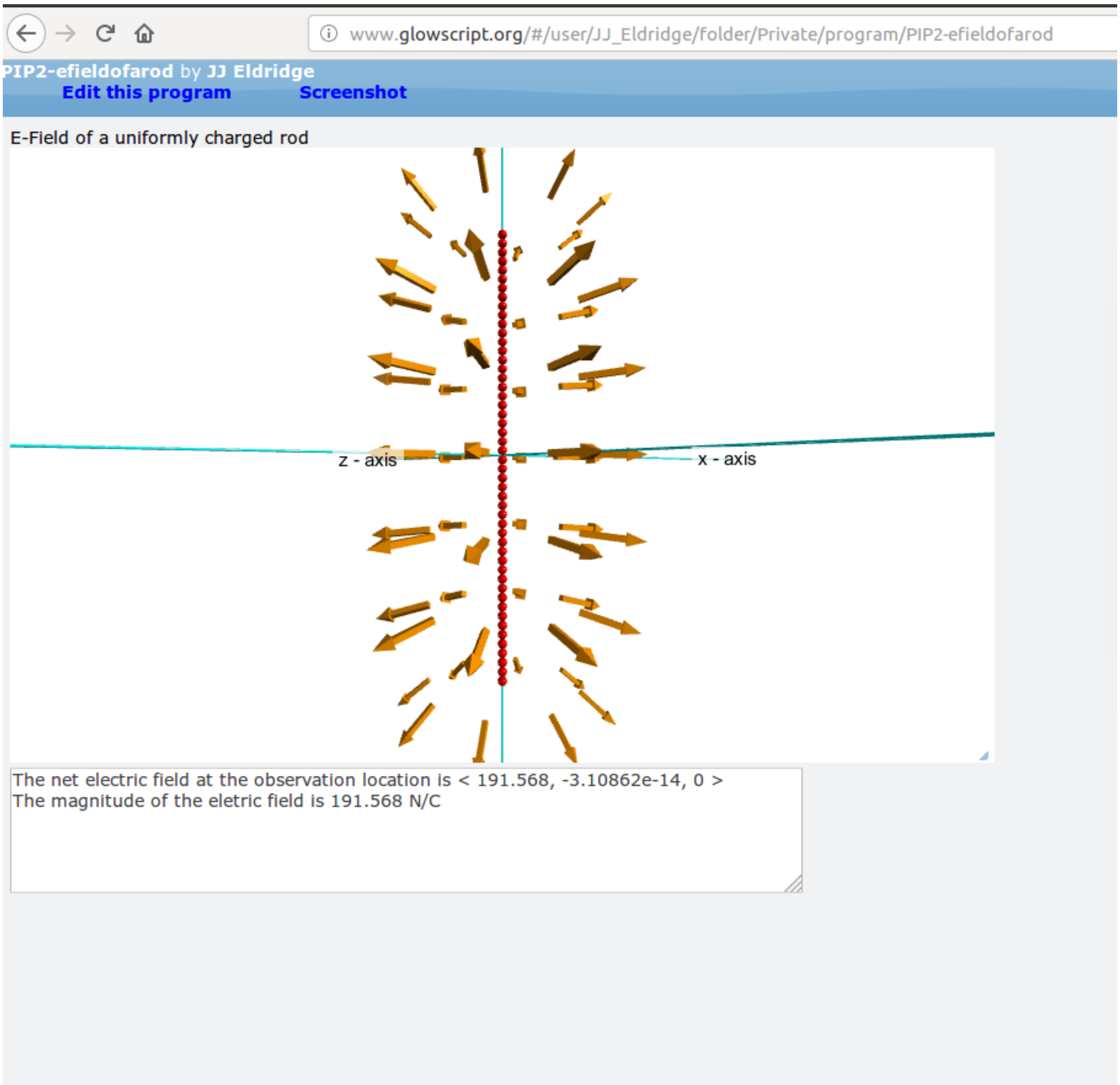
Glowscript running program example:



Following:

1) Glowscript.org coding page

2) Example Jupyter Notebook

PIP2-efieldofarod by **JJ Eldridge**
Run this program **Share or export this program**

