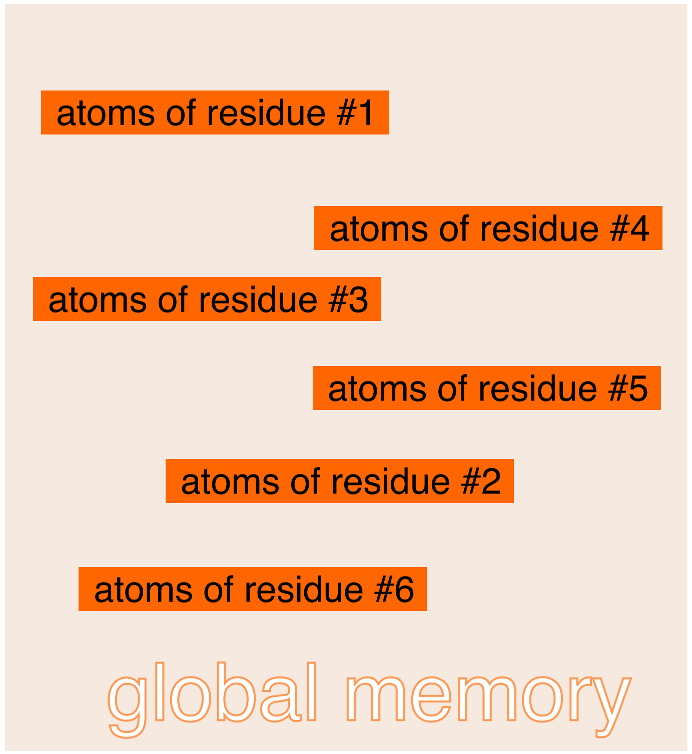
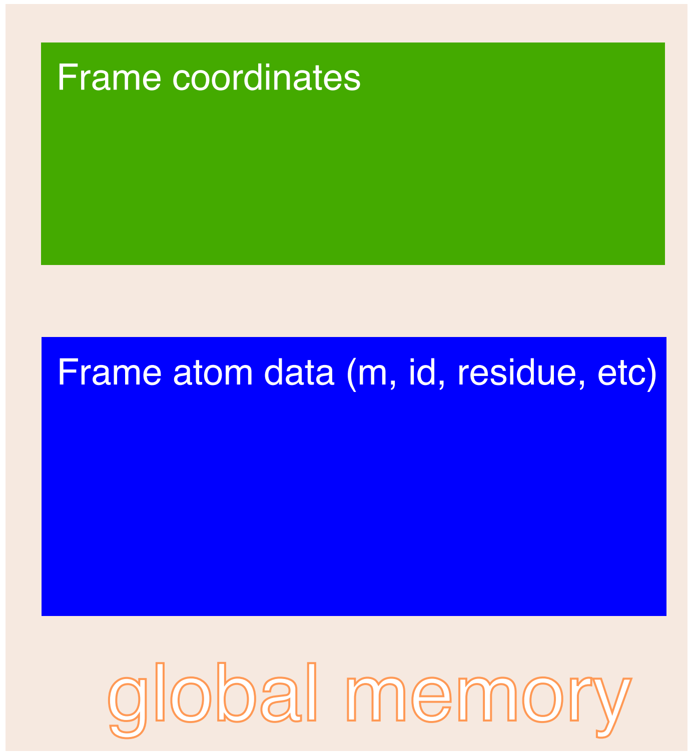
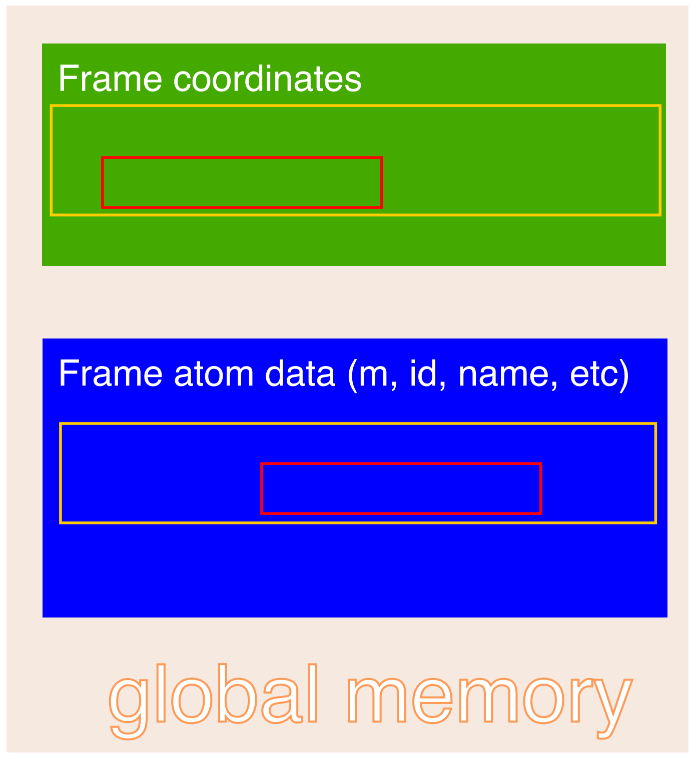
```
Frame
*-Chain
  *-Residue
    *-Atom(r,m,id)
```

```
Frame
|-Molecule
|-Residue
|-Atom (m, id)
|-Coords
```



