```
tt()
```

SCAL
ING

D3

Schedule

1. Scales in D3

Three types of scales

Scales help you **calculate** how big elements of your graph are going to be and where they should be positioned based on a **value** in your dataset. You can also use them for colour coding elements.

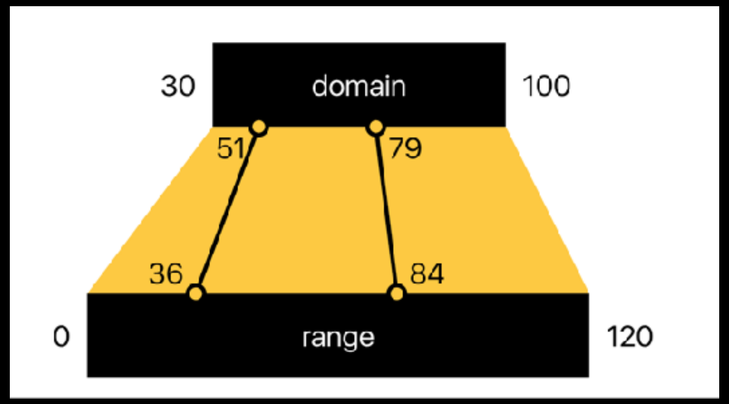
Two types of values:

Continuous values are kinda like measuring something with a ruler. The value could be any number between 0 and 100, for example 43,50

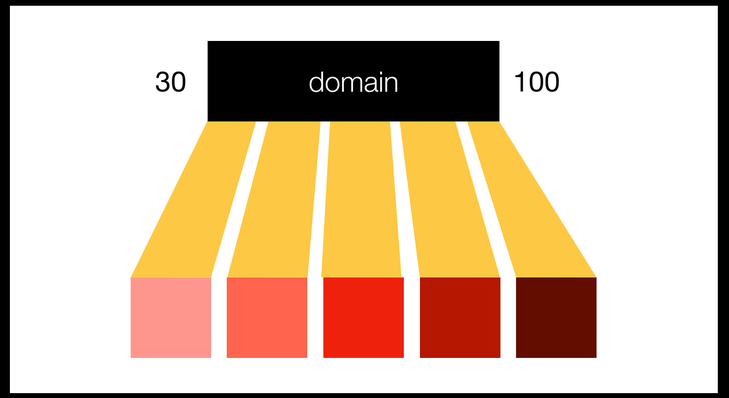
Discrete values are kinda like sorting something in different trays or folders. A value is either in one tray or in the other. For example, your values may be days of the week. The day is either Monday or Tuesday, but not Monday and a half.

Three types of scales

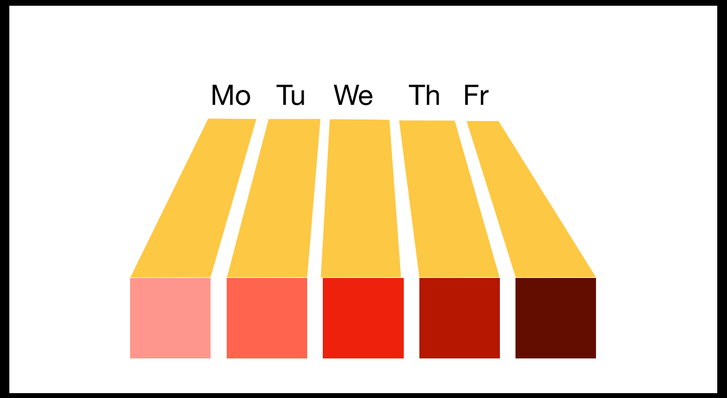
Using these two types of values, we can create three kinds of scales:



Continuous -> Continuous



Continuous -> Discrete



Discrete -> Discrete

The *domain* is the value that you put into the scale function. It usually comes from your dataset. The *range* is the range of values you want to translate to and are the output of your map function. The range can be a number but also a colour.

