Initialization code: get files from the internet

In [1]:
```
!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 16:58:35--  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)... 185.199.111.133, 185.199.110.133, 185.199.10
9.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|: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 0.01s

2023-03-16 16:58:35 (4.25 MB/s) - 'AOLME_Fraction_v2.py' saved [47739/47739]

--2023-03-16 16:58:35--  https://raw.githubusercontent.com/ivpcl/REVISION-3-Level-1-2019-English/main/Level1-Pr
ojects/Bob.jpg
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.11
1.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|: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 0.005s

2023-03-16 16:58:35 (7.22 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))
```

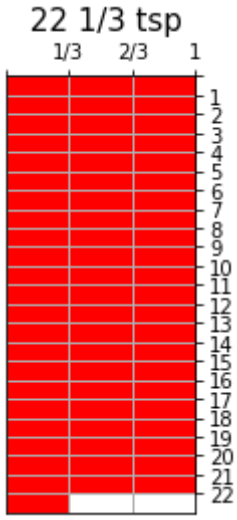
Once the video is created, you can click on it's window to download and save it.

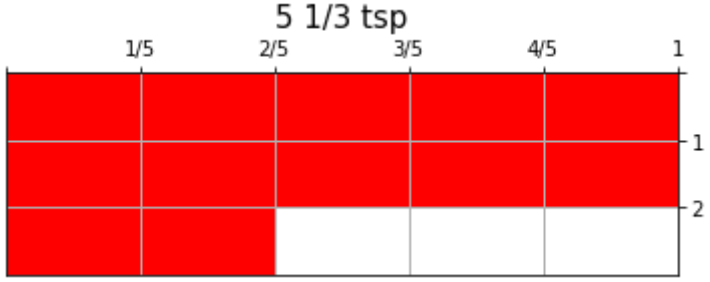
Run the code below to see how it works.

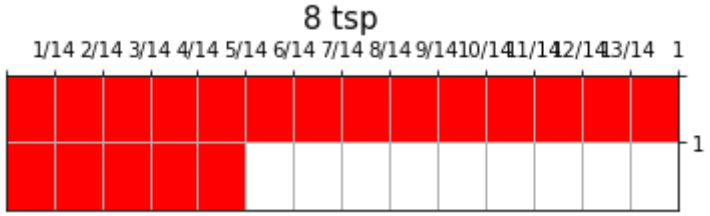
Adjust the number of frames per second to change how the video appears!

In [2]:

```
# Video creation demo
frac = FrV()
f1 = frac.AddFrac(67,3, comment=" 22 1/3 tsp")
f2 = frac.AddFrac(12,5, comment="5 1/3 tsp")
f3 = frac.AddFrac(19,14, comment="8 tsp")

# Create and display the video:
video_name = "myvideo.mp4"
my_fps=      2
HTML(frac.CreateVideo(video_name, fps=my_fps))
```








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

Out[2]:

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.

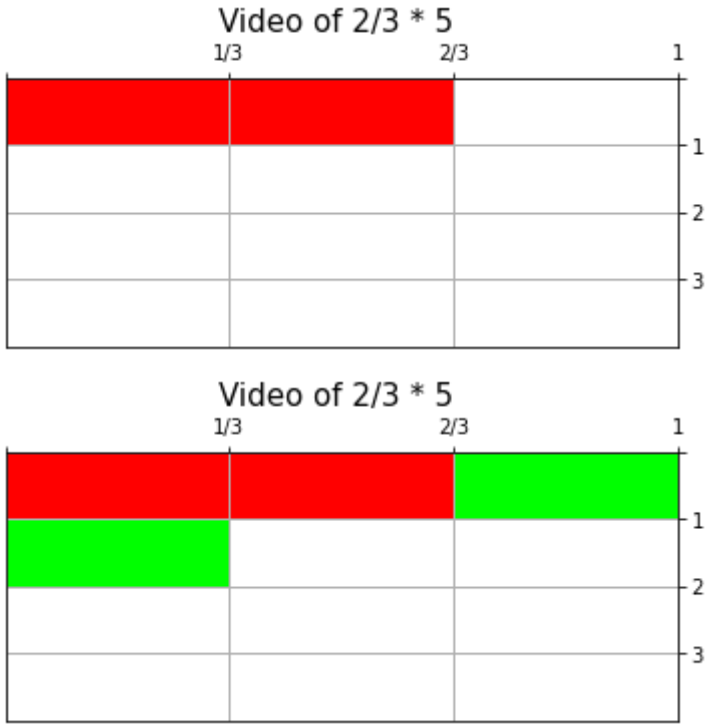
We can then add the video frames to our video using:

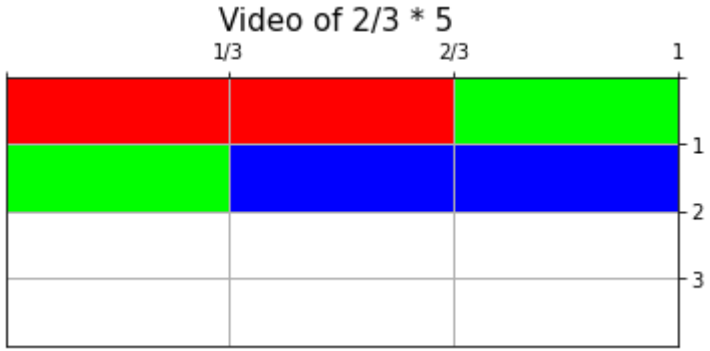
```
frac.AddMult(num, den, mult, comment=c)
```

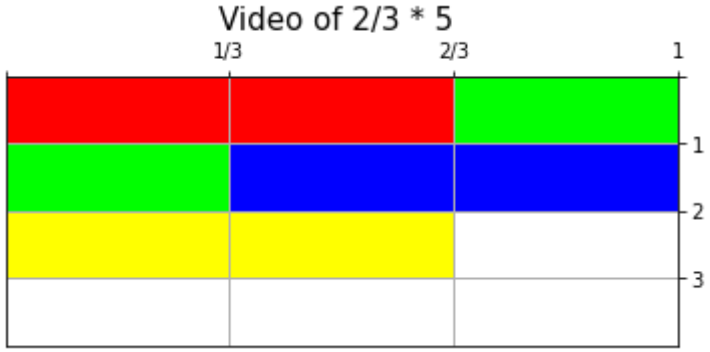
Run the code below to see how it works!

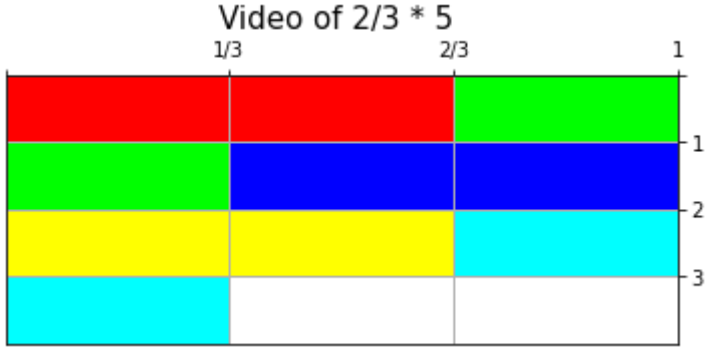
Note that we also have to save and display the video as before.

In [3]:
```
# Create the fraction object
frac = FrV()

c = "Video of 2/3 * 5 "
num  = 2 # Numerator
den  = 3 # Denominator
mult = 5 # Multplier

frac.AddMult(num, den, mult, comment=c)
# Create and display the video:
video_name = "video.mp4"
my_fps = 2
HTML(frac.CreateVideo(video_name, fps=my_fps))
```









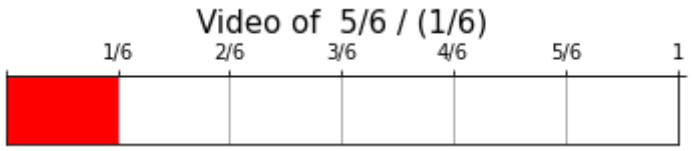

```
Compressed video.mp4 into temp_video.mp4
```

