ViewerManual

July 9, 2023

1 Py3Dmol

1.1 addArrow

`addArrow(spec)`

Name	Type	Description
spec	ArrowSpec	Style specification

Name	Type	Description
start	$3Dmol.Vector3	
end	$3Dmol.Vector3	
radius	number	
color	ColorSpec	solid color
hidden	boolean	
rtadiusRatio	number	1.618034
mid	number	0.618034
midpos	number	,
alpha	number	
wireframe	boolean	
linewidth	number	
clickable	boolean	callback
callback	function	
frame	number	

[ ]:
```
import py3Dmol
view=py3Dmol.view(query="pdb:4DM7")
view.setBackgroundColor("0xffffffff")
view.addArrow({
    "start":{"x":-10.0,"y":0.0,"z":0.0},
    "end":{"x":0.0,"y":-10.0,"z":0.0},
    "radius":1.0,
    "radiusRatio":1.0,
    "mid":1.0,
    "clickable":True,
```

```
    "callback": '''function(){
                        this.color.setHex(0xFF0000FF);
                        viewer.render( );
                    }''',
})
view.render()
```

1.2 addAsOneMolecule

`addAsOneMolecule(data, format)` viewer

Name	Type	Description
data	string	
format	string	

1.2.1 FileFormats

Name	Type	Description
cdjson,json		Chemical JSON format
cube		Gaussian cube format
gro		Gromacs topology format, need to add coordinates to resulting model.
mcif,cif		Crystallographic Information File, the successor to PDB that makes you miss the PDB file format
mmtf		Macromolecular Transmission Format, the successor to PDB that is totally awesome
mol2		Sybyl Mol2 format
pdb		The venerable Protein Data Bank format
pqr		Like PDB but with partial charges which are read into the partialcharge atom property
prmtop		Amber topology file, must add coordinates
sdf		MDL MOL format, supports multiple models and meta data
vasp		VASP format (CONTCAR, POSCAR)
xyz		XYZ cartesian coordinates format

[ ]:
```
import py3Dmol
benz='''
     RDKit          3D

  6  6  0  0  0  0  0  0  0  0999 V2000
   -0.9517    0.7811   -0.6622 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.2847    1.3329   -0.3121 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.2365    0.5518    0.3512 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.9517   -0.7811    0.6644 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.2847   -1.3329    0.3144 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.2365   -0.5518   -0.3489 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  2  0
  2  3  1  0
  3  4  2  0
  4  5  1  0
  5  6  2  0
  6  1  1  0
M  END
$$$$'''
view=py3Dmol.view()
view.addAsOneMolecule(benz, "sdf")
view.setStyle({"stick":{}})
view.zoomTo()
view.render()
```

1.3 addBox

`addBox(spec)`

Name	Type	Description
spec	BoxSpec	

1.3.1 BoxSpec

Name	Type	Description
corner	$3Dmol.Vector3	
center	$3Dmol.Vector3	corner ( )
dimensions	Object	{w:width, h:height, d:depth}; can be either scalars or vectors (for not-axis aligned boxes)

[ ]:
```
import py3Dmol
view=py3Dmol.view()
```

```
view.addLine({"color":'red',"start":{"x":0,"y":0,"z":0},"end":{"x":5,"y":0,"z":
↪0}})
view.addLine({"color":'blue',"start":{"x":0,"y":0,"z":0},"end":{"x":0,"y":5,"z":
↪0}})
view.addLine({"color":'green',"start":{"x":0,"y":0,"z":0},"end":{"x":0,"y":
↪0,"z":5}})

view.addBox({"center":{"x":0,"y":0,"z":0},"dimensions": {"w":3,"h":4,"d":
↪2},"color":'magenta'})
view.zoomTo()
view.rotate(45, {"x":1,"y":1,"z":1})
view.render()
```

1.4 addCurve

`addCurve(spec)`

Name	Type	Description
spec	CurveSpec	

1.4.1 CurveSpec

Name	Type	Description
points	$3Dmol.Vector3	list of (x,y,z) points to interpolate between to make curve
smooth	number	shuamount of interpolation
radius	number	
fromArrow	boolean	
toArrow	boolean	

[ ]:
```
import py3Dmol
view=py3Dmol.view()
view.addCurve({"points":[
    {"x":0.0,"y":0.0,"z":0.0},
    {"x":5.0,"y":3.0,"z":0.0},
    {"x":5.0,"y":7.0,"z":0.0},
    {"x":0.0,"y":10.0,"z":0.0}],
    "radius":0.5,
    "smooth":100,
    "fromArrow":False,
    "toArrow":True,
    "color":"orange"})
```

```
view.addCurve({"points":[
    {"x":-1.0,"y":0.0,"z":0.0},
    {"x":-5.0,"y":5.0,"z":0.0},
    {"x":-2.0,"y":10.0,"z":0.0}],
    "radius":1,
    "fromArrow":True,
    "toArrow":False,
    "color":"purple"})
view.zoomTo()
view.render()
```

1.5 addCustom

addCustom(spec)

Name	Type	Description
spec	CustomSpec	

[ ]:
```
#
import py3Dmol
view=py3Dmol.view()
view.addCustom('''function triangle(view) {
    var vertices = [];
    var normals = [];
    var colors = [];
    var r = 20;
    //triangle
    vertices.push(new $3Dmol.Vector3(0,0,0));
    vertices.push(new $3Dmol.Vector3(r,0,0));
    vertices.push(new $3Dmol.Vector3(0,r,0));

    normals.push(new $3Dmol.Vector3(0,0,1));
    normals.push(new $3Dmol.Vector3(0,0,1));
    normals.push(new $3Dmol.Vector3(0,0,1));

    colors.push({r:1,g:0,b:0});
    colors.push({r:0,g:1,b:0});
    colors.push({r:0,g:0,b:1});

    var faces = [ 0,1,2 ];

    var spec = {vertexArr:vertices, normalArr: normals, faceArr:faces,color:
 ↪colors};
    view.addCustom(spec);
}'''
)
```

```
view.render()
```

