inZ: Input Z-coordinate :type inZ: float

**Returns**

**Return type** None

**SetX** (*self*: *pyOBabel.pyOBabel.vector3*, *inX*: *float*) → None

To set the x-coordinate of the vector :param inX: Input X-coordinate :type inX: float

**Returns**

**Return type** None

**SetY** (*self*: *pyOBabel.pyOBabel.vector3*, *inY*: *float*) → None

To set the y-coordinate of the vector :param inX: Input Y-coordinate :type inX: float

**Returns**

**Return type** None



**SetZ** (*self*: pyOBabel.pyOBabel.vector3, *inZ*: float) → None

To set the z-coordinate of the vector :param inX: Input Z-coordinate :type inX: float

**Returns**

**Return type** None

**copy** (*self*: pyOBabel.pyOBabel.vector3) → pyOBabel.pyOBabel.vector3

To copy this vector

**Parameters** **inpVector** (*vector3*) – Vector to copy

**Returns**

**Return type** None

**createOrthoVector** (*self*: pyOBabel.pyOBabel.vector3) → pyOBabel.pyOBabel.vector3

Creates a vector of length one, orthogonal to this

Create a unit orthogonal vector

**Returns** **v** – Unit orthogonal vector or *None* if not successfull

**Return type** *vector3*

**distSq** (*self*: pyOBabel.pyOBabel.vector3, *v*: pyOBabel.pyOBabel.vector3) → float

Square of distance between self and input vector.

square of the distance between self and vv. equivalent to length\_2(self-vv)

**Parameters** **v** (*vector3*) – Vector to copy

**Returns** **length** – The square of distance between self and input vector

**Return type** float

**length** (*self*: pyOBabel.pyOBabel.vector3) → float

The length of the vector

**Returns** **length** – The length of the vector

**Return type** float

**length\_2** (*self*: pyOBabel.pyOBabel.vector3) → float

The length of the vector squared

**Returns** **length2** – The squared length of the vector

**Return type** float

**normalize** (*self*: pyOBabel.pyOBabel.vector3) → pyOBabel.pyOBabel.vector3

Normalize the vector

Scales a vector to give it length one.

**Returns** **normalized** – The normalized vector

**Return type** *vector3*

**dot** (*v1*: pyOBabel.pyOBabel.vector3, *v2*: pyOBabel.pyOBabel.vector3) → float

Dot product of two vectors

**Parameters**

- **v1** (*vector3*) – First vector
- **v2** (*vector3*) – Second vector

**Returns** **product** – Dot product of two vectors

**Return type** float

**cross** (*v1*: *pyOBabel.pyOBabel.vector3*, *v2*: *pyOBabel.pyOBabel.vector3*) → *pyOBabel.pyOBabel.vector3*  
Cross product of two vectors

**Parameters**

- **v1** (*vector3*) – First vector
- **v2** (*vector3*) – Second vector

**Returns** **cross** – Cross product of two vectors

**Return type** *vector3*

**vectorAngle** (*v1*: *pyOBabel.pyOBabel.vector3*, *v2*: *pyOBabel.pyOBabel.vector3*) → float  
Calculate the angle between vectors (in degrees)

**Parameters**

- **v1** (*vector3*) – First vector
- **v2** (*vector3*) – Second vector

**Returns** **angle** – Angle between vectors (in degrees)

**Return type** float

**CalcTorsionAngle** (*a*: *pyOBabel.pyOBabel.vector3*, *b*: *pyOBabel.pyOBabel.vector3*, *c*: *pyOBabel.pyOBabel.vector3*, *d*: *pyOBabel.pyOBabel.vector3*) → float  
Calculate the torsion angle between vectors (in degrees)

**Parameters**

- **a** (*vector3*) – First vector
- **b** (*vector3*) – Second vector
- **c** (*vector3*) – Third vector
- **d** (*vector3*) – fourth vector

**Returns** **angle** – Torsion angle between vectors (in degrees)

**Return type** float

**Point2PlaneSigned** (*a*: *pyOBabel.pyOBabel.vector3*, *b*: *pyOBabel.pyOBabel.vector3*, *c*: *pyOBabel.pyOBabel.vector3*, *d*: *pyOBabel.pyOBabel.vector3*) → float  
Calculate the signed distance of point a to the plane determined by b,c,d

**Parameters**

- **a** (*vector3*) – First vector
- **b** (*vector3*) – Second vector
- **c** (*vector3*) – Third vector
- **d** (*vector3*) – fourth vector

**Returns** **distance** – Signed distance between point to a plane

**Return type** float

**Point2Plane** (*a*: *pyOBabel.pyOBabel.vector3*, *b*: *pyOBabel.pyOBabel.vector3*, *c*: *pyOBabel.pyOBabel.vector3*, *d*: *pyOBabel.pyOBabel.vector3*) → float  
Calculate the distance of point a to the plane determined by b,c,d

**Parameters**

- **a** (`vector3`) – First vector
- **b** (`vector3`) – Second vector
- **c** (`vector3`) – Third vector
- **d** (`vector3`) – fourth vector

**Returns** **distance** – Distance between point to a plane

**Return type** float

**Point2PlaneAngle** (*a: pyOBabel.pyOBabel.vector3, b: pyOBabel.pyOBabel.vector3, c: pyOBabel.pyOBabel.vector3, d: pyOBabel.pyOBabel.vector3*) → float

Calculate the angle between point a and the plane determined by b,c,d

**Parameters**

- **a** (`vector3`) – First vector
- **b** (`vector3`) – Second vector
- **c** (`vector3`) – Third vector
- **d** (`vector3`) – fourth vector

**Returns** **angle** – angle between a point and plane

**Return type** float

**Point2Line** (*a: pyOBabel.pyOBabel.vector3, b: pyOBabel.pyOBabel.vector3, c: pyOBabel.pyOBabel.vector3*) → float

Calculate the distance of a point a to a line determined by b and c

**Parameters**

- **a** (`vector3`) – First vector
- **b** (`vector3`) – Second vector
- **c** (`vector3`) – Third vector

**Returns** **distance** – Distance between a point and a line

**Return type** float

## OBBitVec

---

*OBBitVec*

---

## Documentation

**class** OBBitVec

**BitIsOn** (*self: pyOBabel.pyOBabel.OBBitVec, arg0: int*) → bool

**Clear** (*self: pyOBabel.pyOBabel.OBBitVec*) → None

**CountBits** (*self: pyOBabel.pyOBabel.OBBitVec*) → int

**EndBit** (*self: pyOBabel.pyOBabel.OBBitVec*) → int

**FirstBit** (*self: pyOBabel.pyOBabel.OBBitVec, arg0: int*) → int

**Fold** (*self*: pyOBabel.pyOBabel.OBBitVec, *arg0*: int) → None

**FromString** (*self*: pyOBabel.pyOBabel.OBBitVec, *arg0*: str, *arg1*: int) → None

**FromVecInt** (*self*: pyOBabel.pyOBabel.OBBitVec, *arg0*: List[int]) → None

**GetSize** (*self*: pyOBabel.pyOBabel.OBBitVec) → int

**IsEmpty** (*self*: pyOBabel.pyOBabel.OBBitVec) → bool

**Negate** (*self*: pyOBabel.pyOBabel.OBBitVec) → None

**NextBit** (*self*: pyOBabel.pyOBabel.OBBitVec, *arg0*: int) → int

**Resize** (*self*: pyOBabel.pyOBabel.OBBitVec, *arg0*: int) → bool

**ResizeWords** (*self*: pyOBabel.pyOBabel.OBBitVec, *arg0*: int) → bool

**SetBitOff** (*self*: pyOBabel.pyOBabel.OBBitVec, *arg0*: int) → None

**SetBitOn** (*self*: pyOBabel.pyOBabel.OBBitVec, *arg0*: int) → None

**SetRangeOff** (*self*: pyOBabel.pyOBabel.OBBitVec, *arg0*: int, *arg1*: int) → None

**SetRangeOn** (*self*: pyOBabel.pyOBabel.OBBitVec, *arg0*: int, *arg1*: int) → None

## OBConversion

---

*OBConversion*

---

## Documentation

### class OBConversion

**AddChemObject** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: pyOBabel.pyOBabel.OBBase) → int

**AddOption** (*self*: pyOBabel.pyOBabel.OBConversion, *opt*: str, *opttyp*: pyOBabel.pyOBabel.OBConversion.Option\_type=Option\_type.OUTOPTIONS, *txt*: str=0) → None

