Week 11 - 2nd Order ODEs

Review

> Manipulate[expression[a], {a,min, max}]
>DSolve[differntial equation, y[x], x]
> Curly brackets used to add in extra information
>Remove[y]

Introduction

Same as DSolve in the case that we don't specify initial conditions however we now have two constants c[1] and c[2].

```
DSolve[y''[x] == -y[x], y[x], x]
```

```
{{y[x] → C[1] Cos[x] + C[2] Sin[x]}}
```

Therefore we now need to have two initial conditions to fully define our function

```
DSolve[{y''[x] == -y[x], y'[0] == 0, y[0] == 1}, y[x], x]
```

```
{{y[x] → Cos[x]}}
```

We have so far used y(x), we can also use x(t) or f(r) or any other combination as long as we are careful not to use variables that are defined as something different elsewhere.

```
DSolve[differential equation, function, variable]
```

A useful application of studying differential equations are in the context of oscilations. We can study solutions of x''(t) = - k*x(t)/m

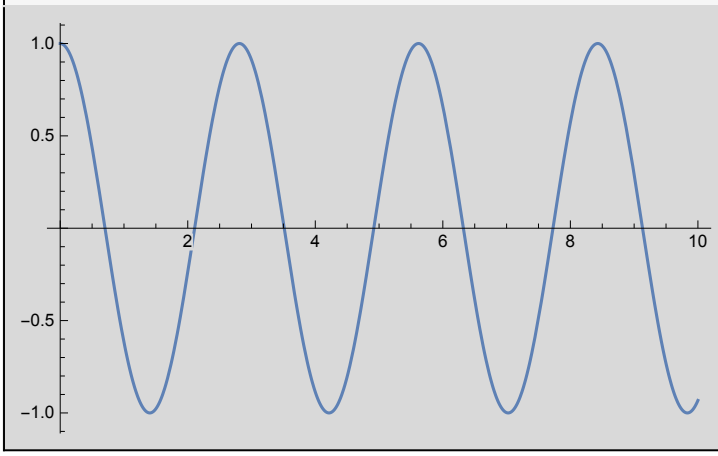
```
m = 1(*set the mass = 1*)
```

```
DSolve[{x''[t] == -k * x[t] / m, x'[0] == 0, x[0] == 1}, x[t], t]
```

```
{{x[t] → Cos[√k t]}}
```

```
function1[k_, t_] := Cos[√k t]
```

```
Plot[function1[5, t], {t, 0, 10}]
```




```
function2[k_] := Plot[function1[k, t], {t, 0, 10}]
```

```
Manipulate[function2[k], {k, 1, 5}]
```

Summary

> As in maths we can use any symbol to define our functions. However we need to be extra careful not to use symbols we have defined differently elsewhere.
> DSolve is the same for 1st and 2nd order differential equations but in the case of 2nd order differential equations we need to define 2 initial conditions.
> .Functions are a good way of splitting up code into understandable pieces that can be reused with different input values.