1.6 addCylinder

`addCylinder(spec)`

Name	Type	Description
spec	CylinderSpec	

1.6.1 CylinderSpec

Name	Type	Description
start	$3Dmol.Vector3	
end	$3Dmol.Vector3	
radius	number	
fromCap	`$3Dmol.CAP`	0 for none, 1 for flat, 2 for round
toCap	`$3Dmol.CAP`	0 for none, 1 for flat, 2 for round
dashed	boolean	
color	ColorSpec	solid color
hidden	boolean	
alpha	number	
wireframe	boolean	
linewidth	number	
clickable	boolean	callback
callback	function	
frame	number	

[ ]:
```
import py3Dmol
view=py3Dmol.view()
view.setBackgroundColor("0xffffffff")
view.addCylinder({
    "start":{"x":0.0,"y":0.0,"z":0.0},
    "end":{"x":10.0,"y":0.0,"z":0.0},
    "radius":1.0,
    "fromCap":1,
    "toCap":2,
    "color":"red",
    "hoverable":True,
    "clickable":True,
    "callback":'''function(){ this.color.setHex(0x00FFFF00);viewer.render( );
 ↪}''',
    "hover_callback":'''function(){ viewer.render( );}''',
    "unhover_callback":'''function(){ this.color.setHex(0xFF000000);viewer.
 ↪render( );}'''
```

```
})
view.addCylinder({
    "start":{"x":0.0,"y":2.0,"z":0.0},
    "end":{"x":0.0,"y":10.0,"z":0.0},
    "radius":0.5,
    "fromCap":False,
    "toCap":True,
    "color":"teal",
})
view.addCylinder({
    "start":{"x":15.0,"y":0.0,"z":0.0},
    "end":{"x":20.0,"y":0.0,"z":0.0},
    "radius":1.0,
    "fromCap":False,
    "toCap":False,
    "color":"black",
})
view.render()
```

1.7 addIsosurface

`addIsosurface(data, spec)` `addVolumetricData`
py3Dmol

Name	Type	Description
data	$3Dmol.VolumeData	
spec	IsoSurfaceSpec	Shape style specification

1.7.1 IsoSurfaceSpec

Name	Type	Description
isoval	number	
color	ColorSpec	solid color
opacity	number	, between 0 and 1
wireframe	number	draw as wireframe, not surface
linewidth	number	
smoothness	number	amount to smooth surface (default 1)
coords	list	`viewer.selectedAtoms()` `AtomSelecti`
seldist	number	[= 2.0]
voldata	$3Dmol.VolumeData	`VolumeData` `volformat`
volscheme	$3Dmol.Gradient	
volformat	string	`$3Dmol.VolumeData` `voldata`

Name	Type	Description
clickable	boolean	callback
callback	function	

1.8 addLabel

`addLabel(text, options, sel, noshow)` Add label to viewer

Name	Type	Description
text	string	Label
options	LabelSpec	Label
sel	AtomSelection	
noshow	boolean	true -

1.8.1 LabelSpec

Name	Type	Description
font	string	sans-serif
fontSize	number	18
fontColor	ColorSpec	
fontOpacity	number	1
borderThickness	number	0
borderColor	ColorSpec	backgroundColor
borderOpacity	string	
backgroundColor	ColorSpec	
backgroundOpacity	string	1
position	$3Dmol.Vector3	xyz
screenOffset	$3Dmol.Vector2	x,y
inFront	boolean	
showBackground	boolean	true
fixed	boolean	
useScreen	boolean	
backgroundImage	Object	
		`CanvasImageSource`
alignment	string	topLeft topCenter topRight centerl
frame	number	

[ ]:
```
import py3Dmol
viewer=py3Dmol.view(query="pdb:2EJ0")
viewer.addLabel("Aromatic", {'position': {'x':-6.89, 'y':0.75, 'z':0.35},␣
↪'backgroundColor': '0x800080', 'backgroundOpacity': 0.8})
viewer.addLabel("Label",{'font':'sans-serif','fontSize':18,'fontColor':
↪'white','fontOpacity':1,'borderThickness':1.0,
```

```
                        'borderColor':'red','borderOpacity':0.
 ↪5,'backgroundColor':'black','backgroundOpacity':0.5,
                        'position':{'x':50.0,'y':0.0,'z':0.0},'inFront':
 ↪True,'showBackground':True})
viewer.setStyle({'chain':'A'},{'cross':{'hidden':True}})
viewer.setStyle({'chain':'B'},{'cross':{'hidden':False,
                                    'linewidth':1.0,
                                    'colorscheme':'greenCarbon'}})
viewer.setStyle({'chain':'C'},{'cross':{'hidden':False,
                                    'linewidth':1.0,
                                    'radius':0.5}})
viewer.setStyle({'chain':'D'},{'cross':{'hidden':False,
                                    'linewidth':10.0}})
viewer.setStyle({'chain':'E'},{'cross':{'hidden':False,
                                    'linewidth':1.0,
                                    'color':'black'}})

viewer.render()
```

1.9 addLine

`addLine(spec)`

Name	Type	Description
spec	LineSpec	dashed dashLength gapLength

1.9.1 LineSpec

Name	Type	Description
start	$3Dmol.Vector3	
end	$3Dmol.Vector3	
color	ColorSpec	solid color
alpha	number	
wireframe	boolean	
hidden	boolean	
linewidth	number	
clickable	boolean	callback
callback	function	
frame	number	

[ ]:
```
import py3Dmol
view=py3Dmol.view(query="pdb:2ABJ")
view.setViewStyle({"style":"outline"})
view.setStyle({"chain":"A"},{"sphere":{"hidden":True}})
```

```
view.setStyle({"chain":"D"},{"sphere":{"radius":3.0}})
view.setStyle({"chain":"G"},{"sphere":{"colorscheme":"greenCarbon"}})
view.setStyle({"chain":"J"},{"sphere":{"color":"blue"}})
view.addLine({"dashed":True,"start":{"x":0,"y":0,"z":0},"end":{"x":100,"y":
  ↪100,"z":100}})
view.render()
```

