# Tidal

## 1 Introduction

Welcome to this workshop on tidalcycles, known as *tidal* for short.

### 1.1 What is a cycle?

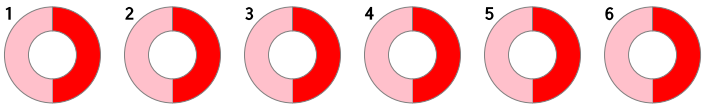
- Cyclic notion of time from Indian Classical music
- The end is also the beginning (the *sam*)
- Time in Tidal is based on cycles, rather than beats
- Cycles are ticking over all the time
- Cycles have fixed duration (which you can change with the *cps* command)

## 2 Basics of polyrhythmic sequencing with Tidal

Before we get hands on, lets look at some visual renderings of tidal patterns.

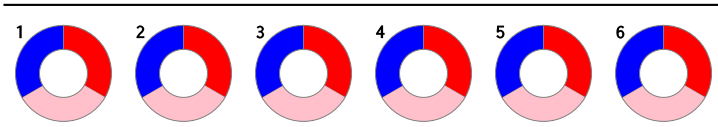
Sequences in tidal are generally denoted with double quotes:

```
"red pink"
```




You can 'read' the above diagram clockwise, from the top.

```
"red pink blue"
```




## 3 Introduction to patterns - repetition, symmetry, interference and glitch

## 4 Haskell syntax

## 5 Ensemble play

## 6 More complex patternings

## 7 Strategies for live coding performance

## 8 Composing with tidal

## 9 Superdirt - synths, customisation, multichannel, midi control

## 10 Visualisation

## 11 Community

- http://tidalcycles.org/
- http://talk.lurk.org (e.g. #tidal, #livecode, #algorave channels)
- http://algorave.com/
- http://github.com/tidalcycles/