Signed in as **JJ Eldridge**(**Sign out**)
Help

```
GlowScript 2.6 VPython
from visual import * # this imports the ivisual package allowing us to draw 3D objects
# Section 1: Defining Constants - Section giving the names and values of any constants

L = 6 # length of the rod see the tutorial description for details
N = 50 # number of point charges, see the tutorial description for details
Q = 6e-8 # total charge on the rod, see the torial description for details
oofpez = 9e9 # creates constant oofpez (One Over Four Pi Epsilon-Zero) and gives it the
scalefactor = 5e-3 # arbitory set scale factor to reduce the size of the electric field


deltay = L/N # symbolic expression for the length of each point-like segment of the rod
deltaQ = Q/N # symbolic expression for the amount of charge given to each point-like se

# initial values

y = 1/2*(deltay - L) # symbolic expression for the y-coordinate of the center of piece
# Section 2: Visual Objects - Setting up the canvas
# right click and drag to rotate the view

scene = canvas(title='E-Field of a uniformly charged rod', background=color.white,  wid

# This creates three rods that follow the x,y,z axis. This allows you to better visuali
rodx = cylinder(pos=vector(-5,0,0), axis=vector(10,0,0), radius=deltay/10, color=color.
rody = cylinder(pos=vector(0,-5,0), axis=vector(0,10,0), radius=deltay/10, color=color.
rodz = cylinder(pos=vector(0,0,-5), axis=vector(0,0,10), radius=deltay/10, color=color.

# This labels the three rods so that it is easier to know which direction is which
label(pos=vector(5,0,0), text='x - axis', color = color.black, background= color.white,
label(pos=vector(0,-5,0), text='y - axis', color = color.black, background= color.white
label(pos=vector(0,0,-5), text='z - axis', color = color.black, background= color.white

# this defines our inital particle, a red sphere located at [1e-10,0,0] with a radius o
# This is much larger then the radius of a proton, 1e-15m, but it makes it easier to vi

#while y < 0.5*L :


obslocation = vector(.9,0,0) # set the observation location

Enet = vector(0,0,0) # set the initial electric field vector at zero

y = -1/2*(L-deltay) # reset y to its inital value

while y < 0.5*L :
    sphere(pos=vector(0,y,0), radius=deltay/2, color=color.red)
    r = obslocation - vector(0,y,0)
    rmag = r.mag
    rhat = r/rmag
    deltaE = (oofpez * (deltaQ/(rmag**2))) * rhat
    Enet = Enet + deltaE
    y = y + deltay

Emag = Enet.mag
print ("The net electric field at the observation location is",Enet)
print ("The magnitude of the eletric field is", Emag ,"N/C")

ea = arrow(pos=(obslocation), axis=scalefactor*Enet, color=color.orange)

Enet = vector(0,0,0) # set the initial electric field vector at zero

```

Dam I forgot my Tea!

I love to drink tea, I follow a very precise method to make my tea. However I am often distracted and forget to drink my tea. Using the laws of thermodynamics predict the time I have to drink my tea before it cools to much. I would like a nice chart to place on my wall so that I can know the tempurature of my tea as a function of time.

Tea recipe

1. 275ml water at a temperature of 100°C
2. 25ml milk at a fridge temperature of 4°C
3. Leave for 4 minutes to reach 70°C
4. My office temperature is 20°C
5. I throw away tea that is 30°C

</head>

Fluid	Density	Heat Capacity
Water	1000 kg/m^3	4198 J/kg
Milk	1033 kg/m^3	3930 J/kg

You will need to use the first law of thermodynamics and Newtons law of cooling.

Newtons law of cooling

Newtons law of cooling states that the rate of heat lost is proportional to the difference between the object and its environment. Ie hotter things cool faster then cooler things!

This can be expressed as

$$\frac{dT}{dt} \propto (T - T_a)$$

Where we introduce the contsant k this is expressed as

$$\frac{dT}{dt} = -k(T - T_a)$$

Where T_a is the ambient tempurature. We add a negtive sign for this to make physical sense. If the tempurature of the object, T is > T_a then we would expect that the rate of heat exchange to be negative ie cooling, and visa versa for T < T_a.

We will make a few assumptions in this model to simply things

1. During the mixing of the milk no heat is lost to the environment and only exchanged between the water and the milk
2. There is negligable energy lost to heat the vessel containing the tea
3. The environment is large and hence the cup of tea does not have an effect to warm up the ambient tempurature of the room.

Hint: You will need to intergrate Newtons law to derivate an equation for Temperature as a function of time.

In [13]:

```
#From 1st Law of thermodynamics, in a closed system, we can say that the energy
 lost by one body is equal to the energy gained by the other

#Ie, Q_loss = Q_gain
vol_water = 275*1e-6 # define the volume of water
vol_milk = 25*1e-6 # define the volume of milk

density_water = 1000 # define the density of water
density_milk = 1033 # define the denisty of milk

heat_capacity_water = 4198 # define the heat capacity of water
heat_capacity_milk = 3930 # define the heat capacity of milk

mass_water = vol_water * density_water # calculate the mass of water
mass_milk = vol_milk * density_milk # calculate the mass of milk

initial_temp_water = 100 # define the initial temp of water
initial_temp_milk = 4 # define the initail temp of milk

#We can say that Q_loss = m1 * C1 * (Initial_temp1 - End_temp) &  Q_gain = m2 *
 C2 * (End_temp - Initial_temp2)

a_1 = mass_water* heat_capacity_water
a_2 = mass_milk* heat_capacity_milk

final_temp = ((a_1*initial_temp_water)+(a_2*initial_temp_milk))/(a_1+a_2) #calcu
late the final tempurature of the mixuture

print('The final temp is',round(final_temp,1), u'\u2103') # prints the final tem
p to 1 decimal place
```

```
The final temp is 92.2 ℃
```