1.10 addMesh

`addMesh(mesh, style)`

Name	Type	Description
mesh	$3Dmol.Mesh	
style	Object	

1.11 addModel

`addModel(data, format, options)` view

Name	Type	Description
data	string	
format	string	`pdb` `sdf` `xyz` `pqr` `mol2`
options	ParserOptionsSpec	

1.11.1 ParserOptionsSpec

GLModel

Name	Type	Description
frames	boolean	true false
vibrate	object	vibration
multimodel	boolean	xyz sdf mol2
onemol	boolean	xyz sdf mol2
keepH	boolean	sdf mol2
parseStyle	object	ChemDoodle cdjson
doAssembly	boolean	mcif
duplicateAssemblyAtoms	boolean	true false
normalizeAssembly	boolean	
dontConnectDuplicatedAtoms	boolean	AssemblyAtoms cif -
noSecondaryStructure	boolean	pdb
noComputeSecondaryStructure	boolean	pdb mmtf cif

Name	Type	Description
hbondCutoff	number	3.2
		pdb mmtf cif
altLoc	string	'*' pdb
assemblyIndex	number	index mmtf
assignBonds	boolean	PDB xyz true

```
viewer.setViewStyle({style:"outline"});
                $.get('data/1fas.pqr', function(data){
                    viewer.addModel(data, "pqr");
                    $.get("data/1fas.cube",function(volumedata){
                        viewer.addSurface($3Dmol.SurfaceType.VDW, {opacity:0.85,voldata: new $3Dr

                    viewer.render();
                    });
                    viewer.zoomTo();
                });
```

1.12 addModels

`addModels(data, format)` viewer

Name	Type	Description
data	string	
format	string	`pdb sdf xyz pqr mol2`

1.13 addModelsAsFrames

`addModelsAsFrames(data, format)` Create and add model to viewer. Given multimodel file and its format, different atomlists are stored in model's frame property and model's atoms are set to the 0th frame.

Name	Type	Description
data	string	
format	string	`pdb sdf xyz pqr mol2`

[ ]:
```
# !!!       open(xxx).read()
import py3Dmol

view=py3Dmol.view()
view.setBackgroundColor("0xffffffff")
data=open("models.xyz").read()
view.addModelsAsFrames(data,format="xyz")
view.animate({"loop":"forward","reps": 1})
view.setStyle({"stick":{"colorscheme":"magentaCarbon"}})
view.zoomTo()
```

```
view.render()
```

1.14 addPropertyLabels

`addPropertyLabels(prop, sel, style)`

Name	Type	Description
prop	string	
sel	Object	
style	Object	example

```
js @jsexample $3Dmol.download("cid:5291",viewer,{},function(){
viewer.setStyle({stick: {radius:.2}}); viewer.addPropertyLabels("index",{not:{elem:'H'}},
{fontColor:'black',font: 'sans-serif', fontSize: 28, showBackground:false,alignment:'center'})
viewer.zoomTo(); viewer.render(); });
```

[ ]:
```
import py3Dmol

view=py3Dmol.view(query="cid:5291")
view.setStyle({"stick":{"radius":0.2}})
view.addPropertyLabels("index",{"not":{"elem":"H"}},{"fontColor":"black","font":
  ↪"sans-seif","fontSize":28,"showBackground":False,"alignment":"center"})
view.zoomTo()
view.show()
```

1.15 addResLabels

`addResLabels(sel, style, byframe)`

[resn][resi]

Name	Type	Description
sel	Object	
style	Object	
byframe	boolean	if true, create labels for every individual frame, not just current

[ ]:
```
import py3Dmol

view=py3Dmol.view(query="mmtf:2ll5")
view.setStyle({"stick":{"radius":0.15},"cartoon":{}})
view.addResLabels({"hetflag":False},{"font":"Arial","fontColor":
  ↪"black","showBackground":False,"screenOffset":{"x":0,"y":0}})
view.zoomTo()
view.render()
```

1.16 addShape

`addShape(shapeSpec)` viewer

Name	Type	Description
shapeSpec	ShapeSpec	

1.16.1 ShapeSpec

Name	Type	Description
color	ColorSpec	solid color
alpha	number	
wireframe	boolean	
hidden	boolean	
linewidth	number	
clickable	boolean	callback
callback	function	
frame	number	

1.17 addSphere

`addSphere(spec)`

Name	Type	Description
spec	SphereShapeSpec	

1.17.1 ShapeSpec

Name	Type	Description
center	$3Dmol.Vector3	
radius	number	
color	ColorSpec	solid color
alpha	number	
wireframe	boolean	
hidden	boolean	
linewidth	number	
clickable	boolean	callback
callback	function	
frame	number	

[ ]:
```
import py3Dmol

view=py3Dmol.view()
view.addSphere({"center":{"x":0,"y":0,"z":0},"radius":10.0,"color":"red"})
```

```
view.zoomTo()
view.render()
```

1.18 addStyle

`addStyle(sel, style)`

Name	Type	Description
sel	AtomSelectionSpec	
style	AtomStyleSpec	

1.18.1 AtomSelectionSpec

Name	Type	Description
...	`AtomSpec`	`AtomSpec`
model	GLModel	-1
bonds	number	{bonds: 0}
predicate	function	{AtomSpec} true
invert	boolean	
byres	boolean	
expand	number	
within	`WithinSelectionSpec`	
and	Array.<`AtomSelectionSpec`>	{AtomSelectionSpec}
or	Array.<`AtomSelectionSpec`>	{AtomSelectionSpec}
not	AtomSelectionSpec	{AtomSelectionSpec}