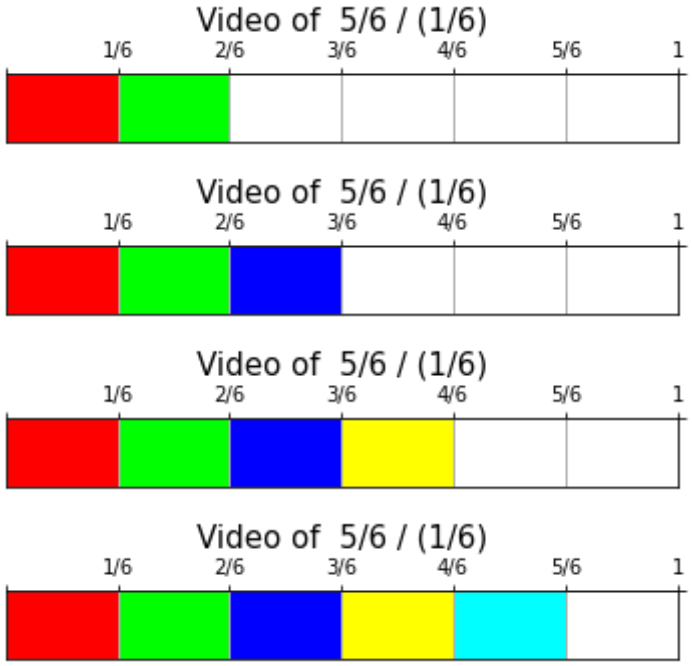
Out[3]:

0:00 / 0:02

In [4]:

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

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 = "video.mp4"
my_fps = 2
HTML(frac.CreateVideo(video_name, fps=my_fps))
```





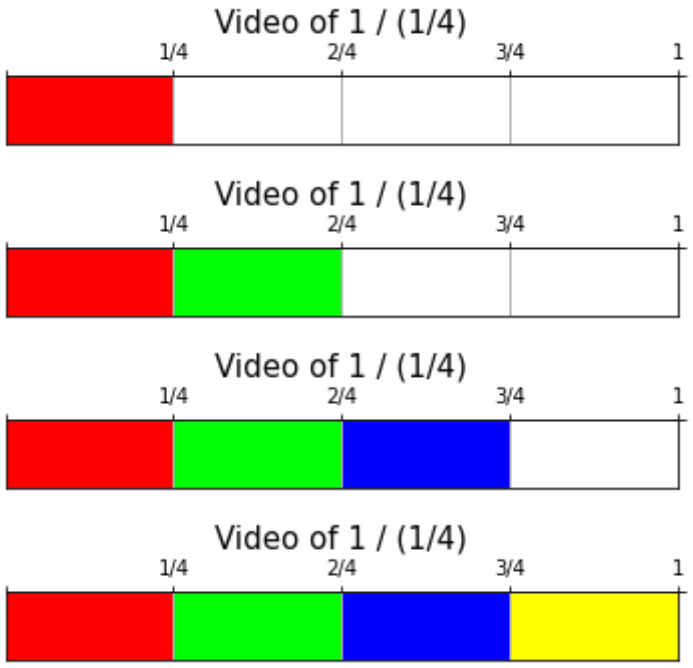

```
Compressed video.mp4 into temp_video.mp4
```

Out[4]:

0:00 / 0:02

In [5]:

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

c = "Video of 1 / (1/4)"
num  = 1 # Numerator
den  = 4 # Denominator
mult = 4 # Multplier

frac.AddMult(num, den, mult, comment=c)
# Create and display the video:
video_name = "video.mp4"
my_fps = 2
HTML(frac.CreateVideo(video_name, fps=my_fps))
```

