Initialization code: get files from the internet

In [2]:

```
!wget https://raw.githubusercontent.com/ivpcl/REVISION-3-Level-1-2019-English/main/Level1-Projects/AOLME_Fracti
!wget https://raw.githubusercontent.com/ivpcl/REVISION-3-Level-1-2019-English/main/Level1-Projects/Bob.jpg
from AOLME_Fraction_v2 import FrV
from IPython.display import HTML
```

```
--2023-03-15 14:37:11--  https://raw.githubusercontent.com/ivpcl/REVISION-3-Level-1-2019-English/main/Level1-Pr
ojects/AOLME_Fraction_v2.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.11
0.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 47739 (47K) [text/plain]
Saving to: ‘AOLME_Fraction_v2.py.1’

AOLME_Fraction_v2.p 100%[===================>]  46.62K  --.-KB/s    in 0.001s

2023-03-15 14:37:11 (52.3 MB/s) - ‘AOLME_Fraction_v2.py.1’ saved [47739/47739]

--2023-03-15 14:37:11--  https://raw.githubusercontent.com/ivpcl/REVISION-3-Level-1-2019-English/main/Level1-Pr
ojects/Bob.jpg
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.108.133, 185.199.11
0.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 37538 (37K) [image/jpeg]
Saving to: ‘Bob.jpg.1’

Bob.jpg.1           100%[===================>]  36.66K  --.-KB/s    in 0s

2023-03-15 14:37:11 (93.0 MB/s) - ‘Bob.jpg.1’ saved [37538/37538]
```

Making Videos with Fraction Objects

The fraction objects can be used to make videos.

After we create the pictures, the following code creates and displays a video.

The video is specified by assigning two variables:

```
video_name = "myvideo.mp4"
my_fps = 1.0
```

The `video_name` defines the name of the video file. This file is stored on your local directory.

The variable `my_fps` refers to the number of frames per second that we will display the video. Thus, `my_fps=1` means that the video will be displayed at the rate of one frame every second.

Once the video has been created, the fraction objects have a special function called `CreateVideo(video_name, fps=my_fps)` that creates the video.

To display the video on your browser, we need to pass the video output to the `HTML()` function as given by:

```
HTML(frac.CreateVideo(video_name, fps=0.5))
```

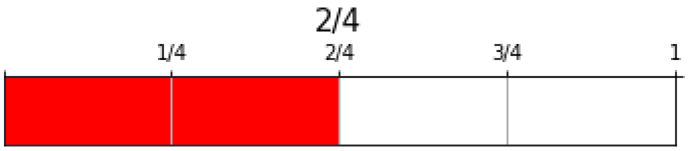
Once the video is created, you can click on it's window to download and save it.

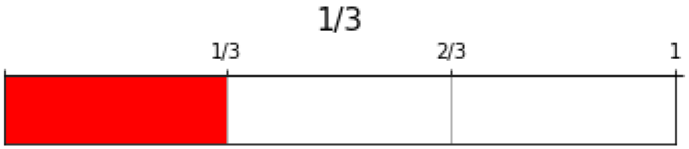
Run the code below to see how it works.

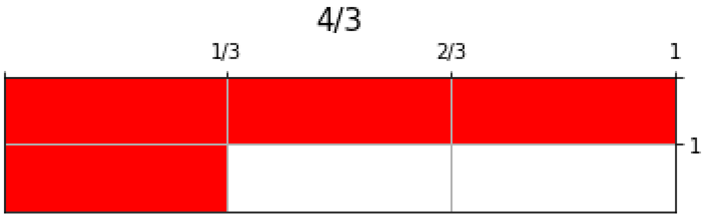
Adjust the number of frames per second to change how the video appears!

In [9]:

```
# Video creation demo
frac = FrV()
f1 = frac.AddFrac(2, 4, comment="2/4")
f2 = frac.AddFrac(1, 3, comment="1/3")
f3 = frac.AddFrac(4, 3, comment="4/3")

# Create and display the video:
video_name = "myvideo.mp4"
my_fps     = 500.0
HTML(frac.CreateVideo(video_name, fps=my_fps))
```

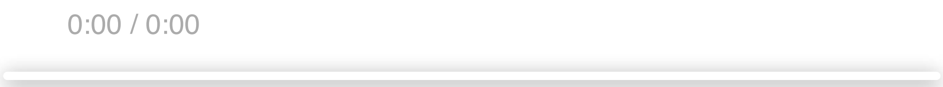






```
Compressed myvideo.mp4 into temp_video.mp4
```

Out[9]:



Creating a video for multiplication

Fractions can create a video of the multiplication process for you!

We specify the multiplication using:

```
c = "Video of 1/3 * 3"
num  = 1 # Numerator
den  = 3 # Denominator
mult = 3 # Multiplier
```

Here, c holds a comment.

As before, 1/3 mean the numerator (num) is 1 and the denominator (den) is 3.