1.18.2 AtomStyleSpec

Name	Type	Description
line	`LineStyleSpec`	
cross	`CrossStyleSpec`	aka stars
stick	`StickStyleSpec`	
sphere	`SphereStyleSpec`	
cartoon	`CartoonStyleSpec`	cartoon
clicksphere	ClickSphereStyleSpec	

AtomSpec

Name	Type	Description
resn	string	
x	number	x

Name	Type	Description
y	number	y
z	number	z
color	ColorSpec	
surfaceColor	ColorSpec	
elem	string	(e.g. ‘H’, ‘Ca’, etc)
hetflag	boolean	true
chain	string	A “A”
resi	number	
icode	number	
rescode	number	
serial	number	Atom
atom	string	“elem” “CA”
bonds	Array.<number>	id
ss	string	cartoon “h” helix
singleBonds	boolean	true
bondOrder	Array.<number>	“bonds”
properties	Object	
b	number	b
pdbline	string	PDB
		PDB
clickable	boolean	callback
callback	function	
invert	boolean	

WithinSelectionSpec `AtomSelectionSpec`

`AtomSelectionSpec`

Name	Type	Description
distance	number	
invert	boolean	
sel	`AtomSelectionSpec`	

LineStyleSpec

Name	Type	Description
hidden	boolean	
linewidth	number	
colorscheme	`ColorschemeSpec`	
color	`ColorSpec`	Colorscheme
opacity	number	

CrossStyleSpec

Name	Type	Description
hidden	boolean	
linewidth	number	
radius	number	
scale	number	
colorscheme	`ColorschemeSpec`	
color	`ColorSpec`	Colorscheme
opacity	number	

StickStyleSpec

Name	Type	Description
hidden	boolean	
radius	number	
singleBonds	boolean	
colorscheme	`ColorschemeSpec`	
color	`ColorSpec`	Colorscheme
opacity	number	

SphereStyleSpec

Name	Type	Description
hidden	boolean	
radius	number	
scale	number	
colorscheme	`ColorschemeSpec`	
color	`ColorSpec`	Colorscheme
opacity	number	

CartoonStyleSpec “N1” resn “A” “G” “DA” “DG” “N3” ‘C’ ‘U’ ‘DC’ ‘DT’ “P” “O5” 5’

Name	Type	Description
color	`ColorSpec`	“spectrum”
style	string	style of cartoon rendering (trace, oval, rectangle(default), parabola, edged)
ribbon	boolean	
arrows	boolean	beta-sheet
tubes	boolean	alpha
thickness	number	cartoon 0.4
width	number	cartoon strand trace ribbon

Name	Type	Description
opacity	number	0-1

ColorschemeSpec

Name	Type	Description
	string	color>Carbon - Carbon html
ssPymol	string	Pymol
ssJmol	string	Jmol
Jmol	string	Jmol
default	string	
amino	string	
shapely	string	shapely
nucleic	string	
chain	string	
chainHetatm	string	
prop	string	“b” AtomSpec
gradient	Gradient	`$3Dmol.Gradient` rwb roygb s
	min	
	max	
	mid	rwb
map	Object	`AtomSpec {'prop': 'elem', map:$3Dmol.elementColors.greenCarbon}`
		“elem”
colorfunc	function	

[ ]:
```
import py3Dmol

view=py3Dmol.view(query="pdb:5IRE",options={"doAssembly":False})
view.setStyle({"cartoon":{}})
view.addStyle({"chain":"A"},{"stick":{"radius":0.5,"colorscheme":
 ↪"magentaCarbon"}})
view.zoomTo()
view.render()
```

[ ]:
```
import py3Dmol

view=py3Dmol.view(query="pdb:2EJ0")
view.setStyle({"chain":"A","within":{"distance":10,"sel":{"chain":
 ↪"B"}}},{"sphere":{}})
view.zoomTo()
view.render()
```

[ ]:
```
import py3Dmol

view=py3Dmol.view(query="pdb:2JE0")

view.setStyle({"chain":"B"},{"cartoon":{"color":"spectrum"}})
view.setStyle({"chain":"B","invert":True},{"cartoon":{}})
view.setStyle({"bonds":0},{"sphere":{"radius":0.5}})
view.setStyle({"resn":"PMP","byres":True,"expand":5},{"stick":{"colorscheme":
 ↪"greenCarbon"}})
view.setStyle({"resi":["91-95","42-50"]},{"cartoon":{"color":
 ↪"green","thickness":1.0}})
view.zoomTo()
view.render()
```

[ ]:
```
#   func
import py3Dmol

view=py3Dmol.view(query="pdb:4UAA")

colorAsSnake ='''function(atom) {
                   return atom.resi % 2 ? 'white': 'green'
                 };
'''

view.setBackgroundColor("0xffffffff")
# view.setStyle( {"chain":'A'}, { "cartoon": {"colorfunc": colorAsSnake }})
view.setStyle( {"chain":'B'}, { "stick": {"colorscheme": 'yellowCarbon'}})
view.render()
```

[ ]:
```
import py3Dmol

view=py3Dmol.view(query="pdb:4ZD3")
view.setBackgroundColor("0xffffffff")
view.setViewStyle({"style":"outline"})
view.setStyle({},{"cartoon":{}})
view.render()
```

1.19 addSurface

`addSurface(type, style, atomsel, allsel, focus, surfacecallback)`

Name	Type	Description
type	$3Dmol.SurfaceType | string	py3Dmol.VDW MS SAS SES
style	`SurfaceStyleSpec`	
atomsel	`AtomSelectionSpec`	
allsel	`AtomSelectionSpec`	“atomsel”

