Lab #3: Web Application with Genie

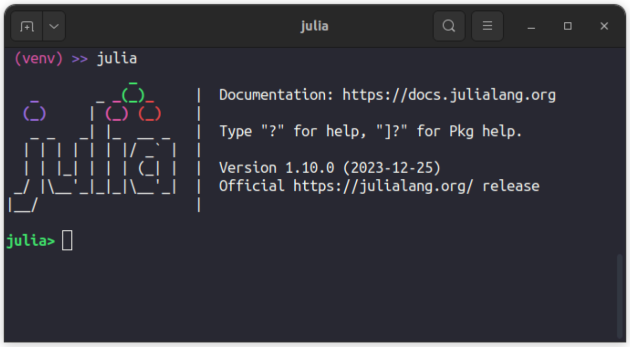
Mohamed Aziz Hajri
Genie electrique
ISET Bizerte — Tunisia
Mohamedazizhajri

Mahjoub Ben Barka
Genie electrique
ISET Bizerte — Tunisia
Mahjoubbenbarka

I. Exercice

A. But

Creat a web app using julia and html coding as in Figure 1.



```
(venv) >> julia
               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.10.0 (2023-12-25)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia>
```

Figure 1: Julia REPL

B. Manipulation

We provide the Julia and HTML codes to build and run a web app that allows us to control the amplitude, the frequency, the offset and the phase of a sine wave. **Plotly** is used to plot the corresponding graph. We also added a slider to change the number of samples used to draw the figure. The latter setting permits to grasp the influence of sampling frequency on the look of our chart.

```julia
using GenieFramework
@genietools

@app begin

    @in N::Int32 = 1000
    @in amp::Float32 = 0.25
    @in freq::Int32 = 1
    @in off::Float32 = 0.0
    @in ph::Float32 = 0.0

    @out my_sine = PlotData()

    @onchange N, amp, freq, ph, off begin
        x = range(0, 1, length=N)
        y = amp*sin.(2*π*freq*x.+ph).+off
        my_sine = PlotData(x=x,
                           y=y,
plot=StipplePlotly.Charts.PLOT_TYPE_LINE)
    end

end

@page("/", "app.jl.html")
```

```html
<header class="st-header q-pa-sm">
    <h1 class="st-header__title text-h3" Sinewave
Dashboard </h1>
</header>

<div class="row">
    <div class="st-col col-12 col-sm st-module">
        <p><b># Samples</b></p>
        <q-slider v-model="N"
    :min="10" :max="1000"
    :step="10" :label="true">
  </q-slider>
    </div>

    <div class="st-col col-12 col-sm st-module">
        <p><b>Amplitude</b></p>
        <q-slider v-model="amp"
    :min="0" :max="3"
    :step=".5" :label="true">
  </q-slider>
    </div>

    <div class="st-col col-12 col-sm st-module">
        <p><b>phase</b></p>
        <q-slider v-model="ph"
    :min="-3.14" :max="3.14"
    :step="0.0314" :label="true">
  </q-slider>
    </div>

    <div class="st-col col-12 col-sm st-module">
        <p><b>offset</b></p>
        <q-slider v-model="off"
    :min=".5" :max="1"
    :step=".1" :label="true">
  </q-slider>
    </div>

    <div class="st-col col-12 col-sm st-module">
        <p><b>Frequency</b></p>
  <q-slider v-model="freq"
    :min="0" :max="10"
    :step="1" :label="true">
  </q-slider>
    </div>
</div>

<div class="row">
    <div class="st-col col-12 col-sm st-module">
  <p><b>Sinewave</b></p>
        <plotly :data="my_sine"> </plotly>
    </div>
</div>
```

```
julia --project
```

```
julia> using GenieFramework
julia> Genie.loadapp() # Load app
julia> up() # Start server
```

We can now open the browser and navigate to the link localhost:8000. We will get the graphical interface as in Figure 2.

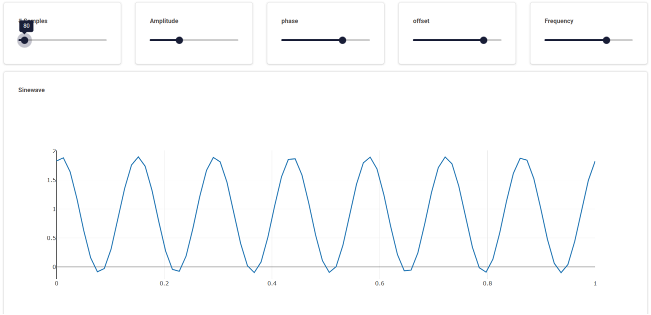


Figure 2: Genie -> Sine Wave

You are asked to add two extra sliders that modify the behaviour of the sine wave graph:

1. *Phase* ranging between $-\pi$ and π, changes by a step of $\frac{\pi}{100}$
2. *Offset* varies from -0.5 to 1, by a step of 0.1.