**CloseOutFile** (*self*: pyOBabel.pyOBabel.OBConversion) → None

**Convert** (*self*: pyOBabel.pyOBabel.OBConversion) → int

**FullConvert** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: List[str], *arg1*: str, *arg2*: List[str]) → int

**GetChemObject** (*self*: pyOBabel.pyOBabel.OBConversion) → pyOBabel.pyOBabel.OBBase

**GetCount** (*self*: pyOBabel.pyOBabel.OBConversion) → int

**GetOptions** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: pyOBabel.pyOBabel.OBConversion.Option\_type) → Dict[str, str]

**GetOutputIndex** (*self*: pyOBabel.pyOBabel.OBConversion) → int

**GetSupportedInputFormat** (*self*: pyOBabel.pyOBabel.OBConversion) → List[str]

**GetSupportedOutputFormat** (*self*: pyOBabel.pyOBabel.OBConversion) → List[str]

**IsFirstInput** (*self*: pyOBabel.pyOBabel.OBConversion) → bool

**IsLast** (*self*: pyOBabel.pyOBabel.OBConversion) → bool

**IsLastFile** (*self*: pyOBabel.pyOBabel.OBConversion) → bool

**IsOption** (*self*: pyOBabel.pyOBabel.OBConversion, *opt*: str, *opttyp*: pyOBabel.pyOBabel.OBConversion.Option\_type=Option\_type.OUTOPTIONS) → str

**OpenInAndOutFiles** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: str, *arg1*: str) → bool

**Read** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: pyOBabel.pyOBabel.OBBase) → bool

**ReadFile** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: pyOBabel.pyOBabel.OBBase, *arg1*: str) → bool

**ReadString** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: pyOBabel.pyOBabel.OBBase, *arg1*: str) → bool

**RemoveOption** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: str, *arg1*: pyOBabel.pyOBabel.OBConversion.Option\_type) → bool

**SetFirstInput** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: bool) → None

**SetInAndOutFormats** (*self*: pyOBabel.pyOBabel.OBConversion, *inID*: str, *outID*: str, *ingzip*: bool=False, *outgzip*: bool=False) → None

**SetInFormat** (*self*: pyOBabel.pyOBabel.OBConversion, *inID*: str, *isgzip*: bool=False) → bool

**SetLast** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: bool) → None

**SetMoreFilesToCome** (*self*: pyOBabel.pyOBabel.OBConversion) → None

**SetOneObjectOnly** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: bool) → None

**SetOptions** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: str, *arg1*: pyOBabel.pyOBabel.OBConversion.Option\_type) → None

**SetOutFormat** (*self*: pyOBabel.pyOBabel.OBConversion, *inID*: str, *isgzip*: bool=False) → bool

**SetOutputIndex** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: int) → None

**Write** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: pyOBabel.pyOBabel.OBBase) → bool

**WriteFile** (*self*: pyOBabel.pyOBabel.OBConversion, *arg0*: pyOBabel.pyOBabel.OBBase, *arg1*: str) → bool

**WriteString** (*self*: pyOBabel.pyOBabel.OBConversion, *obmol*: pyOBabel.pyOBabel.OBBase, *trimWhitespace*: bool=False) → str

## OBuilder

---

*OBuilder*

---

## Documentation

### class OBuilder

**AddNbrs** (*fragment*: pyOBabel.pyOBabel.OBBitVec, *atom*: pyOBabel.pyOBabel.OBAtom) → None

**Build** (*self*: pyOBabel.pyOBabel.OBBuilder, *mol*: pyOBabel.pyOBabel.OBMol, *stereoWarnings*: bool=True) → bool

**Connect** (*\*args*, *\*\*kwargs*)  
Overloaded function.

1. **Connect**(*mol*: pyOBabel.pyOBabel.OBMol, *a*: int, *b*: int, *newpos*: pyOBabel.pyOBabel.vector3, *bondOrder*: int=1) → bool

2. `Connect(mol: pyOBabel.pyOBabel.OBMol, a: int, b: int, bondOrder: int=1) -> bool`

**CorrectStereoAtoms** (*\*args, \*\*kwargs*)

Overloaded function.

1. `CorrectStereoAtoms(mol: pyOBabel.pyOBabel.OBMol) -> bool`

2. `CorrectStereoAtoms(mol: pyOBabel.pyOBabel.OBMol, warn: bool=True) -> bool`

**GetFragment** (*atom: pyOBabel.pyOBabel.OBAtom*)  $\rightarrow$  `pyOBabel.pyOBabel.OBBitVec`

**GetNewBondVector** (*\*args, \*\*kwargs*)

Overloaded function.

1. `GetNewBondVector(atom: pyOBabel.pyOBabel.OBAtom) -> pyOBabel.pyOBabel.vector3`

2. `GetNewBondVector(atom: pyOBabel.pyOBabel.OBAtom, length: float) -> pyOBabel.pyOBabel.vector3`

**IsSpiroAtom** (*atomId: int, mol: pyOBabel.pyOBabel.OBMol*)  $\rightarrow$  `bool`

**LoadFragments** (*self: pyOBabel.pyOBabel.OBBuilder*)  $\rightarrow$  `None`

**SetKeepRings** (*self: pyOBabel.pyOBabel.OBBuilder*)  $\rightarrow$  `None`

**Swap** (*mol: pyOBabel.pyOBabel.OBMol, a: int, b: int, c: int, d: int*)  $\rightarrow$  `bool`

**UnsetKeepRings** (*self: pyOBabel.pyOBabel.OBBuilder*)  $\rightarrow$  `None`

## OBForceField and OBFFConstraints

---

*OBForceField*

---

*OBFFConstraints*

---

## Documentation

### class OBForceField

**AddInterGroup** (*self: pyOBabel.pyOBabel.OBForceField, group: pyOBabel.pyOBabel.OBBitVec*)  $\rightarrow$  `None`

**AddInterGroups** (*self: pyOBabel.pyOBabel.OBForceField, group1: pyOBabel.pyOBabel.OBBitVec, group2: pyOBabel.pyOBabel.OBBitVec*)  $\rightarrow$  `None`

**AddIntraGroup** (*self: pyOBabel.pyOBabel.OBForceField, group: pyOBabel.pyOBabel.OBBitVec*)  $\rightarrow$  `None`

**ClearGroups** (*self: pyOBabel.pyOBabel.OBForceField*)  $\rightarrow$  `None`

**ConjugateGradients** (*self: pyOBabel.pyOBabel.OBForceField, steps: int, econv: float=9.999999974752427e-07, method: int=2*)  $\rightarrow$  `None`