Name	Type	Description
focus	`AtomSelectionSpec`	
surfacecallback	function	

1.19.1 SurfaceStyleSpec

Name	Type	Description
opcaity	number	0 1
colorscheme	`ColorschemeSpec`	
color	`ColorSpec`	`colorscheme`
voldata	`$3Dmol.VolumeData`	VolumeData volformat
volscheme	`$3Dmol.Gradient`	
volformat	string	$3Dmol.VolumeData voldata
map	Object	(prop) ( ) `$3Dmol.Gradient.RWB` colorcheme

```
// js
var setStyles = function(volumedata){
                    var data = new $3Dmol.VolumeData(volumedata, "cube");
                    viewer.addSurface("VDW", {opacity:0.85, voldata: data, volscheme: new $3Dmo
                    viewer.mapAtomProperties($3Dmol.applyPartialCharges);
                    viewer.addSurface($3Dmol.SurfaceType.SAS, {map:{prop:'partialCharge',scheme
                    viewer.addSurface($3Dmol.SurfaceType.VDW, {opacity:0.85,voldata: data, colo
                    viewer.addSurface($3Dmol.SurfaceType.SAS, {opacity:0.85,voldata: data, colo

                viewer.render();
                };
                $3Dmol.download("pdb:4DLN",viewer,{},function(){

                    $.get("data/1fas.cube",setStyles);
                  });
```

1.20 addUnitCell

`addUnitCell(model, spec)`

Name	Type	Description
model	GLModel	pdb
spec	`UnitCellStyleSpec`	

1.20.1 UnitCellStyleSpec

Name	Type	Description
box	`LineStyleSpec`	
astyle	`ArrowSpec`	a
bstyle	`ArrowSpec`	b
cstyle	`ArrowSpec`	c
alabel	string	a
alabelstyle	`LabelSpec`	a
blabel	string	b
blabelstyle	`LabelSpec`	b
clabel	string	c
clabelstyle	`LabelSpec`	c

LineStyleSpec

Name	Type	Description
hidden	boolean	
linewidth	number	
colorscheme	`ColorschemeSpec`	
color	`ColorSpec`	Colorscheme
opacity	number	

ArrowSpec

Name	Type	Description
start	$3Dmol.Vector3	
end	$3Dmol.Vector3	
radius	number	
color	ColorSpec	solid color
hidden	boolean	
rtadiusRatio	number	1.618034
mid	number	0.618034
midpos	number	,
alpha	number	
wireframe	boolean	
linewidth	number	
clickable	boolean	callback
callback	function	
frame	number	

LabelSpec

Name	Type	Description
font	string	sans-serif

Name	Type	Description
fontSize	number	18
fontColor	ColorSpec	
fontOpacity	number	1
borderThickness	number	0
borderColor	ColorSpec	backgroundColor
borderOpacity	string	
backgroundColor	ColorSpec	
backgroundOpacity	string	1
position	$3Dmol.Vector3	xyz
screenOffset	$3Dmol.Vector2	x,y
inFront	boolean	
showBackground	boolean	true
fixed	boolean	
useScreen	boolean	
backgroundImage	Object	
		`CanvasImageSource`
alignment	string	topLeft topCenter topRight centerl
frame	number	

[ ]:
```
import py3Dmol

view=py3Dmol.view()
view.setBackgroundColor("0xffffffff")
data=open("1jpy.cif").read()
view.addModel(data,"cif")
view.setStyle({'cartoon':{}})
view.addUnitCell(data,{"box":{"color":"purple"},"alabel":"x","blabel":
 ↪"y","clabel":"z","alabelstyle": {"fontColor": 'black',"backgroundColor":
 ↪'white',"inFront":True,"fontSize":40},"astyle":{"color":'darkred', "radius":
 ↪5,"midpos": -10}})
view.zoomTo()
view.render()
```

1.21 addVolumetricData

`addVolumetricData(data, format, or)` guassian cube

Name	Type	Description
data	string	
format	string	
or	`IsoSurfaceSpec`	{VolumetricRenderSpec} spec - Shape style specification

1.21.1 IsoSurfaceSpec

Name	Type	Description
isoval	number	
color	ColorSpec	solid color
opacity	number	, between 0 and 1
wireframe	number	draw as wireframe, not surface
linewidth	number	
smoothness	number	amount to smooth surface (default 1)
coords	list	`viewer.selectedAtoms()` `AtomSelecti`
seldist	number	[= 2.0]
voldata	$3Dmol.VolumeData	`VolumeData` `volformat`
volscheme	$3Dmol.Gradient	
volformat	string	`$3Dmol.VolumeData` `voldata`
clickable	boolean	callback
callback	function	

[ ]:
```
# fail
import py3Dmol

view=py3Dmol.view()
data=open("test.cube").read()
view.setBackgroundColor("0xffffffff")
view.addVolumetricData(data,"cube",{"isoval":-0.01,"color":"red","opacity":0.
↪95})
view.setStyle({"cartoon":{},"stick":{}})
view.zoomTo()
view.render()
```

1.22 addVolumetricRender

`addVolumetricRender(data, spec)` volumetricData

Name	Type	Description
data	$3Dmol.VolumeData	
spec	VolumetricRenderSpec	

1.23 animate

`animate(options)` viewer

Name	Type	Description
options	Object	can specify interval (speed of animation), loop (direction of looping, 'backward', 'forward' or 'backAndForth'), step interval between frames ('step'), and reps (numer of repetitions, 0 indicates infinite loop)

1.24 apngURI

`apngURI()`

Return a promise that resolves to an animated PNG image URI of viewer contents (base64 encoded) for nframes of viewer changes.

1.25 center

`center(sel, animationDuration, fixedPath)` viewer ZoomTo

Name	Type	Argument	Description
sel	Object	<optional>	
animationDuration	number	<optional>	
fixedpath	Boolean	<optional>	
			true *

```
[ ]: import py3Dmol

view=py3Dmol.view(query="pdb:4csv")
view.setStyle({"cartoon":{},"stick":{}})
view.center()
view.render()
```

1.26 clear

`clear()`

1.27 createModelFrom

`createModelFrom(sel, extract)` sel extract

Name	Type	Argument	Description
sel	Object		
extract	boolean	<optional>	if true

1.28 enableContextMenu

`enableContextMenu(sel, contextMenuEnabled)` enable context menu and callback of selected atoms

Name	Type	Description
sel	`AtomSelectionSpec`	atom selection to apply hoverable settings to
contextMenuEnabled	boolean	

1.29 enableFog

`enableFog(fog)`

Name	Type	Description
fog	boolean	

1.30 exportJSON

`exportJSON(includeStyles, modelID)` ChemDoodle JSON

Name	Type	Description
includeStyles	boolean	styles
modelID	number	

1.31 exportVRML

`exportVRML` VRML VRML

1.32 fitSlab

`fitSlab(sel)` Adjust slab to fully enclose selection (default everything).