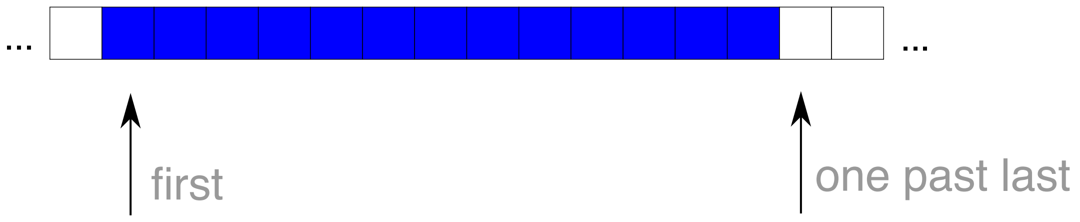


```
Frame
|-Molecule
|-Residue
|-Atom (m, id)
|-Coords
```



Span

`frame.residues[1:10]`

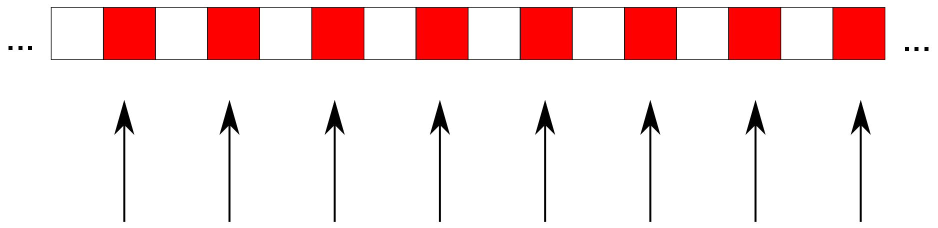


always two pointers

O(1) operation complexity
(constant time)

Selection

`frame.residues[::2]`



n pointers

O(n) operation complexity
(linear time)

Migration highlights

1. no more submodules (almost)

```
from pyxmolpp2.geometry import XYZ
from pyxmolpp2 import XYZ
```

2. chain renamed to molecule

```
residue.chain
residue.molecule
```

3. asXxxx renamed to xxxx

```
residue.asAtoms
residue.atoms
```

4. XYZ is immutable object (like str)

```
atom.r.x += 5
atom.r += XYZ(5,0,0)
```

5. New type of selection: coords

```
frame.coords
atoms.coords
```

6. XxxxName classes are removed from python side

```
atom.name = AtomName("CA")
atom.name = "CA"
```

7. Conversion to numpy array

```
crds = frame.asAtoms.toCoords.to_numpy()
crds = frame.coords.values

crd = atom.r.to_np
crd = atom.r.values
```

8. Reading PDB

```
frame = PdbFile("1.pdb", altered_records).get_frame()
frame = PdbFile("1.pdb").frames()[0]
# uses AMBER convention by default
```

9. UnitCell = LatticeVectors + BestShiftFinder

```
cell = UnitCell(...)
cell.scale_by(1.05)
var_img = cell.closet_image_to(ref, var)
```

10. free functions are replaced with methods

```
dr = distance(N.r, H.r)
dr = N.r.distance(H.r)
# same for angle & dihedral angle
```

New features

Atoms have mass!

```
atoms = frame.atoms
atoms.guess_mass() # good enough for proteins & nucleic
print(atom[0].mass)
```

Weighted variants of alignment & rmsd

```
alignment = atoms.alignment_to(other, weighted=True)
rmsd = atoms.rmsd(other, weighted=True)
```

Note: by default atoms aligment is non-weighted

Note: aligment of coordinates is *always* non-weighted

Trajectory piping (experimental)

Idea: wrap repeated trajectory (pre)processing

Example: pre-align trajectory frames by "CA" atoms

```
from pyxmolpp2.pipe import Align

for frame in traj[::100] | Align(by=(aName == "CA")):
    print(frame.coords.mean())
```