**ConjugateGradientsInitialize** (*self: pyOBabel.pyOBabel.OBForceField, steps: int, econv: float=9.999999974752427e-07, method: int=2*)  $\rightarrow$  `None`

**ConjugateGradientsTakeNSteps** (*self: pyOBabel.pyOBabel.OBForceField, n: int*)  $\rightarrow$  `bool`

**CorrectVelocities** (*self: pyOBabel.pyOBabel.OBForceField*)  $\rightarrow$  `None`

**DetectExplosion** (*self: pyOBabel.pyOBabel.OBForceField*)  $\rightarrow$  `bool`

**DistanceGeometry** (*self: pyOBabel.pyOBabel.OBForceField*)  $\rightarrow$  `None`

**E\_Angle** (*self*: pyOBabel.pyOBabel.OBForceField, *arg0*: bool) → float  
**E\_Bond** (*self*: pyOBabel.pyOBabel.OBForceField, *arg0*: bool) → float  
**E\_Electrostatic** (*self*: pyOBabel.pyOBabel.OBForceField, *arg0*: bool) → float  
**E\_OOP** (*self*: pyOBabel.pyOBabel.OBForceField, *arg0*: bool) → float  
**E\_StrBnd** (*self*: pyOBabel.pyOBabel.OBForceField, *arg0*: bool) → float  
**E\_Torsion** (*self*: pyOBabel.pyOBabel.OBForceField, *arg0*: bool) → float  
**E\_VDW** (*self*: pyOBabel.pyOBabel.OBForceField, *arg0*: bool) → float  
**EnableAllPairs** (*self*: pyOBabel.pyOBabel.OBForceField) → None  
**EnableCutOff** (*self*: pyOBabel.pyOBabel.OBForceField, *enable*: bool) → None  
**Energy** (*self*: pyOBabel.pyOBabel.OBForceField, *arg0*: bool) → float  
**FastRotorSearch** (*self*: pyOBabel.pyOBabel.OBForceField, *permute*: bool=True) → int  
**FindForceField** (*arg0*: str) → pyOBabel.pyOBabel.OBForceField  
**FindType** (*arg0*: str) → pyOBabel.pyOBabel.OBForceField  
**GenerateVelocities** (*self*: pyOBabel.pyOBabel.OBForceField) → None  
**GetAtomTypes** (*self*: pyOBabel.pyOBabel.OBForceField, *mol*: pyOBabel.pyOBabel.OBMol) → bool  
**GetConformers** (*self*: pyOBabel.pyOBabel.OBForceField, *mol*: pyOBabel.pyOBabel.OBMol) → bool  
**GetConstraints** (*self*: pyOBabel.pyOBabel.OBForceField) → pyOBabel.pyOBabel.OBFFConstraints  
**GetCoordinates** (*self*: pyOBabel.pyOBabel.OBForceField, *mol*: pyOBabel.pyOBabel.OBMol) → bool  
**GetElectrostaticCutOff** (*self*: pyOBabel.pyOBabel.OBForceField) → float  
**GetLineSearchType** (*self*: pyOBabel.pyOBabel.OBForceField) → int  
**GetLogLevel** (*self*: pyOBabel.pyOBabel.OBForceField) → int  
**GetNumElectrostaticPairs** (*self*: pyOBabel.pyOBabel.OBForceField) → int  
**GetNumPairs** (*self*: pyOBabel.pyOBabel.OBForceField) → int  
**GetNumVDWPairs** (*self*: pyOBabel.pyOBabel.OBForceField) → int  
**GetUnit** (*self*: pyOBabel.pyOBabel.OBForceField) → str  
**GetUpdateFrequency** (*self*: pyOBabel.pyOBabel.OBForceField) → int  
**GetVDWCutOff** (*self*: pyOBabel.pyOBabel.OBForceField) → float  
**HasAnalyticalGradients** (*self*: pyOBabel.pyOBabel.OBForceField) → bool  
**HasGroups** (*self*: pyOBabel.pyOBabel.OBForceField) → bool  
**IsCutOffEnabled** (*self*: pyOBabel.pyOBabel.OBForceField) → bool  
**IsSetupNeeded** (*self*: pyOBabel.pyOBabel.OBForceField, *mol*: pyOBabel.pyOBabel.OBMol) → bool  
**LineSearch** (*self*: pyOBabel.pyOBabel.OBForceField, *atom*: pyOBabel.pyOBabel.OBAtom, *direction*: pyOBabel.pyOBabel.vector3) → pyOBabel.pyOBabel.vector3  
**MakeNewInstance** (*self*: pyOBabel.pyOBabel.OBForceField) → pyOBabel.pyOBabel.OBForceField

**MolecularDynamicsTakenNSteps** (*self*: pyOBabel.pyOBabel.OBForceField, *steps*: int, *Temperature*: float, *timestep*: float=0.001, *method*: int=2) → None

**OBFFLog** (*self*: pyOBabel.pyOBabel.OBForceField, *msg*: str) → None

**ParseParamFile** (*self*: pyOBabel.pyOBabel.OBForceField) → bool

**PrintFormalCharges** (*self*: pyOBabel.pyOBabel.OBForceField) → None

**PrintPartialCharges** (*self*: pyOBabel.pyOBabel.OBForceField) → None

**PrintTypes** (*self*: pyOBabel.pyOBabel.OBForceField) → None

**PrintVelocities** (*self*: pyOBabel.pyOBabel.OBForceField) → None

**RandomRotorSearch** (*self*: pyOBabel.pyOBabel.OBForceField, *conformers*: int, *geomSteps*: int=2500, *sampleRingBonds*: bool=False) → None

**RandomRotorSearchInitialize** (*self*: pyOBabel.pyOBabel.OBForceField, *conformers*: int, *geomSteps*: int=2500, *sampleRingBonds*: bool=False) → None

**RandomRotorSearchNextConformer** (*self*: pyOBabel.pyOBabel.OBForceField, *geomSteps*: int=2500) → bool

**SetConformers** (*self*: pyOBabel.pyOBabel.OBForceField, *mol*: pyOBabel.pyOBabel.OBMol) → bool

**SetConstraints** (*self*: pyOBabel.pyOBabel.OBForceField, *constraints*: pyOBabel.pyOBabel.OBFFConstraints) → None

**SetCoordinates** (*self*: pyOBabel.pyOBabel.OBForceField, *mol*: pyOBabel.pyOBabel.OBMol) → bool

**SetElectrostaticCutOff** (*self*: pyOBabel.pyOBabel.OBForceField, *r*: float) → None

**SetFixAtom** (*self*: pyOBabel.pyOBabel.OBForceField, *index*: int) → None

**SetFormalCharges** (*self*: pyOBabel.pyOBabel.OBForceField) → bool

**SetIgnoreAtom** (*self*: pyOBabel.pyOBabel.OBForceField, *arg0*: int) → None

**SetLineSearchType** (*self*: pyOBabel.pyOBabel.OBForceField, *type*: int) → None

**SetLogLevel** (*self*: pyOBabel.pyOBabel.OBForceField, *arg0*: int) → bool

**SetParameterFile** (*self*: pyOBabel.pyOBabel.OBForceField, *filename*: str) → None

**SetPartialCharges** (*self*: pyOBabel.pyOBabel.OBForceField) → bool

**SetTypes** (*self*: pyOBabel.pyOBabel.OBForceField) → bool

**SetUpdateFrequency** (*self*: pyOBabel.pyOBabel.OBForceField, *f*: int) → None

**SetVDWCutOff** (*self*: pyOBabel.pyOBabel.OBForceField, *r*: float) → None

**Setup** (*self*: pyOBabel.pyOBabel.OBForceField, *arg0*: pyOBabel.pyOBabel.OBMol) → bool

**SetupCalculations** (*self*: pyOBabel.pyOBabel.OBForceField) → bool

**SetupPointers** (*self*: pyOBabel.pyOBabel.OBForceField) → bool

**SetupWithConstraints** (*self*: pyOBabel.pyOBabel.OBForceField, *arg0*: pyOBabel.pyOBabel.OBMol, *arg1*: pyOBabel.pyOBabel.OBFFConstraints) → bool

