BASIC TUTORIAL

Before using the nume module in *IDLE*, the nume module must be imported:

```
> import nume as e
```

or

```
> import nume
```

or

```
> import nume*
```

Calculating the current of resistor R1 with a value of 5.1 kΩ:

```
> v= 12           #voltage 12 V
> r1= e.k(5,1) #resistor 5,1 kΩ
```

next:

```
> i= v/r1
> e.unit(i) #e.unit(v/r1)
(2,352941176470588, 'm') #current ≈2,35mA
```

We convert the obtained number to scientific format:

```
> e.sci(i) #lub e.sci((2,352941176470588, 'm'))
'2,352941176470588e-03'
```

Calculating the resistor that limits the current to 50mA at 12V:

```
> r= v/e.m(50) #R= V/50mA
> e.unit(r)
(240,0, '') #240,0Ω
```

However, we do not have such a resistor, so the closest value in the E12 series will be:

```
> e.valofrow(i)
220,0
```

If this value does not suit us, we must use a parallel connection, e.g. two resistors. We have a 510Ω resistor and we look for a second resistor for 240Ω parallel resistance:

```
> e.parajointfind(240, 510)
453,3333333333333
```

however, we did not find the correct value in the E12 series. Only in the E96 series:

```
> e.valofrow(453) #wartość jest nieodpowiednia
470,0
```

```
> e.valofrow(453, 'E96') #value ok
453,0
```

Impedance Calculation

Calculating the Impedance of a 22n Capacitor at 500 Hz and 1.5 kHz:

```
> import nume*
> import math
> for freq in (k(0,5), k(1,5)):
      print(unit(1/(2*math.pi*freq*n(22))))

(14,468631190172303, 'k')

(4,822877063390768, 'k')
```

Value 5.1k, 500k expressed in M():

```
> k(5,1)/M(1)
0,0051 # 0,0051MΩ
```

```
> k(500)/M(1)
0,5 #0,5MΩ
```

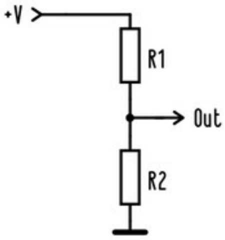
```
> 500/m(1) #500 expressed in m(0,001)
500000,0 #500000m
```

Generating a list of numbers

Generating a list of numbers from smallest to largest, running a *Nota-numbers.py* script in the console or simply clicking the script icon.

Resistive voltage divider

Voltage divider powered by voltage Vc= 600V, VR2= 3.3V (Out):



VR2= 3.3V, Ig= 0.5mA, Vc= 600V.

Calculation of resistance R1+R2

Voltage Vc= 600V, and divider current Ig= 0.5mA. Calculation of the divider resistance resulting from the assumptions:

```
> Vc= 600
> Ig= nume.m(0.5)
> R1R2= Vc/Ig
> nume.unit(R1R2)
(1.2, 'M')
```

Calculation of resistance R2:

```
> VR2= 3.3
> R2= R1R2/(Vc/VR2)
> nume.unit(R2)
(6.599999999999999, 'k')
```

Calculation of resistance R1:

```
> R1= R1R2-(R1R2/(Vc/VR2))
> nume.unit(R1)
(1.1934, 'M')
```

or

```
> R1= R1R2-R2
nume.unit(R1)
(1.1934, 'M')
```

Checking calculations:

```
> Ig= Vc/R1R2
> nume.unit(Ig)
(500.00000000000006, 'u')
```

```
> VR2= (R2/(R1+R2))*Vc
> VR2
3.2999999999999994
```

Power calculation

Now that everything is correct, let's calculate the divisor power:

```
> PR1R2= Vc*Ig
> nume.unit(PR1R2)
(300.0, 'm')     # 0.3W
```

however, the voltage on R1 is higher, so the power stored on it:

```
> VR1= Vc-VR2  # Volty
> VR1
596.7
```

```
> PR1= VR1*Ig
> nume.unit(PR1)
(298.35, 'm')   # ~0.3W
```

while the power stored on resistor R2:

```
> PR2= VR2*Ig
nume.unit(PR2)
(1.65, 'm')    # ~1.7mW
```

Selecting resistance from series

```
R1 1.1934M ~1.19
```

```
R2= 6.599999999999999= 6.59 ~6.6
```

using a simple script to search in selected rows:

```
def ofrows(res):
    for s in ('E12', 'E24', 'E48'):
        r= nume.valofrow(res, s)
        print(r, s)
```

```
> ofrows(1.19) # R1
1.2 E12
1.2 E24
1.21 E48
```

However, there is no corresponding value in all series, but the value of 6.49 from series E48 raises hopes:

```
> ofrows(6.59) # R2
6.8 E12
6.8 E24
6.49 E48
```

```
> ofrows(6.6) # R2
6.8 E12
6.8 E24
6.49 E48
```

In this case, to maintain maximum precision, resistor R2 can be reduced and connected in series with a tuning potentiometer to select the correct resistance. The selection of the value does not have to be so precise:

```
                             /\
          ,---------,   ,----|----,
       ---|    Rz   |---|    |Rp  |---
          '---------'   '----|----'
          \_________ R1 __________/
```

For example:

```
> Rz= (R2/3)*2
> Rp= (R2/3)*2
```

which in total gives:

```
> nume.unit(Rz+Rp)
(8.799999999999999, 'k')
```

```
> nume.unit(Rz)
(4.3999999999999995, 'k')
```

```
> nume.unit(Rp)
(4.3999999999999995, 'k')
```

where with the potentiometer set at half value:

```
> nume.unit(Rz+(Rp/2))
(6.599999999999998, 'k')
```

Divider function

If we want to calculate other dividers for other voltages, it is worth writing a function that will facilitate all calculations:

```
def div2r2(Vc, VR2, Ig):
    r1r2= Vc/Ig
    r2= r1r2/(Vc/VR2)
    r1= r1r2-r2
    pr1r2= Vc*Ig
    pr1= nume.unit((Vc-VR2)*Ig)
    pr2= nume.unit(VR2*Ig)
    r1j= round(nume.unit(r1)[0], 2) #value r1
    r1u= nume.unit(r1)[1]+'Ω'       #unit r1
    r2j= round(nume.unit(r2)[0], 2) #value r2
    r2u= nume.unit(r2)[1]+'Ω'       #unit r2
    pr1j= round(pr1[0], 2) #power value r1
    pr1u= pr1[1]+'W'       #power unit r1
    pr2j= round(pr2[0], 2) #power value r2
    pr2u= pr2[1]+'W'       #power unit r2
    print('R1:', r1j, r1u, ', PR1:', pr1j, pr1u)
    print('R2:', r2j, r2u, ', PR2:', pr2j, pr2u)
```

Executing the above *div2r2()* function in *IDLE*:

```
> div2r2(600, 3.3, nume.m(0.5))
R1: 1.19 MΩ , PR1: 298.35 mW
R2: 6.6 kΩ , PR2: 1.65 mW
```

for voltage 600V, Ig= 0.25mA the function shows less power accumulated on resistors. The results show a reduction in power accumulated on the divider:

```
> div2r2(600, 3.3, nume.m(0.25))
R1: 2.39 MΩ , PR1: 149.18 mW
R2: 13.2 kΩ , PR2: 825.0 uW
```

for voltage 200V:

```
> div2r2(200, 3.3, nume.m(0.25))
R1: 786.8 kΩ , PR1: 49.17 mW
R2: 13.2 kΩ , PR2: 825.0 uW
```

If we need this function, it is worth saving it as a module for future use.