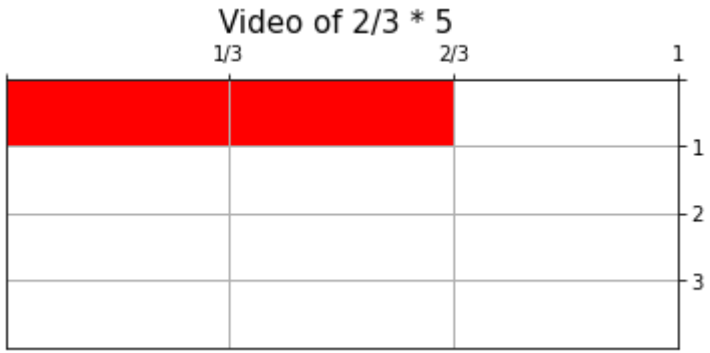


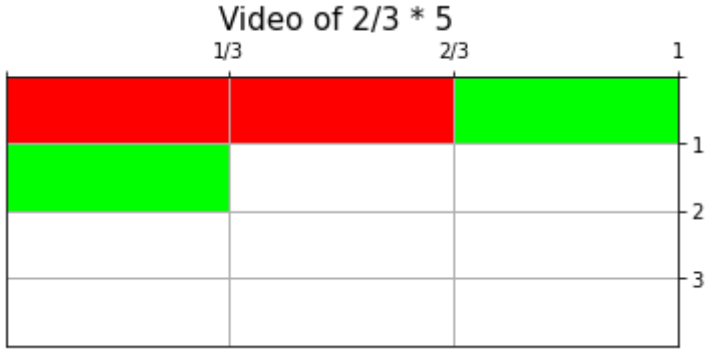

```
Compressed video.mp4 into temp_video.mp4
```

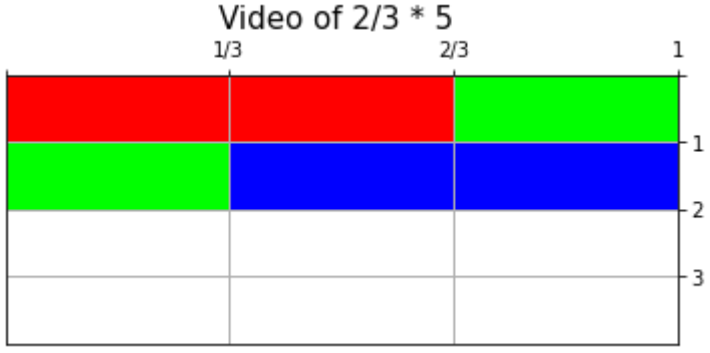
Out[5]:

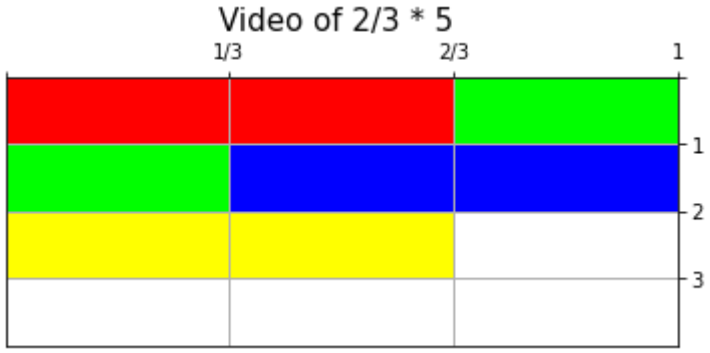
In [6]:

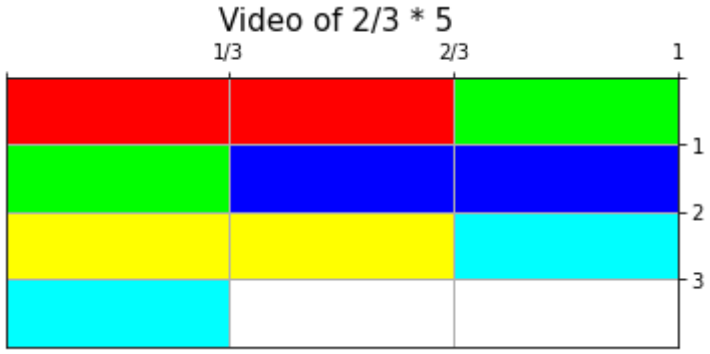
```
# Create the fraction object
frac = FrV()

c = "Video of 2/3 * 5 "
num  = 2 # Numerator
den  = 3 # Denominator
mult = 5 # Multplier

frac.AddMult(num, den, mult, comment=c)
# Create and display the video:
video_name = "video.mp4"
my_fps = 2
HTML(frac.CreateVideo(video_name, fps=my_fps))
```