Name	Type	Argument	Description
sel	Object		

1.33 getFrame

`getFrame()`

1.34 getInternalState

`getInternalState()` setInternalState view

1.35 getModel

`getModel(id)`

Name	Type	Argument	Default	Description
id	number	<optional>	last model id	id model

Default viewer null

```
// Retrieve reference to first GLModel added var m =
   $3Dmol.download("pdb:1UBQ",viewer,{},function(m1){
                   $3Dmol.download("pdb:1UBI", viewer,{}, function(m2) {
                     viewer.zoomTo();
                     m1.setStyle({cartoon: {color:'green'}});
                     //could use m2 here as well
                     viewer.getModel().setStyle({cartoon: {color:'blue'}});
                     viewer.render();
                   })
                 });
```

1.36 getNumFrames

`getNumFrames()` viewer

1.37 getPerceivedDistance

`getPerceivedDistance()` model camera z

1.38 getSlab

`getSlab()` Get slab of view (contents outside of slab are clipped).

Name	Type	Description
near	number	near clipping plane distance
far	number	far clipping plane distance

1.39 getSurface

`getSurface(surf)` surface

Name	Type	Description
surf	number	surface id

1.40 getView()

`getview()` view Translation, zoom, and rotation quaternion.

1.41 hasVolumetricRender

`hasVolumetricRender()` true WebGL 2.0

1.42 isAnimated

`isAnimated()` true false

1.43 linkViewer

`linkViewer(otherview)` viewer viewer

Name	Type	Description
otherview	`$3Dmol.GLViewer`	

1.44 mapAtomProperties

`mapAtomProperties(props,, sel)`

Name	Type	Description
props,	Object	either array of atom selectors with associated props, or function that takes atom and sets its properties
sel	`AtomSelectionSpec`	-

[ ]:
```
import py3Dmol

view=py3Dmol.view()
data=open("b.sdf").read()
view.addModel(data,"sdf")
props=[]
for i in range(7):
    props.append({"index":i,"props":{"x":i}})

view.mapAtomProperties(props)
view.setStyle({"sphere":{"colorscheme":{"gradient":'roygb',"prop":'x',"min":
 ↪0,"max":8}}})
view.zoomTo()
view.render()
```

1.45 modelToScreen

`modelToScreen()`

Type	Description
object|list	an object or list of objects with x,y,z attributes (e.g. an atom)

1.46 pauseAnimate

`pauseAnimate()`

1.47 pdbData

`pdbData(sel)` pdb pdb

Name	Type	Argument	Description
sel	Object	<optional>	Selection specification specifying model and atom properties to select. Default: all atoms in viewer

1.48 pngURI

`pngURI()` Return image URI of viewer contents (base64 encoded).

1.49 removeAllLabels

viewer

[ ]:
```
import py3Dmol

view=py3Dmol.view(query="pdb:1ubq")
view.addResLabels()
view.setStyle({"stick":{}})
view.render()
#
view.removeAllLabels()
view.render()
```

1.50 removeAllModels

`removeAllModels()` model

1.51 removeAllShapes

`removeAllShapes()` shape

1.52 removeAllSurfaces

`removeAllSurfaces()`

1.53 removeLabel

`removeLabel(label)` viewer

Name	Type	Description
label	`$3Dmol.Label`	`$3Dmol.Label`

[ ]:
```
import py3Dmol

view=py3Dmol.view(query="pdb:2EJ0")

toremove=["Aromatic",{"position":{"x":-6.89,"y":0.75,"z":0.
 ↪35},"backgroundColor":"0x800080","backgroundOpacity":0.8}]
view.addLabel("Label",{"font":"sans-seif","fontSize":18,"fontColor":
 ↪"white","fontOpacity":1,"borderThickness":1.0,
                       "borderColor":"red","borderOpacity":0.
 ↪5,"backgroundColor":"black","backgroundOpacity":0.5,
                       "position":{"x":50.0,"y":0.0,"z":0.0},"inFront":
 ↪True,"showBackground":True})

#
view.removeLabel(toremove)
view.render()
```

1.54 removeModel

`removeModel(model)` model

Name	Type	Description
model	`$3Dmol.GLModel`	

1.55 removeShape

`removeShape(shape)` shape

Name	Type	Description
shape	`$3Dmol.GLShape`	Reference to shape object to remove

1.56 removeSurface

`removeSurface(surf)` ID surface

Name	Type	Description
surf	number	surface id

1.57 removeUnitCell

`removeUnitCell(model)`

Name	Type	Description
model	GLModel	Model with unit cell information (e.g., pdb derived). If omitted uses most recently added model.

