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
```

```
--2023-03-16 14:34:06--  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)... 185.199.110.133, 185.199.109.133, 185.199.10
8.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 47739 (47K) [text/plain]
Saving to: 'AOLME_Fraction_v2.py'

AOLME_Fraction_v2.p 100%[===================>]  46.62K  --.-KB/s    in 0.008s

2023-03-16 14:34:06 (5.97 MB/s) - 'AOLME_Fraction_v2.py' saved [47739/47739]

--2023-03-16 14:34:06--  https://raw.githubusercontent.com/ivpcl/REVISION-3-Level-1-2019-English/main/Level1-Pr
ojects/Bob.jpg
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.111.133, 185.199.10
9.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 37538 (37K) [image/jpeg]
Saving to: 'Bob.jpg'

Bob.jpg              100%[===================>]  36.66K  --.-KB/s    in 0.003s

2023-03-16 14:34:06 (13.1 MB/s) - 'Bob.jpg' 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 [ ]:
```python
# 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))
```
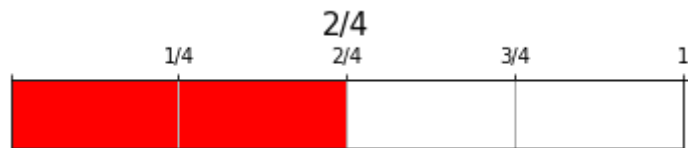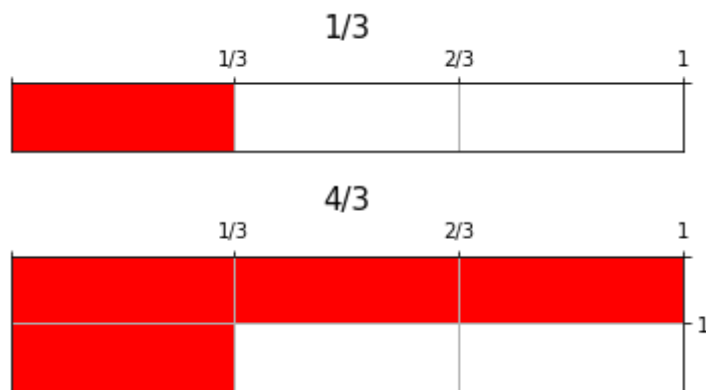
Compressed myvideo.mp4 into temp_video.mp4

Out[ ]:

0:00 / 0:00

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 [5]:
```python
# Create the fraction object
frac = FrV()

#PROJECT QUESTION
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:
```
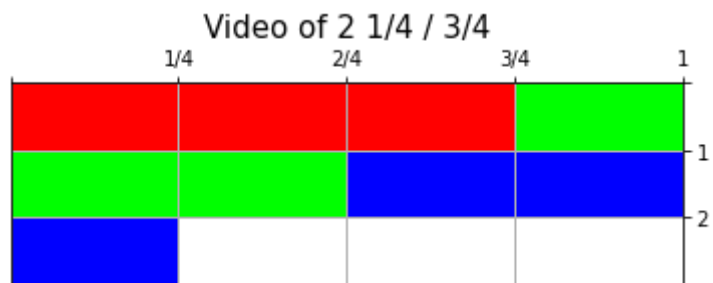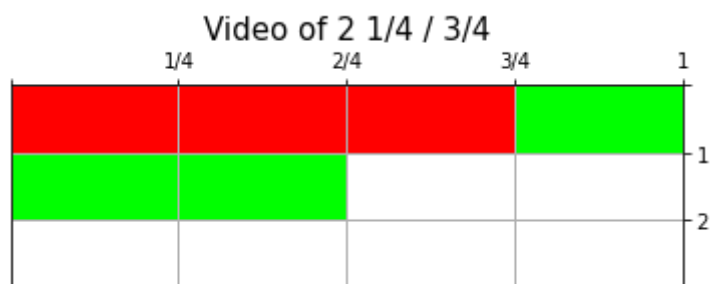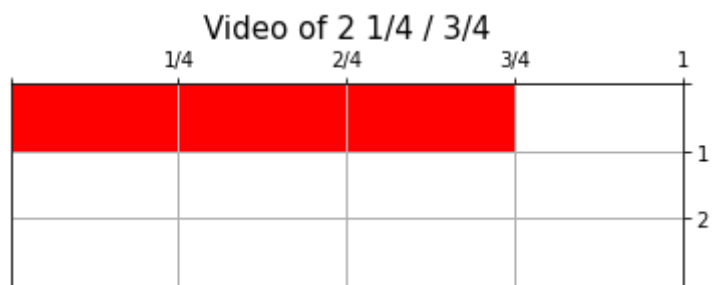
```
video_name = "michaeljakson.mp4"
my_fps = 2.5
HTML(frac.CreateVideo(video_name, fps=my_fps))
```

