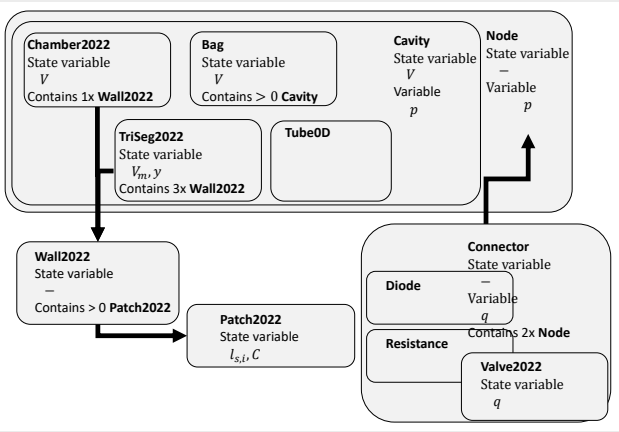
pyCircAdapt Cheat Sheet

CircAdapt

```
>>> import circadapt
```

Components



Solvers

Solvers in the package.

Loading models

```
>>> import circadapt.model
```

and create model VanOsta2023

```
>>> model = circadapt.model.VanOsta2023()
```

Models are loaded without signals, so you must run at least 1 beat.

Creating models

Always set the solver while creating a custom model.

```
>>> model = circadapt.CircAdapt(solver=solver)
```

Add components to the model (see Components).

```
>>> model.add_component(type, name, parent="")
```

Run a beat

```
>>> model.run()
```

By default, only 1 beat is stored. Store more beats with

```
>>> model['Solver']["store_beats"] = 2
```

Run 10 beats with

```
>>> model.run(10)
```

Pressure-flow-control module determines hemodynamic stability.

```
>>> model.run(stable=True)
```

Handling errors

After experiencing numerical instabilities, the ModelCrashed error is raised. To continue, catch the error.

```
>>> from circadapt.error import ModelCrashed
>>> try:
>>>  model.run()
>>> except ModelCrashed:
>>>  # do something
```

The model raises a ModelNotStable error when no hemodynamic stability is reached after run stable.

```
>>> from circadapt.error import ModelCrashed
>>> try:
>>>  model.run(stable=True)
>>> except ModelNotStable:
>>>  # do something
```

Get and set data

Parameters act like one dimensional numpy arrays. Signals act like two dimensoinal numpy arrays with time and objects on first and second dimension. Two examples:

```
>>> model["Patch2022"]["Sf_act"][["pLv1", "pSv1", "pRv1"]]
>>> model["Patch2022"]["l_s"][50:, ["pLv1", "pSv1", "pRv1"]]
```