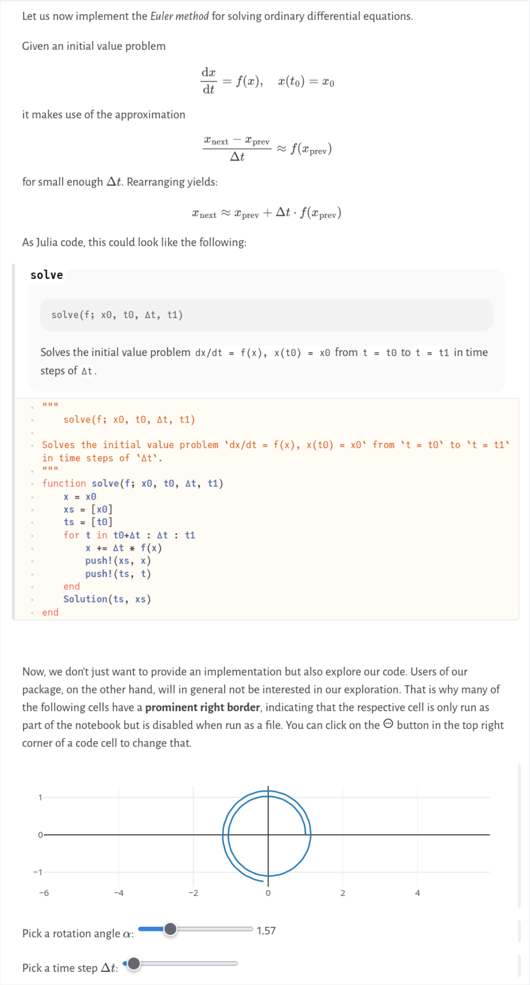
Literate package development with Julia and Pluto.jl



Let us now implement the *Euler method* for solving ordinary differential equations.

Given an initial value problem

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x), \quad x(t_0) = x_0$$

it makes use of the approximation

$$\frac{x_{\mathrm{next}} - x_{\mathrm{prev}}}{\Delta t} \approx f(x_{\mathrm{prev}})$$

for small enough Δt. Rearranging yields:

$$x_{\mathrm{next}} \approx x_{\mathrm{prev}} + \Delta t \cdot f(x_{\mathrm{prev}})$$

As Julia code, this could look like the following:

solve

```
solve(f; x0, t0, Δt, t1)
```

Solves the initial value problem `dx/dt = f(x), x(t0) = x0` from `t = t0` to `t = t1` in time steps of `Δt`.

```julia
"""
    solve(f; x0, t0, Δt, t1)

Solves the initial value problem `dx/dt = f(x), x(t0) = x0` from `t = t0` to `t = t1`
in time steps of `Δt`.
"""
function solve(f; x0, t0, Δt, t1)
    x = x0
    xs = [x0]
    ts = [t0]
    for t in t0+Δt : Δt : t1
        x += Δt * f(x)
        push!(xs, x)
        push!(ts, t)
    end
    Solution(ts, xs)
end
```

Now, we don't just want to provide an implementation but also explore our code. Users of our package, on the other hand, will in general not be interested in our exploration. That is why many of the following cells have a **prominent right border**, indicating that the respective cell is only run as part of the notebook but is disabled when run as a file. You can click on the ⋯ button in the top right corner of a code cell to change that.

Pick a rotation angle α: 1.57

Pick a time step Δt:

> *"The exploration*
> *is the implementation*
> *is the documentation"*

- Crashcourse: Julia and Pluto.jl notebooks
- How to develop packages in Julia
- Create your first literal Julia package as a notebook

Access the GitHub repository!