[ ]:
```
import py3Dmol

view=py3Dmol.view()
data=open("icsd_200866.cif").read()
view.addModel(data)
view.setStyle({"sphere":{}})
view.addUnitCell()
view.zoomTo()
#
view.removeUnitCell()
view.render()
```

1.58 render

`render()` viewer

1.59 replicateUnitCell

`replicateUnitCell(A, B, C, model)`

Name	Type	Description
A	integer	X
B	integer	Y , X
C	integer	Z , X
model	GLModel	pdb

[ ]:
```
import py3Dmol

view=py3Dmol.view()
data=open("icsd_200866.cif").read()
view.addModel(data)
view.setStyle({"sphere":{"scale":0.25}})
view.addUnitCell()
view.zoomTo()
view.replicateUnitCell(3,2,1,data)
```

```
view.render()
```

1.60 resize

`resize()` Resize viewer according to containing HTML element's dimensions

1.61 resumeAnimate

`resumeAnimate()` Resume animation of all models in viewer

1.62 rotate

`rotate(angle, axis, animationDuration, fixedPath)`

Name	Type	Argument	Description
angle	number	<optional>	degrees
axis	string	<optional>	"x" "y" "z" "vx" "vy" "vz" "y" v
animationDuration	number	<optional>	
fixedpath	Boolean	<optional>	true *

[ ]:
```
import py3Dmol

view=py3Dmol.view(query="cid:4000")
view.setStyle({"stick":{}})
view.zoomTo()
view.rotate(90,"y",1000)
view.render()
```

1.63 screenOffsetToModel

`screenOffsetToModel()` For a given screen (x,y) displacement return model displacement

1.64 screenToModelDistance

`screenToModelDistance()`

Distance from screen coordinate to model coordinate assuming screen point is projected to the same depth as model coordinate

1.65 selectedAtoms

`selectedAtoms(sel)` sel

Name	Type	Description
sel	AtomSelectionSpec	

1.66 setAutoEyeSeparation

`setAutoEyeSeparation()` Used for setting an approx value of eyeSeparation. Created for calling by StereoViewer object

1.67 setBackgroundColor

`setBackgroundColor(hex, a)`

Name	Type	Description
hex	number	
a	number	default 1.0

1.68 setCameraParameters

`setCameraParameters()` (distance to the origin and field of view)

|Type|Description| |parameters| fov z false |

[ ]:
```
import py3Dmol

view=py3Dmol.view()
data=open("set1_122_complex.mol2").read()
view.addModel(data)
view.setStyle({"stick":{}})
view.zoomTo()
#
view.setCameraParameters({"fov":10,"z":300})
view.center()
```

1.69 setClickable

`setClickable(sel, clickable, callback)`

Name	Type	Description
sel	`AtomSelectionSpec`	click
clickable	boolean	callback
callback	function	

[ ]:
```
import py3Dmol

fnc='''
function(atom,viewer,event,container) {
                        viewer.addLabel(atom.resn+":"+atom.atom,{position: atom,␣
 ↪backgroundColor: 'darkgreen', backgroundOpacity: 0.8});}
'''

view=py3Dmol.view(query="cid:307900")
```

```
view.setStyle({},{"sphere":{}})
view.setClickable({},True,fnc)
view.render()
```

1.70 setColorByElement

`setColorByElement(sel, colors)`

Name	Type	Description
sel	`AtomSelectionSpec`	
colors	type	

1.71 setColorByProperty

`setColorByProperty(sel, prop, scheme)`

Name	Type	Description
sel	`AtomSelectionSpec`	
prop	type	
scheme	type	

1.72 setContainer

`setContainer(element)`
Change the viewer's container element
Also useful if the original container element was removed from the DOM.

Name	Type	Description
element	Object|string	Either HTML element or string identifier. Defaults to the element used to initialize the viewer.

1.73 setFrame

`setFrame(framenum)`
Sets the atomlists of all models in the viewer to specified frame.
Shapes and labels can also be displayed by frame.
Sets to last frame if framenum out of range

Name	Type	Description
framenum	number	fame index to use, starts at zero

1.74 setHeight

`setHeight(h)`

Name	Type	Description
h	number	height in pixels

1.75 setHoverable

`setHoverable(sel, hoverable, hover_callback, unhover_callback)` Set hoverable and callback of selected atoms

Name	Type	Description
sel	`AtomSelectionSpec`	
hoverable	boolean	
hover_callback	function	
unhover_callback	function	

```
// js
$3Dmol.download("pdb:1ubq",viewer,{},function(){

                viewer.setHoverable({},true,function(atom,viewer,event,container) {
                    if(!atom.label) {
                     atom.label = viewer.addLabel(atom.resn+":"+atom.atom,{position: atom, backg
                    }
                },
                function(atom) {
                    if(atom.label) {
                     viewer.removeLabel(atom.label);
                     delete atom.label;
                    }
                 }
                );
                viewer.setStyle({},{stick:{}});
                viewer.render();
        });
```

[ ]:
```
#  js
v = py3Dmol.view(query="pdb:1ubq",style={'cartoon':{},'stick':{}})
v.setHoverable({},True,'''function(atom,viewer,event,container) {
                    if(!atom.label) {
                     atom.label = viewer.addLabel(atom.resn+":"+atom.
 ↪atom,{position: atom, backgroundColor: 'mintcream', fontColor:'black'});
                    }}''',
                '''function(atom,viewer) {
                    if(atom.label) {
                     viewer.removeLabel(atom.label);
```

```
                    delete atom.label;
                }
            }''')
```

1.76 setHoverDuration

`setHoverDuration(hoverDuration)`

Name	Type	Argument	Description
hoverDuration	number	<optional>	

1.77 setInternalState

`setInternalState()` `getInternalState` viewer

1.78 setLabelStyle

`setLabelStyle(label, stylespec)`

Name	Type	Description
label	$3Dmol.Label	$3Dmol.Label
stylespec	Object	

1.79 setLabelText

`setLabelText(label, text)`

Name	Type	Description
label	$3Dmol.Label	$3Dmol.Label
text	string	

1.80 setPerceivedDistance

`setPerceivedDistance()`
Set the distance between the model and the camera
Essentially zooming. Useful while stereo rendering.

1.81 setProjection()

`setProjection()`

orthographic true viewer

[ ]:
```
# Fail
import py3Dmol
```

```
view=py3Dmol.view()
data=open("1fas.pqr").read()

view.addModel(data,"pqr")
cube=open("1fas.cube").read()
view.addSurface(py3Dmol.VDW,{"opacity":0.85,"voldata":cube,"volscheme":{}},{})
view.zoomTo()
view.setProjection("orthographic")
view.render()
```

1.82 setSlab

`setSlab(near, far)` render

Name	Type	Description
near	number	near clipping plane distance
far	number	far clipping plane distance

1.83 setStateChangeCallback

`setStateChangeCallback()`Set a callback to call when the view has potentially changed.