We can then add the video frames to our video using:

```
frac.AddMult(num, den, mult, comment=c)
```

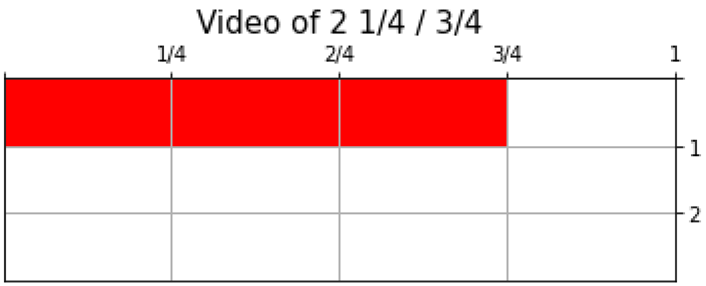
Run the code below to see how it works!

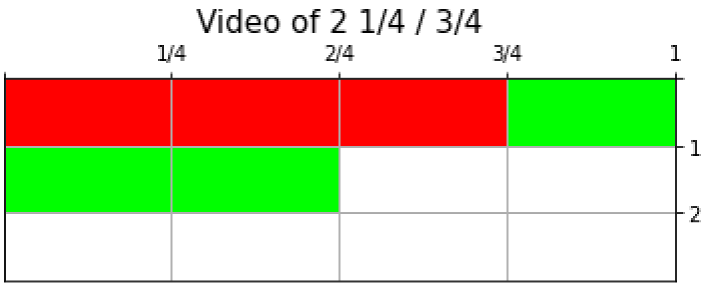
Note that we also have to save and display the video as before.

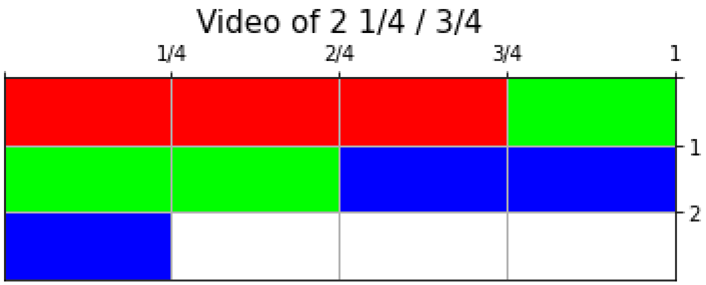
In [25]:

```
# Create the fraction object
frac = FrV()

c = "Video of 2 1/4 / 3/4"
num  = 3 # Numerator
den  = 4# Denominator
mult = 3 # Multplier
frac.AddMult(num, den, mult, comment=c)

#c = "Video of 1 / 1/4"
#num  = 1 # Numerator
#den  = 4# Denominator
#mult = 4 # Multplier
#frac.AddMult(num, den, mult, comment=c)

#c = "Video of 5/6 / 1/6"
#num  = 1   # Numerator
#den  = 6# Denominator
#mult = 5# Multplier


#frac.AddMult(num, den, mult, comment=c)

# Create and display the video:
video_name = "michaeljakson.mp4"
```

```
my_fps = 8
HTML(frac.CreateVideo(video_name, fps=my_fps))
```







```
Compressed michaeljakson.mp4 into temp_video.mp4
```

Out[25]:

0:00 / 0:00

Adding text to your video

You can add simple text to your video using:

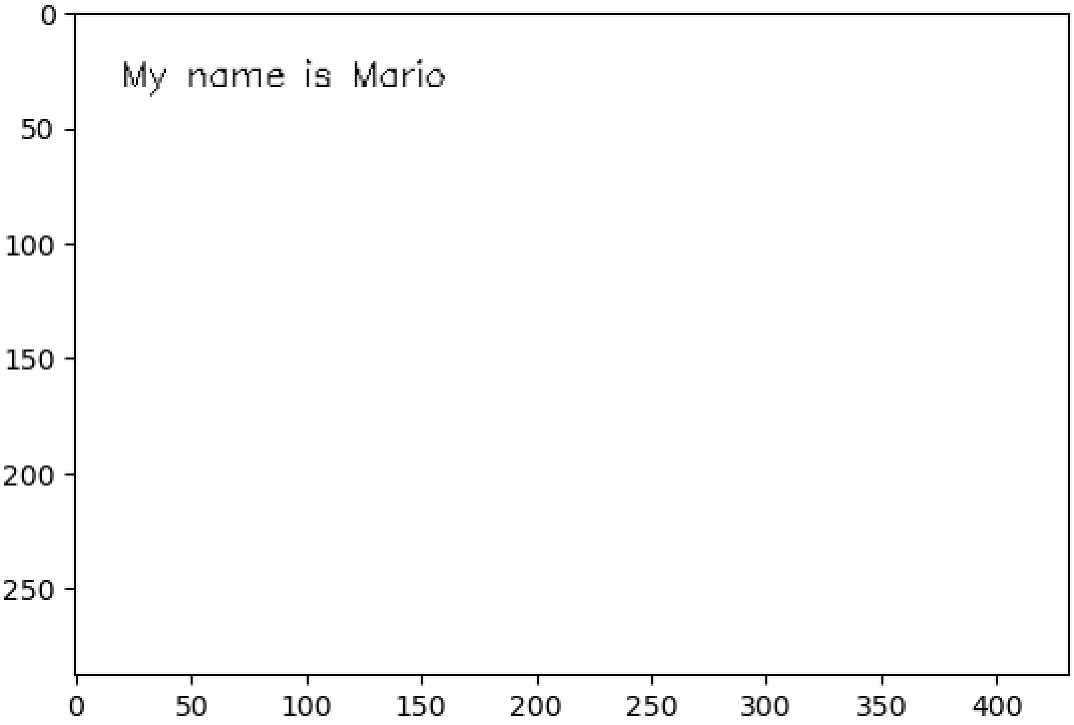
```
my_string = "My name is Mario"
frac.addTextFrame(text=my_string)
```