Deriving Newtons law of cooling

$$\frac{dT}{dt} = -k\,(T - T_a)$$

Seperate out variables

$$\frac{1}{T - T_a}\, dT = -k\, dt$$

$$\int \frac{1}{T - T_a}\, dT = \int -k\, dt$$

By substitution

$$u = T - T_a$$

Therefore

$$\frac{du}{dT} = 1$$

$$dT = du$$

$$\int \frac{1}{u} du = ln\, |u|$$

Subsituting back

$$ln\, |T - T_a| = -k\, t + c_1$$

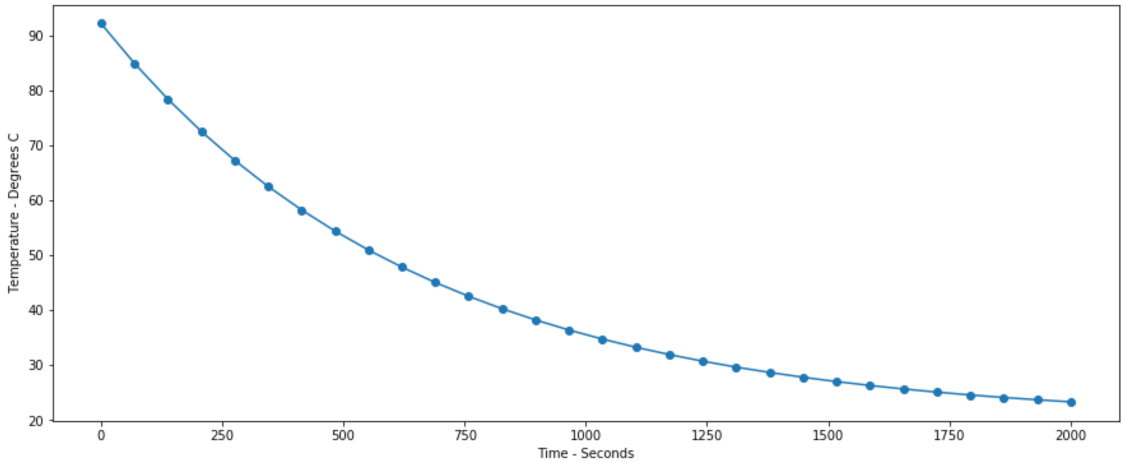
$$\begin{aligned} |T - T_a| &= e^{-k\, t + c_1} \\ &= e^{-k\, t} e^{c_1} \\ &= C\, e^{-k\, t} \end{aligned}$$

If T > T_a ie hotter then the ambient

$$T(t) = C\, e^{-k\, t} + T_a$$

In [14]:

```
from pylab import * # import the pylab package

ambient_temp = 20 # define the ambient temp of the room
temp_4min = 70 # define the temp at 4mins (240 seconds)
delta_t = final_temp - ambient_temp # calculate delta_t
time = 4*60 # define time 4mins

k = (log((temp_4min-ambient_temp)/delta_t))/-time # calculate the k coefficient

#Plot this

t = linspace(0,2000,30) # generate a linspace from 0 to 2000
T = delta_t * exp(-1*k*t) + ambient_temp # calculate the temp for each point
figure(figsize=(15,6)) # create a large figure
plot(t,T, 'o-') # plot the graph
xlabel('Time - Seconds') # set xlabel to 'Time - seconds'
ylabel('Temperature - Degrees C') # set ylabel to 'Temperature - Degrees C'
show() # show the graph
```



Context

Does it make a difference if you add the milk to the cup before the water? Does you answer make sense in relation to the 1st Law of Thermodynamics? What about the asumptions we have made for this model?

Answer:

Given the simplified model we have made, no. As the system is closed and we have ignored the thermal properties of the cup then all energy lost by the water is absorbed by the milk and it does not matter which order they are mixed.

However if we make the model more advanced we would have to take into consideration the effect of the heating of the cup itself. Adding the hot water first would reduce the tempurature of the water as the cup is warmed. This change would be larger then the effect of the cup warming the milk if the milk were to be added first.