**SteepestDescent** (*self*: pyOBabel.pyOBabel.OBForceField, *steps*: int, *econv*: float=9.99999974752427e-07, *method*: int=2) → None

**SteepestDescentInitialize** (*self*: pyOBabel.pyOBabel.OBForceField, *steps*: int, *econv*: float=9.99999974752427e-07, *method*: int=2) → None



```

SteepestDescentTakeNSteps (self: pyOBabel.pyOBabel.OBForceField, n: int) → bool
SystematicRotorSearch (self: pyOBabel.pyOBabel.OBForceField, geomSteps: int=2500, sampleRingBonds: bool=False) → None
SystematicRotorSearchInitialize (self: pyOBabel.pyOBabel.OBForceField, geomSteps: int=2500, sampleRingBonds: bool=False) → int
SystematicRotorSearchNextConformer (self: pyOBabel.pyOBabel.OBForceField, geomSteps: int=2500) → bool
UnsetFixAtom (self: pyOBabel.pyOBabel.OBForceField) → None
UnsetIgnoreAtom (self: pyOBabel.pyOBabel.OBForceField) → None
UpdateConformers (self: pyOBabel.pyOBabel.OBForceField, mol: pyOBabel.pyOBabel.OBMol) → bool
UpdateCoordinates (self: pyOBabel.pyOBabel.OBForceField, mol: pyOBabel.pyOBabel.OBMol) → bool
UpdatePairsSimple (self: pyOBabel.pyOBabel.OBForceField) → None
Validate (self: pyOBabel.pyOBabel.OBForceField) → bool
ValidateConjugateGradients (self: pyOBabel.pyOBabel.OBForceField, steps: int) → None
ValidateGradientError (self: pyOBabel.pyOBabel.OBForceField, numgrad: pyOBabel.pyOBabel.vector3, anagrad: pyOBabel.pyOBabel.vector3) → pyOBabel.pyOBabel.vector3
ValidateGradients (self: pyOBabel.pyOBabel.OBForceField) → bool
ValidateLineSearch (self: pyOBabel.pyOBabel.OBForceField, atom: pyOBabel.pyOBabel.OBAtom, direction: pyOBabel.pyOBabel.vector3) → pyOBabel.pyOBabel.vector3
ValidateSteepestDescent (self: pyOBabel.pyOBabel.OBForceField, steps: int) → None
WeightedRotorSearch (self: pyOBabel.pyOBabel.OBForceField, conformers: int, geomSteps: int=2500, sampleRingBonds: bool=False) → None

```

#### class OBFFConstraints

```

AddAngleConstraint (self: pyOBabel.pyOBabel.OBFFConstraints, a: int, b: int, c: int, angle: float) → None
AddAtomConstraint (self: pyOBabel.pyOBabel.OBFFConstraints, a: int) → None
AddAtomXConstraint (self: pyOBabel.pyOBabel.OBFFConstraints, a: int) → None
AddAtomYConstraint (self: pyOBabel.pyOBabel.OBFFConstraints, a: int) → None
AddAtomZConstraint (self: pyOBabel.pyOBabel.OBFFConstraints, a: int) → None
AddDistanceConstraint (self: pyOBabel.pyOBabel.OBFFConstraints, a: int, b: int, length: float) → None
AddIgnore (self: pyOBabel.pyOBabel.OBFFConstraints, a: int) → None
AddTorsionConstraint (self: pyOBabel.pyOBabel.OBFFConstraints, a: int, b: int, b: int, d: int, torsion: float) → None
Clear (self: pyOBabel.pyOBabel.OBFFConstraints) → None
DeleteConstraint (self: pyOBabel.pyOBabel.OBFFConstraints, index: int) → None
GetConstraintAtomA (self: pyOBabel.pyOBabel.OBFFConstraints, index: int) → int

```

**GetConstraintAtomB** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *index*: int) → int  
**GetConstraintAtomC** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *index*: int) → int  
**GetConstraintAtomD** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *index*: int) → int  
**GetConstraintEnergy** (*self*: pyOBabel.pyOBabel.OBFFConstraints) → float  
**GetConstraintType** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *index*: int) → int  
**GetConstraintValue** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *index*: int) → float  
**GetFactor** (*self*: pyOBabel.pyOBabel.OBFFConstraints) → float  
**GetFixedBitVec** (*self*: pyOBabel.pyOBabel.OBFFConstraints) → pyOBabel.pyOBabel.OBBitVec  
**GetGradient** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *a*: int) → pyOBabel.pyOBabel.vector3  
**GetIgnoredBitVec** (*self*: pyOBabel.pyOBabel.OBFFConstraints) → pyOBabel.pyOBabel.OBBitVec  
**IsFixed** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *a*: int) → bool  
**IsIgnored** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *a*: int) → bool  
**IsXFixed** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *a*: int) → bool  
**IsYFixed** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *a*: int) → bool  
**IsZFixed** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *a*: int) → bool  
**SetFactor** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *factor*: float) → None  
**Setup** (*self*: pyOBabel.pyOBabel.OBFFConstraints, *mol*: pyOBabel.pyOBabel.OBMol) → None

## OBError and OBMessageHandler

---

*OBError*

---

*OBMessageHandler*

---

## Documentation

### class OBError

**GetError** (*self*: pyOBabel.pyOBabel.OBError) → str  
**GetExplanation** (*self*: pyOBabel.pyOBabel.OBError) → str  
**GetLevel** (*self*: pyOBabel.pyOBabel.OBError) → pyOBabel.pyOBabel.obMessageLevel  
**GetMethod** (*self*: pyOBabel.pyOBabel.OBError) → str  
**GetPossibleCause** (*self*: pyOBabel.pyOBabel.OBError) → str  
**GetSuggestedRemedy** (*self*: pyOBabel.pyOBabel.OBError) → str  
**message** (*self*: pyOBabel.pyOBabel.OBError) → str

### class OBMessageHandler

**ClearLog** (*self*: pyOBabel.pyOBabel.OBMessageHandler) → None  
**GetAuditMessageCount** (*self*: pyOBabel.pyOBabel.OBMessageHandler) → int