Scale examples

color scale

```
var color = d3.scaleLinear()
  .domain([10, 100]) //data
  .range(['brown', 'steelblue']) //colorrange

color(20) // Output: '#9a3439'
color(50) // Output: ’#7b5167'
```

Scale examples

```
var scale = d3.scaleLinear()
  .domain([10, 130])
  .range([0, 960])

scale(-10) // Output: -160, outside range

scale.clamp(true)
scale(-10) // Output: 0, clamped to range
```

clamping scales

By default, D3 scales will try to use the still return a scaled value if the data you give it is outside the domain. This could be a weird outlier in your dataset for example or just a bug in your API.

If you want your scale to always stay within the range, you can add `.clamp(true)` to your scale function.

Scale examples

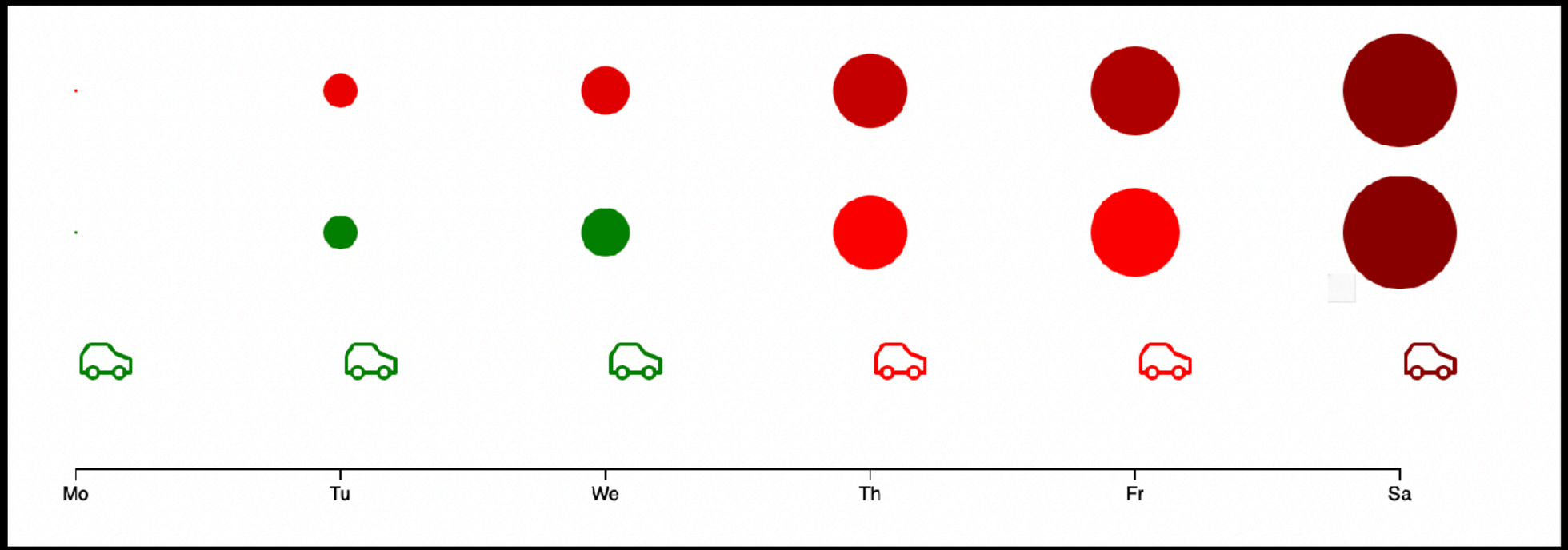
time scale

```
var time = d3.scaleTime()
  .domain([new Date(2000, 0, 1), new Date(2000, 0, 2)])
  .range([0, 960])

time(new Date(2000, 0, 1,  5)) // Output: 200
time(new Date(2000, 0, 1, 16)) // Output: 640
```

All things combined

https://codepen.io/vijnv/pen/eYEKXve?editors=1010



```
Uncaught SyntaxError
Unexpected end of input
```