Video of 2/3 * 5
1/3
2/3
1
1
2
3


Video of 2/3 * 5
1/3
2/3
1
1
2
3


Video of 2/3 * 5
1/3
2/3
1
1
2
3





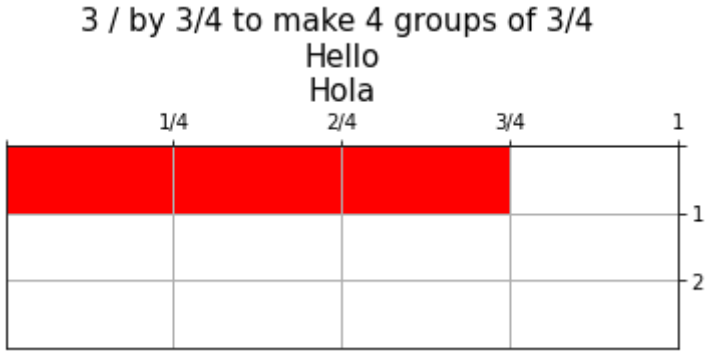

```
Compressed video.mp4 into temp_video.mp4
```

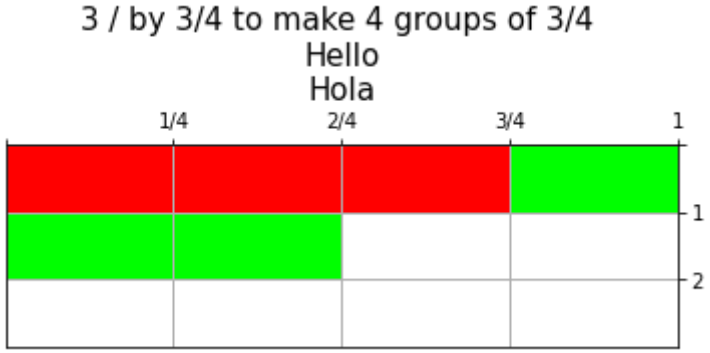
Out[6]:

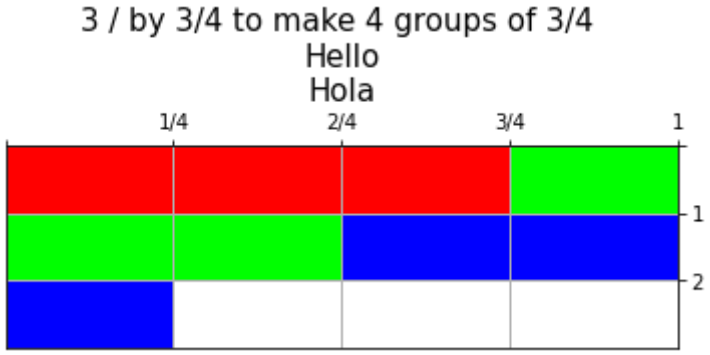
0:00 / 0:02

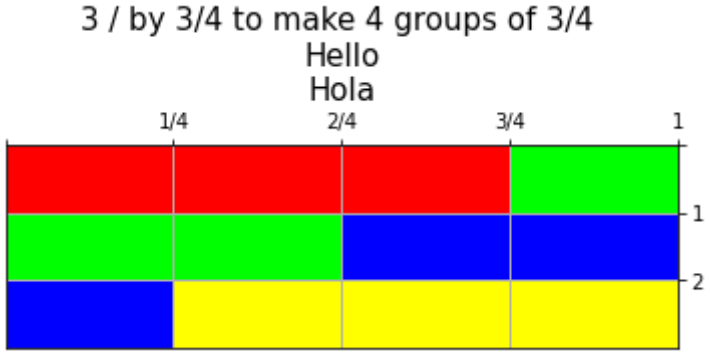
In [7]:

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

c = "3 / by 3/4 to make 4 groups of 3/4 \nHello\nHola"
num  = 3 # Numerator
den  = 4 # Denominator
mult =  4 #Multplier

frac.AddMult(num, den, mult, comment=c)
# Create and display the video:
video_name = "video.mp4"
my_fps = 2
HTML(frac.CreateVideo(video_name, fps=my_fps))
```










```
Compressed video.mp4 into temp_video.mp4
```

Out[7]:

0:00 / 0:02

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)
```

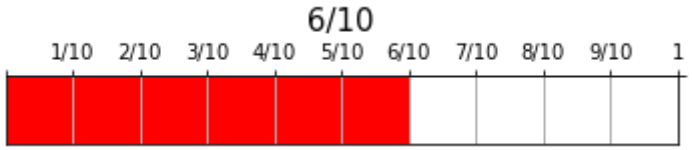
Run the code below to see how you can add text.