---

```

GetDebugMessageCount (self: pyOBabel.pyOBabel.OBMessageHandler) → int
GetErrorMessageCount (self: pyOBabel.pyOBabel.OBMessageHandler) → int
GetInfoMessageCount (self: pyOBabel.pyOBabel.OBMessageHandler) → int
GetMaxLogEntries (self: pyOBabel.pyOBabel.OBMessageHandler) → int
GetMessageSummary (self: pyOBabel.pyOBabel.OBMessageHandler) → str
GetMessagesOfLevel (self: pyOBabel.pyOBabel.OBMessageHandler, arg0: pyOBabel.pyOBabel.obMessageLevel) → List[str]
GetOutputLevel (self: pyOBabel.pyOBabel.OBMessageHandler) → pyOBabel.pyOBabel.obMessageLevel
GetOutputStream (self: pyOBabel.pyOBabel.OBMessageHandler) → object
GetWarningMessageCount (self: pyOBabel.pyOBabel.OBMessageHandler) → int
SetMaxLogEntries (self: pyOBabel.pyOBabel.OBMessageHandler, arg0: int) → None
SetOutputLevel (self: pyOBabel.pyOBabel.OBMessageHandler, arg0: pyOBabel.pyOBabel.obMessageLevel) → None
SetOutputStream (self: pyOBabel.pyOBabel.OBMessageHandler, arg0: object) → None
StartLogging (self: pyOBabel.pyOBabel.OBMessageHandler) → None
StopErrorWrap (self: pyOBabel.pyOBabel.OBMessageHandler) → bool
ThrowErrorByMessage (self: pyOBabel.pyOBabel.OBMessageHandler, method: str, errorMsg: str, level: pyOBabel.pyOBabel.obMessageLevel=obMessageLevel.obDebug, qualifier: pyOBabel.pyOBabel.errorQualifier=errorQualifier.always) → None
ThrowErrorByOSError (self: pyOBabel.pyOBabel.OBMessageHandler, err: pyOBabel.pyOBabel.OSError, qualifier: pyOBabel.pyOBabel.errorQualifier=errorQualifier.always) → None

```



## CHAPTER 4

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



**p**

pyOBabel, [21](#)





**A**

AddAngleConstraint() (OBFFConstraints method), 29  
AddAtom() (OBMol method), 8  
AddAtomConstraint() (OBFFConstraints method), 29  
AddAtomXConstraint() (OBFFConstraints method), 29  
AddAtomYConstraint() (OBFFConstraints method), 29  
AddAtomZConstraint() (OBFFConstraints method), 29  
AddBond() (OBAtom method), 13  
AddBond() (OBMol method), 8  
AddBondBetweenAtoms() (OBMol method), 8  
AddChemObject() (OBConversion method), 24  
AddConformerCoords() (OBMol method), 8  
AddDistanceConstraint() (OBFFConstraints method), 29  
AddHydrogens() (OBMol method), 8  
AddIgnore() (OBFFConstraints method), 29  
AddInterGroup() (OBForceField method), 26  
AddInterGroups() (OBForceField method), 26  
AddIntraGroup() (OBForceField method), 26  
AddNbrs() (OBBuilder method), 25  
AddNewHydrogens() (OBMol method), 8  
AddNonPolarHydrogens() (OBMol method), 8  
AddOption() (OBConversion method), 24  
AddPolarHydrogens() (OBMol method), 8  
AddResidue() (OBAtom method), 13  
AddResidue() (OBMol method), 8  
AddTorsionConstraint() (OBFFConstraints method), 29  
Align() (OBMol method), 8  
AreInSameRing() (OBMol method), 8  
AssignSpinMultiplicity() (OBMol method), 9  
AssignTotalChargeToAtoms() (OBMol method), 9  
atoms (OBMol attribute), 12  
AutomaticFormalCharge() (OBMol method), 9  
AutomaticPartialCharge() (OBMol method), 9  
AverageBondAngle() (OBAtom method), 13

**B**

BeginModify() (OBMol method), 9  
BitIsOn() (OBBitVec method), 23  
bonds (OBMol attribute), 12

BOSum() (OBAtom method), 13  
Build() (OBBuilder method), 25

**C**

CalcTorsionAngle() (in module pyOBabel), 22  
CanBeNormalized() (vector3 method), 19  
Center() (OBMol method), 9  
CenterOfConformer() (OBMol method), 9  
Clear() (OBAtom method), 13  
Clear() (OBBitVec method), 23  
Clear() (OBFFConstraints method), 29  
Clear() (OBMol method), 9  
ClearBond() (OBAtom method), 13  
ClearGroups() (OBForceField method), 26  
ClearLog() (OBMessageHandler method), 30  
CloseOutFile() (OBConversion method), 24  
ConjugateGradients() (OBForceField method), 26  
ConjugateGradientsInitialize() (OBForceField method), 26  
ConjugateGradientsTakeNSteps() (OBForceField method), 26  
Connect() (OBBuilder method), 25  
ConnectTheDots() (OBMol method), 9  
ContigFragList() (OBMol method), 9  
Convert() (OBConversion method), 24  
ConvertDativeBonds() (OBMol method), 9  
ConvertZeroBonds() (OBMol method), 9  
copy() (OBMol method), 13  
copy() (vector3 method), 21  
CorrectForPH() (OBMol method), 9  
CorrectStereoAtoms() (OBBuilder method), 26  
CorrectVelocities() (OBForceField method), 26  
CountBits() (OBBitVec method), 23  
CountBondsOfOrder() (OBAtom method), 13  
CountFreeOxygens() (OBAtom method), 13  
CountFreeSulfurs() (OBAtom method), 13  
CountRingBonds() (OBAtom method), 13  
createOrthoVector() (vector3 method), 21  
cross() (in module pyOBabel), 22

## D

DeleteAtom() (OBMol method), 9  
DeleteBond() (OBMol method), 9  
DeleteConformerByIndex() (OBMol method), 9  
DeleteConstraint() (OBFFConstraints method), 29  
DeleteHydrogen() (OBMol method), 9  
DeleteHydrogensAll() (OBMol method), 9  
DeleteHydrogensFromAtom() (OBMol method), 9  
DeleteNonPolarHydrogens() (OBMol method), 9  
DeletePolarHydrogens() (OBMol method), 9  
DeleteResidue() (OBAAtom method), 13  
DeleteResidue() (OBMol method), 9  
DetectExplosion() (OBForceField method), 26  
DistanceGeometry() (OBForceField method), 26  
distSq() (vector3 method), 21  
dot() (in module pyOLabel), 21  
Duplicate() (OBAAtom method), 13

## E

E\_Angle() (OBForceField method), 26  
E\_Bond() (OBForceField method), 27  
E\_Electrostatic() (OBForceField method), 27  
E\_OOP() (OBForceField method), 27  
E\_StrBnd() (OBForceField method), 27  
E\_Torsion() (OBForceField method), 27  
E\_VDW() (OBForceField method), 27  
Empty() (OBMol method), 9  
EnableAllPairs() (OBForceField method), 27  
EnableCutOff() (OBForceField method), 27  
EndBit() (OBBitVec method), 23  
EndModify() (OBMol method), 9  
Energy() (OBForceField method), 27  
ExplicitHydrogenCount() (OBAAtom method), 13

## F

FastRotorSearch() (OBForceField method), 27  
FindAngles() (OBMol method), 9  
FindChiralCenters() (OBMol method), 9  
FindForceField() (OBForceField method), 27  
FindLSSR() (OBMol method), 9  
FindRingAtomsAndBonds() (OBMol method), 9  
FindSSSR() (OBMol method), 9  
FindTorsions() (OBMol method), 9  
FindType() (OBForceField method), 27  
FirstBit() (OBBitVec method), 23  
Fold() (OBBitVec method), 23  
FromString() (OBBitVec method), 24  
FromVecInt() (OBBitVec method), 24  
FullConvert() (OBConversion method), 24