Video of 2 1/4 / 3/4

Video of 2 1/4 / 3/4

Video of 2 1/4 / 3/4

Compressed michaeljakson.mp4 into temp_video.mp4

`Out[5]:`

0:00 / 0:01

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.

```python
In [30]:   # Create the video object
           frac = FrV()

           # Add the name
           my_string = "Our names are Jayden,Brielle,Joaquin and we are the Crepes. "
           frac.addTextFrame(text=my_string)

           #Crepe pic
           frac.insertImage('crepe.jpeg')
```

```python
# Add some more text
my_string = "Our equation is A chef used 2 and 1/4 cups of flour while cooking. \n \
The recipe called for 3/4 cups of flour.How many batches did the chef make?"
frac.addTextFrame(text=my_string)

#rat chef
frac.insertImage('Chefrat.jpeg')
#greg
frac.insertImage('greg.png')
#decfrcfrfrfood truck
frac.insertImage('foodtruck.jpeg')

# Equation
my_string = "Our equation is 2and 1/4 / 3/4= 3. \n 2 1/4 is the total cups of flour 3/4 is how much needed per
frac.addTextFrame(text=my_string)

# Add the name
my_string = "2 and 1/4 cups of flour is the total.We split the 2 in to 4th which was 8/4+1/4=9/4.so the total w

frac.addTextFrame(text=my_string)

# Add the name
my_string = "and the recipe is 3/4 cups of flour so we have to group 3 squares and the amount of squares is nin

frac.addTextFrame(text=my_string)

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

HTML(frac.CreateVideo(video_name, fps=0.25))
```

Our names are Jayden,Brielle,Joaquin and we
are the Crepes.

Our equation is A chef used 2 and 1/4 cups of flour while cooking.
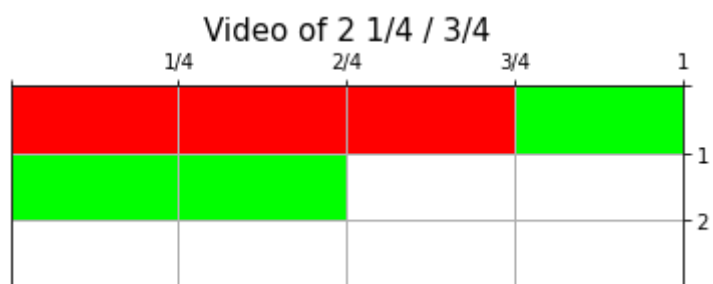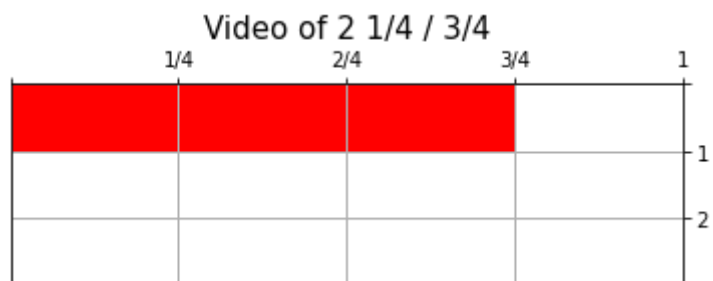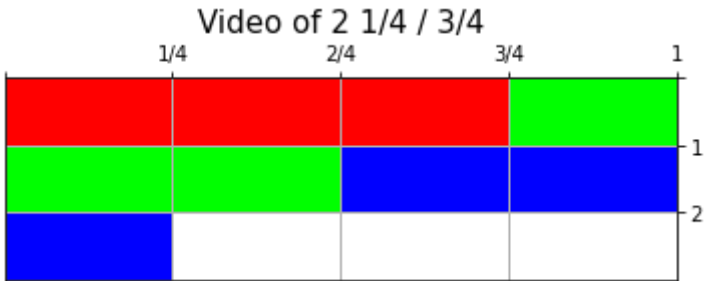The recipe called for 3/4 cups of flour.How many batches did the chef make?

Our equation is 2and 1/4 /
3/4= 3.
2 1/4 is the total cups of
flour 3/4 is how much needed
per recipe the 3 is the
batches that the chef made.

2 and 1/4 cups of flour is
the total.We split the 2 in to
4th which was
8/4+1/4=9/4.so the total
was 9 squares every square is
1/4.

and the recipe is 3/4 cups of
flour so we have to group 3
squares and the amount of
squares is nine. so 3 batches
were made

Video of 2 1/4 / 3/4

Video of 2 1/4 / 3/4

Video of 2 1/4 / 3/4

Compressed michaeljakson.mp4 into temp_video.mp4

Out[30]:

0:00 / 0:48