1.84 setStyle

`setStyle(sel, style)`

Name	Type	Description
sel	`AtomSelectionSpec`	
style	`AtomStyleSpec`	

[ ]:
```
import py3Dmol

view=py3Dmol.view("pdb:5IRE",style={"doAssembly":False})
view.setStyle({"chain":"A"},{"cartoon":{"color":"specturm"}})
view.setStyle({"chain":"C"},{"cartoon":{"style":'trace',"color":'blue'}})
view.setStyle({"chain":"E"},{"cartoon":{"tubes":True,"arrows":True,"color":
 ↪'green',"opacity":0.75}})
view.setStyle({"chain":"B"},{"cartoon":{"color":'red',"opacity":0.5}})
view.setStyle({"chain":"D"},{"cartoon":{"style":'trace',"color":
 ↪'grey',"opacity":0.75}})
view.setStyle({"chain":"F"},{"cartoon":{"arrows":True,"color":'white'}})
view.zoomTo()
view.render()
```

1.85 setSurfaceMaterialStyle

`setSurfaceMaterialStyle(surf, style)`

Name	Type	Description
surf	number	Surface ID to apply changes to
style	`SurfaceStyleSpec`	new material style specification

1.85.1 SurfaceStyleSpec

Name	Type	Description
opcaity	number	0 1
colorscheme	`ColorschemeSpec`	
color	`ColorSpec`	`colorscheme`
voldata	`$3Dmol.VolumeData`	VolumeData volformat
volscheme	`$3Dmol.Gradient`	
volformat	string	$3Dmol.VolumeData voldata
map	Object	(prop) ( ) `$3Dmol.Gradient.RWB` colorcheme

```
[ ]: import py3Dmol

view=py3Dmol.view()
data=open("9002806.cif").read()
view.addModel(data,"cif")
view.setStyle({"stick":{}})
view.addSurface("SAS")
view.setSurfaceMaterialStyle(1,{"color":"blue","opacity":0.5})
view.render()
```

1.86 setView

`setView(arg)` Sets the view to the specified translation, zoom, and rotation.

Name	Type	Description
arg	Array.<number>	Array formatted identically to the return value of getView

1.87 setViewChangeCallback

`setViewChangeCallback()` Set a callback to call when the view has potentially changed.

1.88 setViewStyle

`setViewStyle()` Set global view styles.

1.89 setWidth

`setWidth(w)`

Name	Type	Description
w	number	width in pixels

1.90 setZoomLimits

`setZoomLimits()`

Name	Description
lower	limit on zoom in (positive number). Default 0.
upper	limit on zoom out (positive number). Default infinite.

[ ]:
```
import py3Dmol

view=py3Dmol.view()

data=open("b.sdf").read()
view.addModel(data,"sdf")
view.setStyle({"stick":{"colorscheme":"Jmol"}})
view.setZoomLimits(50,100)
view.zoomTo()
# view.zoom(10)
view.render()
```

1.91 spin

`spin(axis, speed)`

Call `view.spin(false)` to stop spinning.

Name	Type	Argument	Description
axis	string	<optional>	Axis (“x”, “y”, “z”, “vx”, “vy”, or “vz”) to rotate around. Default “y”. View relative (rather than model relative) axes are prefixed with v.

Name	Type	Argument	Description
speed	number	<optional>	Speed multiplier for spinning the viewer. 1 is default and a negative value reverses the direction of the spin.

1.92 stopAnimate

`stopAnimate()`

1.93 targetedObjects

`targetedObjects()` Return a list of objects that intersect that at the specified viewer position.

Type	Description
x	x position in screen coordinates
y	y position in screen coordinates
objects	list of objects or selection object specifying what object to check for targeting

1.94 translate

`translate(x, y, animationDuration, fixedPath)` x,y

Name	Type	Argument	Description
x	number		
y	number		
animationDuration	number	<optional>	
fixedPath	Boolean	<optional>	*

[ ]:
```
import py3Dmol

view=py3Dmol.view("pdb:4csv")
view.setStyle({"cartoon":{},"stick":{}})
view.zoomTo()
view.translate(200,50)
view.rotate(90,"z")
view.render()
```

1.95 translateScene

`translateScene(x, y, animationDuration, fixedPath)` x,y

Name	Type	Argument	Description
x	number		x
y	number		y
animationDuration	number	<optional>	
fixedPath	Boolean	<optional>	*

[ ]:
```
import py3Dmol

view=py3Dmol.view("pdb:4csv")
view.setStyle({"cartoon":{},"stick":{}})
view.zoomTo()
view.translateScene(200,50)
view.rotate(90,"z")
view.render()
```

1.96 vibrate

`vibrate(numFrames, amplitude, bothWays, arrowSpec)` dx dy dz xyz

.

Name	Type	Description
numFrames	number	10
amplitude	number	1
bothWays	number	if true, extend both in positive and negative directions by numFrames
arrowSpec	`ArrowSpec`	specification for drawing animated arrows. If color isn't specified, atom color (sphere, stick, line preference) is used.

1.97 zoom

`zoom(factor, animationDuration, fixedPath)`

Name	Type	Argument	Description	
factor	number	<optional>	1	2
animationDuration	number	<optional>		
fixedPath	Boolean	<optional>		*

[ ]:
```
import py3Dmol

view=py3Dmol.view("pdb:4csv")
view.setStyle({"cartoon":{},"stick":{}})
view.zoomTo()
```

```
view.zoom(2,1000)
view.render()
```

1.98 zoomTo

`zoomTo(sel, animationDuration, fixedPath)`
Zoom to center of atom selection. The slab will be set appropriately for the selection, unless an empty selection is provided, in which case there will be no slab.

Name	Type	Argument	Description
sel	Object	<optional>	Selection specification specifying model and atom
animationDuration	number	<optional>	
fixedPath	Boolean	<optional>	*