## G

GenerateVelocities() (OBForceField method), 27  
Get() (vector3 method), 20

GetAngle() (OBAAtom method), 13  
GetAngle() (OBMol method), 9  
GetAngleBetweenAtoms() (OBAAtom method), 13  
GetAtomicMass() (OBAAtom method), 13  
GetAtomicNum() (OBAAtom method), 14  
GetAtomTypes() (OBForceField method), 27  
GetAuditMessageCount() (OBMessageHandler method), 30  
GetBeginAtom() (OBBond method), 16  
GetBeginAtomIdx() (OBBond method), 16  
GetBO() (OBBond method), 16  
GetBond() (OBAAtom method), 14  
GetBondOrder() (OBBond method), 17  
GetChemObject() (OBConversion method), 24  
GetConformerCoordsByIndex() (OBMol method), 9  
GetConformerEnergies() (OBMol method), 9  
GetConformerEnergy() (OBMol method), 10  
GetConformers() (OBForceField method), 27  
GetConformersAllCoords() (OBMol method), 10  
GetConstraintAtomA() (OBFFConstraints method), 29  
GetConstraintAtomB() (OBFFConstraints method), 29  
GetConstraintAtomC() (OBFFConstraints method), 30  
GetConstraintAtomD() (OBFFConstraints method), 30  
GetConstraintEnergy() (OBFFConstraints method), 30  
GetConstraints() (OBForceField method), 27  
GetConstraintType() (OBFFConstraints method), 30  
GetConstraintValue() (OBFFConstraints method), 30  
GetCoordinates() (OBForceField method), 27  
GetCoordinates() (OBMol method), 10  
GetCount() (OBConversion method), 24  
GetDebugMessageCount() (OBMessageHandler method), 30  
GetDimension() (OBMol method), 10  
GetDistanceFromAtom() (OBAAtom method), 14  
GetDistanceFromIndex() (OBAAtom method), 14  
GetDistanceFromVector() (OBAAtom method), 14  
GetElectrostaticCutOff() (OBForceField method), 27  
GetEndAtom() (OBBond method), 17  
GetEndAtomIdx() (OBBond method), 17  
GetEquibLength() (OBBond method), 17  
GetError() (OSError method), 30  
GetErrorMessageCount() (OBMessageHandler method), 31  
GetExactMass() (OBAAtom method), 14  
GetExactMass() (OBMol method), 10  
GetExplanation() (OSError method), 30  
GetFactor() (OBFFConstraints method), 30  
GetFixedBitVec() (OBFFConstraints method), 30  
GetFlags() (OBBond method), 17  
GetFormalCharge() (OBAAtom method), 14  
GetFormula() (OBMol method), 10  
GetFragment() (OBBuilder method), 26  
GetGradient() (OBFFConstraints method), 30  
GetHeatFormation() (OBMol method), 10

- GetHeteroValence() (OBAtom method), 14
  - GetHvyValence() (OBAtom method), 14
  - GetHyb() (OBAtom method), 14
  - GetId() (OBAtom method), 14
  - GetId() (OBBond method), 17
  - GetIdx() (OBAtom method), 14
  - GetIdx() (OBBond method), 17
  - GetIgnoredBitVec() (OBFFConstraints method), 30
  - GetImplicitHCount() (OBAtom method), 14
  - GetIndex() (OBAtom method), 14
  - GetInfoMessageCount() (OBMessageHandler method), 31
  - GetIsotope() (OBAtom method), 14
  - GetLength() (OBBond method), 17
  - GetLevel() (OSError method), 30
  - GetLineSearchType() (OBForceField method), 27
  - GetLogLevel() (OBForceField method), 27
  - GetMaxLogEntries() (OBMessageHandler method), 31
  - GetMessagesOfLevel() (OBMessageHandler method), 31
  - GetMessageSummary() (OBMessageHandler method), 31
  - GetMethod() (OSError method), 30
  - GetMolWt() (OBMol method), 10
  - GetName() (OBResidue method), 18
  - GetNbrAtom() (OBBond method), 17
  - GetNbrAtomIdx() (OBBond method), 17
  - GetNewBondVector() (OBAtom method), 14
  - GetNewBondVector() (OBBuilder method), 26
  - GetNextAtom() (OBAtom method), 14
  - GetNumElectrostaticPairs() (OBForceField method), 27
  - GetNumPairs() (OBForceField method), 27
  - GetNumVDWPairs() (OBForceField method), 27
  - GetOptions() (OBConversion method), 24
  - GetOutputIndex() (OBConversion method), 24
  - GetOutputLevel() (OBMessageHandler method), 31
  - GetOutputStream() (OBMessageHandler method), 31
  - GetParent() (OBAtom method), 14
  - GetParent() (OBBond method), 17
  - GetPartialCharge() (OBAtom method), 14
  - GetPossibleCause() (OSError method), 30
  - GetResidue() (OBAtom method), 14
  - GetSize() (OBBitVec method), 24
  - GetSpacedFormula() (OBMol method), 10
  - GetSpinMultiplicity() (OBAtom method), 14
  - GetSuggestedRemedy() (OSError method), 30
  - GetSupportedInputFormat() (OBConversion method), 24
  - GetSupportedOutputFormat() (OBConversion method), 24
  - GetTitle() (OBMol method), 10
  - GetTorsionByAtoms() (OBMol method), 10
  - GetTorsionByIndex() (OBMol method), 10
  - GetTotalCharge() (OBMol method), 10
  - GetTotalSpinMultiplicity() (OBMol method), 10
  - GetType() (OBAtom method), 14
  - GetUnit() (OBForceField method), 27
  - GetUpdateFrequency() (OBForceField method), 27
  - GetValence() (OBAtom method), 14
  - GetVDWCutOff() (OBForceField method), 27
  - GetVector() (OBAtom method), 14
  - GetWarningMessageCount() (OBMessageHandler method), 31
  - GetX() (OBAtom method), 14
  - GetX() (vector3 method), 20
  - GetY() (OBAtom method), 14
  - GetY() (vector3 method), 20
  - GetZ() (OBAtom method), 14
  - GetZ() (vector3 method), 20
- ## H
- Has2D() (OBMol method), 10
  - Has3D() (OBMol method), 10
  - HasAlphaBetaUnsat() (OBAtom method), 14
  - HasAnalyticalGradients() (OBForceField method), 27
  - HasAromaticBond() (OBAtom method), 14
  - HasAromaticPerceived() (OBMol method), 10
  - HasAtomTypesPerceived() (OBMol method), 10
  - HasBondOfOrder() (OBAtom method), 14
  - HasChainsPerceived() (OBMol method), 10
  - HasChiralityPerceived() (OBMol method), 10
  - HasChiralitySpecified() (OBAtom method), 14
  - HasChiralVolume() (OBAtom method), 14
  - HasClosureBondsPerceived() (OBMol method), 10
  - HasDoubleBond() (OBAtom method), 14
  - HasGroups() (OBForceField method), 27
  - HasHybridizationPerceived() (OBMol method), 10
  - HasHydrogensAdded() (OBMol method), 10
  - HasLSSRPerceived() (OBMol method), 10
  - HasNonSingleBond() (OBAtom method), 14
  - HasNonZeroCoords() (OBMol method), 10
  - HasPartialChargesPerceived() (OBMol method), 10
  - HasResidue() (OBAtom method), 15
  - HasRingAtomsAndBondsPerceived() (OBMol method), 10
  - HasRingTypesPerceived() (OBMol method), 10
  - HasSingleBond() (OBAtom method), 15
  - HasSpinMultiplicityAssigned() (OBMol method), 10
  - HasSSSRPerceived() (OBMol method), 10
  - HighestBondOrder() (OBAtom method), 15
  - HtoMethyl() (OBAtom method), 15
- ## I
- InsertAtom() (OBMol method), 10
  - IsAmide() (OBBond method), 17
  - IsAmideNitrogen() (OBAtom method), 15
  - IsAntiClockwise() (OBAtom method), 15
  - IsApprox() (vector3 method), 20
  - IsAromatic() (OBAtom method), 15
  - IsAromatic() (OBBond method), 17