In [ ]:

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

# Add the name
my_string = "i love dogs"
frac.addTextFrame(text=my_string)

# Add some more text
my_string = "cats"
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))
```

```
i love dogs
```

```
cats
```




```
Compressed video.mp4 into temp_video.mp4
```

Out[ ]:

Adding long texts with multiple lines

You can add multiple lines of text using `\n \` at the end of each line. After that, you need to start at the begining of the following line.

Thus, the following code adds a long string with multiple lines:

```
my_string ="Fraction Division Assignment\n \
Marios S. Pattichis and Sylvia Celedon-Pattichis\n \
March 28, 2023"
frac.addTextFrame(text=my_string)
```

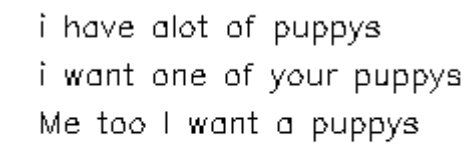
**Make sure that there is no extra space after **

Modify the code below to write your own message.

In [ ]:

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

```
# Add some text
my_string ="i have alot of puppys \n \
i want one of your puppys\n \
Me too I want a puppys"
frac.addTextFrame(text=my_string)

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




```
Compressed video.mp4 into temp_video.mp4
```

Out[ ]:

0:00 / 0:04

Adding pictures to your video

You can add JPEG images to your videos.

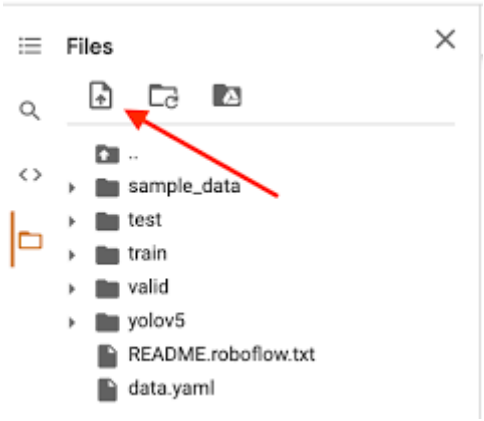
To do this, simply upload the image to your local directory.

This is done using:

1. Click on the folder icon on the left of your browser.

Folder icon

1. Click on file upload icon to upload your picture.



Upload file icon

In the code, you need to add a line with the name of your image file:

```
frac.insertImage('my_image_filename.jpg')
```

Try the code below and see how it works.

In [ ]:

```
frac = FrV()

frac.insertImage('Bob.jpg')

# Create a video
```

```
video_name = "video.mp4"
HTML(frac.CreateVideo(video_name, fps=1))
```

```
Compressed video.mp4 into temp_video.mp4
```

Out[ ]:

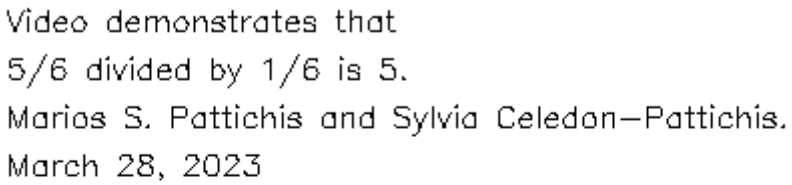
0:00 / 0:01

Putting it all together

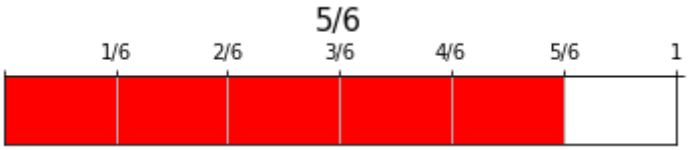
In [ ]:

```
# Demo 5/6 / 1/6
frac = FrV()

my_string = "Video demonstrates that\n 5/6 divided by 1/6 is 5.\n \
Marios S. Pattichis and Sylvia Celedon-Pattichis.\n \
March 28, 2023"
frac.addTextFrame(text=my_string)

frac.insertImage('Bob.jpg')

my_string = "First, we show 5/6."
```