Another answer would be looking at the surface area of the liquid. One could argue that adding the water first fills the cup more and gives a larger surface area for the heat to disapate. In the time taken to add the milk the cup has already lost so of its energy and hence would reach throw away tempurature quicker.

We are also assuming that the system is a lumpped heat source. Ie the tempurature inside the cup of tea is uniform. Where infact the tempurature would be cooler at the surface of the liquid and a convection flow would be present in the liquid.

Lets make this more real! The cup I am using is made of plastic and is very well insulated. We can assume that no heat is lost to the environment through the walls. However all heat loss comes from the surface of the liquid in contact with the air. My cup can be modeled as a hollow cylinder and has a radius 5cm and a height 10cm.

The full formula of Newtons law of cooling can be written as,

$$T(t) = (T(0) - T_a) * e^{-r\,t} + T_a$$

where r is

$$r = \frac{h\,A}{C}$$

h is the heat transfer coefficient in W m^{-2} K^{-1}
A is the area of the heat body in contact with the surrounding m^2
C is the heat capacity of the heat source kg m^2 K^{-1} s^{-2}

I also notice that the tempurature of the water and the milk fluctuates during the week

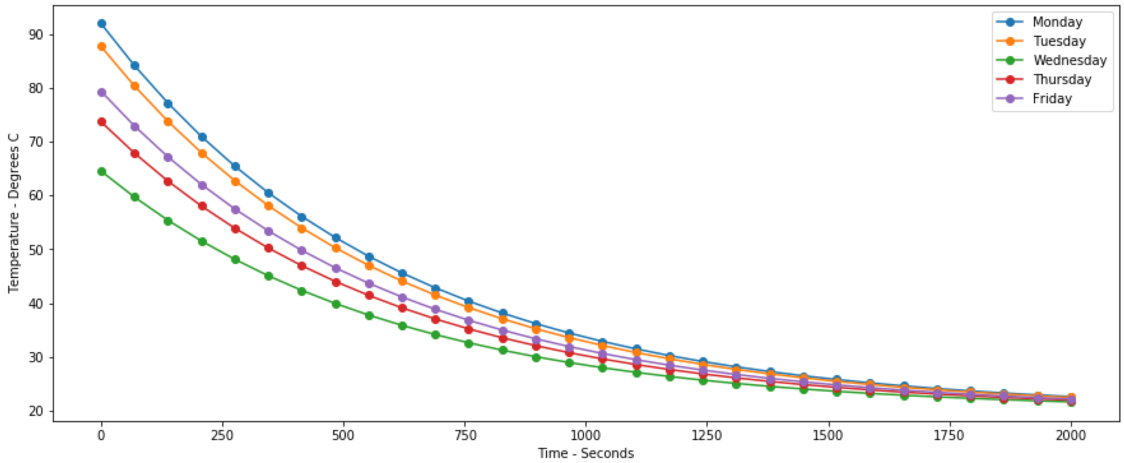
</head>

Day	**Water Temp °C**	**Milk Temp °C**
Monday	100	2
Tuesday	95	6
Wednesday	70	3
Thursday	80	3
Friday	86	5

What is the best and worst day for me to make a cup of tea?

In [15]:

```
# Best to add the tempuratures to a 1D array and loop through the calculations a
nd plot

initial_temp_water = array([100,95,70,80,86]) # define the water temp in an arra
y
initial_temp_milk = array([2,6,3,3,5]) # define the milk temp in an array

final_temp = ((a_1*initial_temp_water)+(a_2*initial_temp_milk))/(a_1+a_2) # calc
ulate the final temp as in first part for each mixture

ambient_temp = 20 # define the ambient temp of the room
temp_4min = 70 # define the temp at 4mins (240 seconds)
delta_t = final_temp - ambient_temp # calculate delta_t

h = 891 # set the heat transfer coefficient to 891
A = pi*(0.05**2) # calculate the area of the cup in contact with air

r = (h * A)/heat_capacity_water # calculate r

t = linspace(0,2000,30) # generate a linspace 0 to 2000

figure(figsize=(15,6)) # create a large figure
for i in range(len(initial_temp_water)): # start a loop the length of the 1D arr
ay
    T = delta_t[i]*exp(-1*r*t) + ambient_temp # calculate the temp date for curr
ent water and milk temp
    plot(t,T, 'o-') # plot this to figure

xlabel('Time - Seconds') # set xlabel to 'Time - seconds'
ylabel('Temperature - Degrees C') # set ylabel to 'Temperature - Degrees C'
legend(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday']) # set the figur
e legend to show the days of the week
show() # show the graph
```



Extension

If you wish to try you can calculate the cooling of the tea as a mixed system. Lets say the cup is made of copper and the heat transfer coefficient if the cup is 300. How long would it take to cool?