IsAromaticNOxide() (OBAtom method), 15  
IsAxial() (OBAtom method), 15  
IsCarbonyl() (OBBond method), 17  
IsCarboxylOxygen() (OBAtom method), 15  
IsChiral() (OBAtom method), 15  
IsChiral() (OBMol method), 10  
IsCisOrTrans() (OBBond method), 17  
IsClockwise() (OBAtom method), 15  
IsClosure() (OBBond method), 17  
IsConnected() (OBAtom method), 15  
IsCorrectedForPH() (OBMol method), 10  
IsCutOffEnabled() (OBForceField method), 27  
IsDoubleBondGeometry() (OBBond method), 17  
IsDown() (OBBond method), 17  
IsEmpty() (OBBitVec method), 24  
IsEster() (OBBond method), 17  
IsFirstInput() (OBConversion method), 24  
IsFixed() (OBFFConstraints method), 30  
IsHash() (OBBond method), 17  
IsHbondAcceptor() (OBAtom method), 15  
IsHbondAcceptorSimple() (OBAtom method), 15  
IsHbondDonor() (OBAtom method), 15  
IsHbondDonorH() (OBAtom method), 15  
IsHetAtom() (OBAtom method), 15  
IsHeteroatom() (OBAtom method), 15  
IsIgnored() (OBFFConstraints method), 30  
IsInRing() (OBAtom method), 15  
IsInRing() (OBBond method), 17  
IsInRingSize() (OBAtom method), 15  
IsLast() (OBConversion method), 24  
IsLastFile() (OBConversion method), 24  
IsNegativeStereo() (OBAtom method), 15  
IsNitroOxygen() (OBAtom method), 15  
IsNonPolarHydrogen() (OBAtom method), 15  
IsOneFour() (OBAtom method), 15  
IsOneThree() (OBAtom method), 15  
IsOption() (OBConversion method), 25  
IsPhosphateOxygen() (OBAtom method), 15  
IsPolarHydrogen() (OBAtom method), 15  
IsPositiveStereo() (OBAtom method), 15  
IsPrimaryAmide() (OBBond method), 17  
IsRotor() (OBBond method), 17  
IsSecondaryAmide() (OBBond method), 17  
IsSetupNeeded() (OBForceField method), 27  
IsSpiroAtom() (OBBuilder method), 26  
IsSulfateOxygen() (OBAtom method), 15  
IsTertiaryAmide() (OBBond method), 17  
IsUp() (OBBond method), 17  
IsWedge() (OBBond method), 17  
IsWedgeOrHash() (OBBond method), 17  
IsXFixed() (OBFFConstraints method), 30  
IsYFixed() (OBFFConstraints method), 30  
IsZFixed() (OBFFConstraints method), 30

## L

length() (vector3 method), 21  
length\_2() (vector3 method), 21  
LewisAcidBaseCounts() (OBAtom method), 15  
LineSearch() (OBForceField method), 27  
LoadFragments() (OBBuilder method), 26

## M

MakeDativeBonds() (OBMol method), 10  
MakeNewInstance() (OBForceField method), 27  
MatchesSMARTS() (OBAtom method), 15  
MemberOfRingCount() (OBAtom method), 15  
MemberOfRingSize() (OBAtom method), 15  
message() (OSError method), 30  
MolecularDynamicsTakeNSteps() (OBForceField method), 27

## N

Negate() (OBBitVec method), 24  
NewAtom() (OBMol method), 11  
NewAtomById() (OBMol method), 11  
NewBond() (OBMol method), 11  
NewBondById() (OBMol method), 11  
NewResidue() (OBAtom method), 15  
NewResidue() (OBMol method), 11  
NextBit() (OBBitVec method), 24  
normalize() (vector3 method), 21  
NumAtoms() (OBMol method), 11  
NumBonds() (OBMol method), 11  
NumConformers() (OBMol method), 11  
NumHvyAtoms() (OBMol method), 11  
NumResidues() (OBMol method), 11  
NumRotors() (OBMol method), 11

## O

OBAtom (class in pyOBabel), 13  
OBBitVec (class in pyOBabel), 23  
OBBond (class in pyOBabel), 16  
OBBuilder (class in pyOBabel), 25  
OBConversion (class in pyOBabel), 24  
OSError (class in pyOBabel), 30  
OBFFConstraints (class in pyOBabel), 29  
OBFFLog() (OBForceField method), 28  
OBForceField (class in pyOBabel), 26  
OBMessageHandler (class in pyOBabel), 30  
OBMol (class in pyOBabel), 8  
OBResidue (class in pyOBabel), 18  
OpenInAndOutFiles() (OBConversion method), 25

## P

ParseParamFile() (OBForceField method), 28  
PerceiveBondOrders() (OBMol method), 11  
Point2Line() (in module pyOBabel), 23



Point2Plane() (in module pyOBabel), 22  
 Point2PlaneAngle() (in module pyOBabel), 23  
 Point2PlaneSigned() (in module pyOBabel), 22  
 PrintFormalCharges() (OBForceField method), 28  
 PrintPartialCharges() (OBForceField method), 28  
 PrintTypes() (OBForceField method), 28  
 PrintVelocities() (OBForceField method), 28  
 pyOBabel (module), 21

## R

RandomRotorSearch() (OBForceField method), 28  
 RandomRotorSearchInitialize() (OBForceField method), 28  
 RandomRotorSearchNextConformer() (OBForceField method), 28  
 Read() (OBConversion method), 25  
 ReadFile() (OBConversion method), 25  
 ReadString() (OBConversion method), 25  
 RemoveOption() (OBConversion method), 25  
 RenumberAtomsByAtoms() (OBMol method), 11  
 RenumberAtomsByIndex() (OBMol method), 11  
 ReserveAtoms() (OBMol method), 11  
 residues (OBMol attribute), 13  
 Resize() (OBBitVec method), 24  
 ResizeWords() (OBBitVec method), 24

## S

Separate() (OBMol method), 11  
 Set() (vector3 method), 20  
 SetAntiClockwiseStereo() (OBAtom method), 15  
 SetAromatic() (OBAtom method), 16  
 SetAromatic() (OBBond method), 17  
 SetAromaticPerceived() (OBMol method), 11  
 SetAtomicNum() (OBAtom method), 16  
 SetAtomTypesPerceived() (OBMol method), 11  
 SetAutomaticFormalCharge() (OBMol method), 11  
 SetAutomaticPartialCharge() (OBMol method), 11  
 SetBegin() (OBBond method), 17  
 SetBitOff() (OBBitVec method), 24  
 SetBitOn() (OBBitVec method), 24  
 SetBO() (OBBond method), 17  
 SetBondOrder() (OBBond method), 17  
 SetChainsPerceived() (OBMol method), 11  
 SetChiral() (OBAtom method), 16  
 SetChiralityPerceived() (OBMol method), 11  
 SetClockwiseStereo() (OBAtom method), 16  
 SetClosure() (OBBond method), 17  
 SetClosureBondsPerceived() (OBMol method), 11  
 SetConformerByIndex() (OBMol method), 11  
 SetConformerEnergies() (OBMol method), 11  
 SetConformers() (OBForceField method), 28  
 SetConformersCoords() (OBMol method), 11  
 SetConstraints() (OBForceField method), 28  
 SetCoordinates() (OBForceField method), 28

