EventFunc/Def Discussion

represents a thing with memory that can turn on and off.
Memory can be manipulated internally (in startFunc and stopFunc) and externally through .put messages

manages any number of synths and tasks that can be updated automatically from the environment.

EventDef(name, startFunc, stopFunc);

name - a symbol, the EventFunc is associate with this key in the EventDef container
startFunc - a Function that will execute when .start method is performed. The first argument into the function is self
stopFunc - a Function that will execute when .start method is performed. The first argument into the function is self

An EventDef has a named pair of functions and responds to start and stop messages.

```
EventDef(\test, { |ed| [ed.name, \start].postln },{ |ed| [ed.name, \stop].postln });

EventDef(\test).start  //first function executes
EventDef(\test).stop   //second function executes
```

An EventDef has its own local Environment that the functions are evaluated within

```
EventDef(\test2, { |ed| [ed.name, ~x, ~y].postln },{ |ed| [ed.name, \stop].postln });

EventDef(\test2).put(\x, 100, \y, 100.rand)

EventDef(\test2).start
EventDef(\test2).stop
```

An EventDef has a state and does not respond to repeated calls to start or stop methods.

```
EventDef(\test2).state //observe the state
EventDef(\test2).start
EventDef(\test2).state //state has changed
EventDef(\test2).start //nothing happens
EventDef(\test2).state //state remains the same
EventDef(\test2).stop
EventDef(\test2).state //state has changed
EventDef(\test2).stop  //nothing happens
```

```
EventDef(\test2).state //state remains the same
```