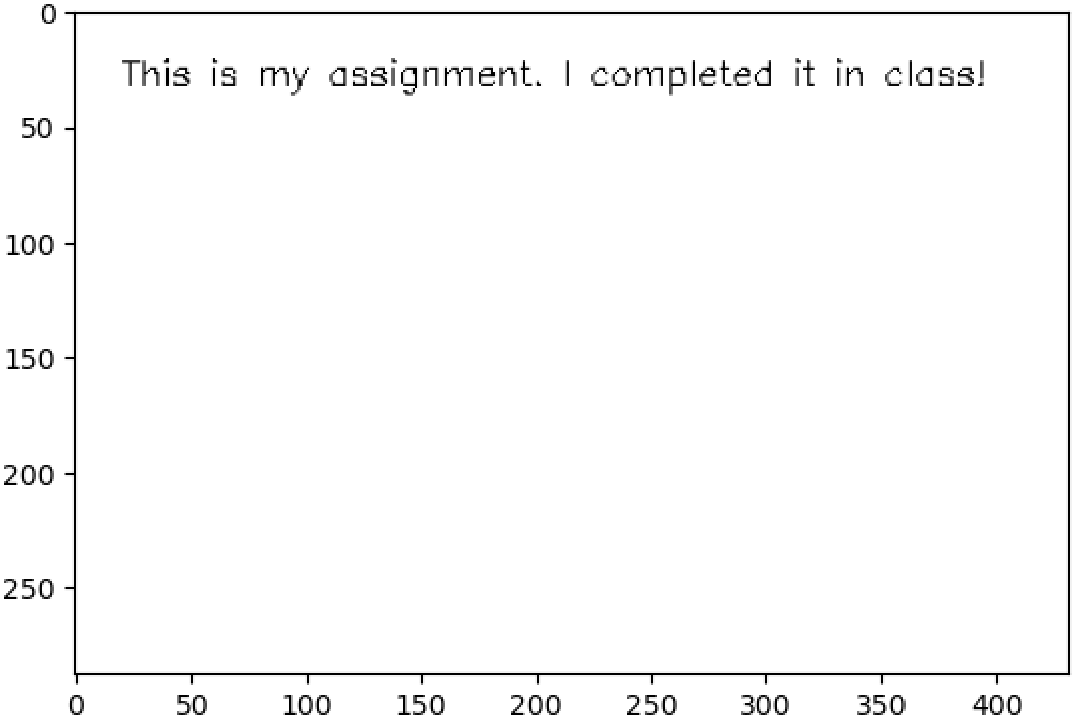
Run the code below to see how you can add text.

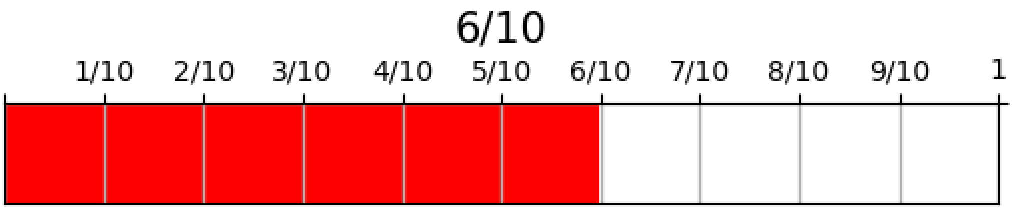
In [ ]:

```
# Create the video object
frac = FrV()

# Add the name
my_string = "My name is Mario"
frac.addTextFrame(text=my_string)

# Add some more text
my_string = "This is my assignment. I completed it in class!"
frac.addTextFrame(text=my_string)
```

```
# A simple fraction:
frac1 = frac.AddFrac(6, 10, comment="6/10")

# Display the video:
HTML(frac.CreateVideo(video_name, fps=0.25))
```







```
[ WARN:0@1.377] global /private/var/folders/sy/f16zz6x50xz3113nwtb9bvq00000gp/T/abs_506zufg7xt/croots/recipe/op
encv-suite_1664548331847/work/modules/videoio/src/cap_gstreamer.cpp (2180) open OpenCV | GStreamer warning: can
not link elements
2023-01-11 15:03:43.790 python[4077:34353] AVF: AVAssetWriter status: Cannot Save
```

```
Out[ ]:
```

0:00