SetCoordinates() (OBMol method), 11  
 SetCorrectedForPH() (OBMol method), 11  
 SetDimension() (OBMol method), 11  
 SetDown() (OBBond method), 17  
 SetElectrostaticCutOff() (OBForceField method), 28  
 SetEnd() (OBBond method), 17  
 SetFactor() (OBFFConstraints method), 30  
 SetFirstInput() (OBConversion method), 25  
 SetFixAtom() (OBForceField method), 28  
 SetFlags() (OBMol method), 11  
 SetFormalCharge() (OBAtom method), 16  
 SetFormalCharges() (OBForceField method), 28  
 SetFormula() (OBMol method), 11  
 SetHash() (OBBond method), 18  
 SetHeatFormation() (OBMol method), 11  
 SetHyb() (OBAtom method), 16  
 SetHybAndGeom() (OBAtom method), 16  
 SetHybridizationPerceived() (OBMol method), 11  
 SetHydrogensAdded() (OBMol method), 11  
 SetId() (OBAtom method), 16  
 SetId() (OBBond method), 18  
 SetIdx() (OBAtom method), 16  
 SetIgnoreAtom() (OBForceField method), 28  
 SetImplicitHCount() (OBAtom method), 16  
 SetInAndOutFormats() (OBConversion method), 25  
 SetInFormat() (OBConversion method), 25  
 SetInRing() (OBAtom method), 16  
 SetInRing() (OBBond method), 18  
 SetIsotope() (OBAtom method), 16  
 SetIsPatternStructure() (OBMol method), 11  
 SetKeepRings() (OBBuilder method), 26  
 SetLast() (OBConversion method), 25  
 SetLength() (OBBond method), 18  
 SetLineSearchType() (OBForceField method), 28  
 SetLogLevel() (OBForceField method), 28  
 SetLSSRPerceived() (OBMol method), 11  
 SetMaxLogEntries() (OBMessageHandler method), 31  
 SetMoreFilesToCome() (OBConversion method), 25  
 SetName() (OBResidue method), 18  
 SetNegativeStereo() (OBAtom method), 16  
 SetOneObjectOnly() (OBConversion method), 25  
 SetOptions() (OBConversion method), 25  
 SetOutFormat() (OBConversion method), 25  
 SetOutputIndex() (OBConversion method), 25  
 SetOutputLevel() (OBMessageHandler method), 31  
 SetOutputStream() (OBMessageHandler method), 31  
 SetParameterFile() (OBForceField method), 28  
 SetParent() (OBAtom method), 16  
 SetParent() (OBBond method), 18  
 SetPartialCharge() (OBAtom method), 16  
 SetPartialCharges() (OBForceField method), 28  
 SetPartialChargesPerceived() (OBMol method), 12  
 SetPositiveStereo() (OBAtom method), 16  
 SetRangeOff() (OBBitVec method), 24

SetRangeOn() (OBBitVec method), 24  
SetResidue() (OAtom method), 16  
SetRingAtomsAndBondsPerceived() (OBMol method), 12  
SetRingTypesPerceived() (OBMol method), 12  
SetSpinMultiplicity() (OAtom method), 16  
SetSpinMultiplicityAssigned() (OBMol method), 12  
SetSSSRPerceived() (OBMol method), 12  
SetTitle() (OBMol method), 12  
SetTorsion() (OBMol method), 12  
SetTotalCharge() (OBMol method), 12  
SetTotalSpinMultiplicity() (OBMol method), 12  
SetType() (OAtom method), 16  
SetTypes() (OBForceField method), 28  
SetUp() (OBBond method), 18  
Setup() (OBFFConstraints method), 30  
Setup() (OBForceField method), 28  
SetupCalculations() (OBForceField method), 28  
SetupUpdateFrequency() (OBForceField method), 28  
SetupPointers() (OBForceField method), 28  
SetupWithConstraints() (OBForceField method), 28  
SetVDWCutOff() (OBForceField method), 28  
SetVector() (OAtom method), 16  
SetWedge() (OBBond method), 18  
SetWedgeOrHash() (OBBond method), 18  
SetX() (vector3 method), 20  
SetY() (vector3 method), 20  
SetZ() (vector3 method), 20  
SmallestBondAngle() (OAtom method), 16  
StartLogging() (OBMessageHandler method), 31  
SteepestDescent() (OBForceField method), 28  
SteepestDescentInitialize() (OBForceField method), 28  
SteepestDescentTakeNSteps() (OBForceField method), 28  
StopErrorWrap() (OBMessageHandler method), 31  
StripSalts() (OBMol method), 12  
Swap() (OBBuilder method), 26  
SystematicRotorSearch() (OBForceField method), 29  
SystematicRotorSearchInitialize() (OBForceField method), 29  
SystematicRotorSearchNextConformer() (OBForceField method), 29

## T

ThrowErrorByMessage() (OBMessageHandler method), 31  
ThrowErrorByOError() (OBMessageHandler method), 31  
ToInertialFrameForAllConformers() (OBMol method), 12  
ToInertialFrameForOneConformer() (OBMol method), 12  
TranslateAllConformers() (OBMol method), 12  
TranslateOneConformer() (OBMol method), 12

## U

UnsetAromatic() (OAtom method), 16  
UnsetAromatic() (OBBond method), 18  
UnsetAromaticPerceived() (OBMol method), 12  
UnsetDown() (OBBond method), 18  
UnsetFixAtom() (OBForceField method), 29  
UnsetFlag() (OBMol method), 12  
UnsetHash() (OBBond method), 18  
UnsetHydrogensAdded() (OBMol method), 12  
UnsetIgnoreAtom() (OBForceField method), 29  
UnsetKeepRings() (OBBuilder method), 26  
UnsetLSSRPerceived() (OBMol method), 12  
UnsetPartialChargesPerceived() (OBMol method), 12  
UnsetRingTypesPerceived() (OBMol method), 12  
UnsetSSSRPerceived() (OBMol method), 12  
UnsetStereo() (OAtom method), 16  
UnsetUp() (OBBond method), 18  
UnsetWedge() (OBBond method), 18  
UpdateConformers() (OBForceField method), 29  
UpdateCoordinates() (OBForceField method), 29  
UpdatePairsSimple() (OBForceField method), 29

## V

Validate() (OBForceField method), 29  
ValidateConjugateGradients() (OBForceField method), 29  
ValidateGradientError() (OBForceField method), 29  
ValidateGradients() (OBForceField method), 29  
ValidateLineSearch() (OBForceField method), 29  
ValidateSteepestDescent() (OBForceField method), 29  
vector3 (class in pyOLabel), 19  
vectorAngle() (in module pyOLabel), 22

## W

WeightedRotorSearch() (OBForceField method), 29  
Write() (OBConversion method), 25  
WriteFile() (OBConversion method), 25  
WriteString() (OBConversion method), 25