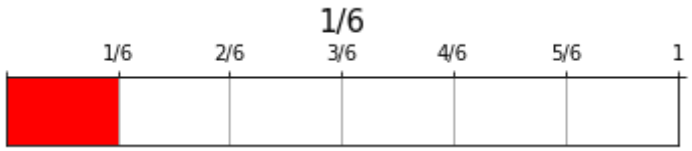
```
frac.addTextFrame(text=my_string)
frac1 = frac.AddFrac(5, 6, comment="5/6")

my_string = "Second, we show 1/6."
frac.addTextFrame(text=my_string)
frac.AddFrac(1, 6, comment="1/6")

my_string = "We multiply 1/6 5 times\n \
to get 5/6."
frac.addTextFrame(text=my_string)
frac.AddMult(1, 6, 5, comment = '5*1/6 is 5/6')


my_string = "Thank you :-)"
frac.addTextFrame(text=my_string)


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



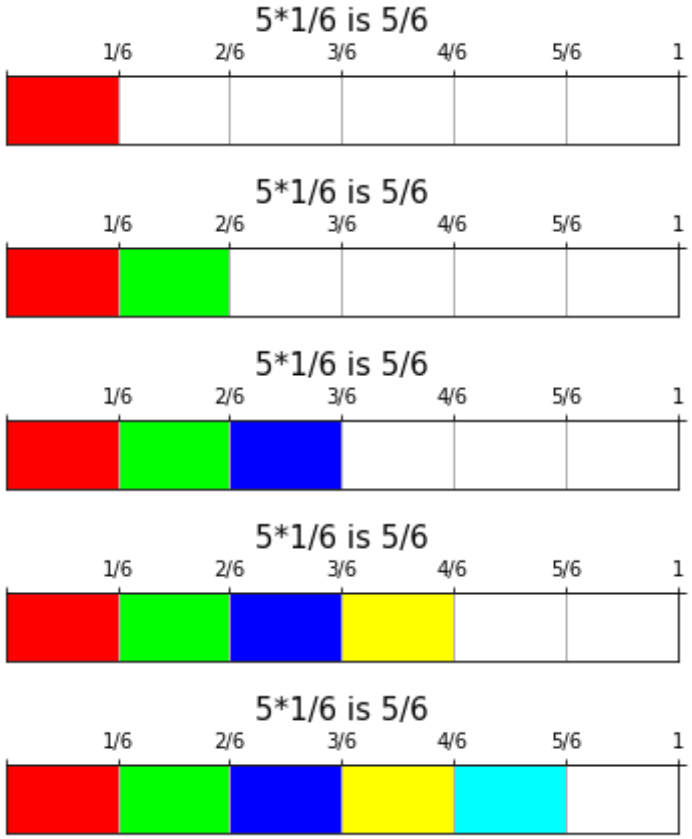
First, we show 5/6.



Second, we show 1/6.



We multiply 1/6 5 times
to get 5/6.



Thank you :-)

```
Compressed video.mp4 into temp_video.mp4
```

Out[ ]:

In [ ]:

Adding text to your video

You can add simple text to your video using:

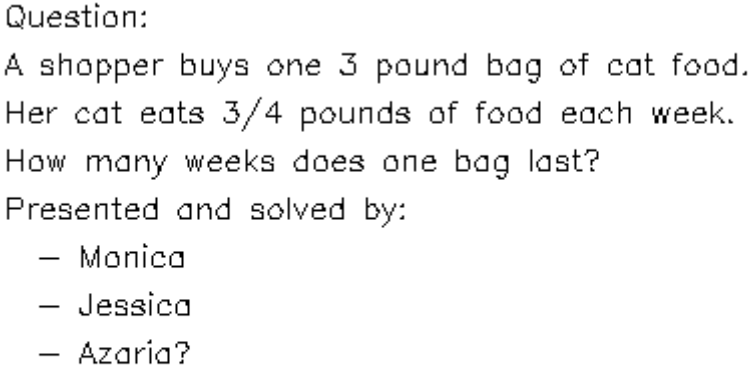
```
my_string = "My name is Mario"
frac.addTextFrame(text=my_string)
```

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

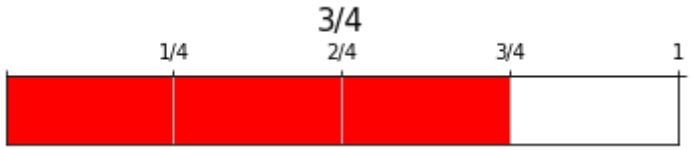
In [18]:

```
# Demo 5/6 / 1/6
frac = FrV()

my_string = """Question:\n \
A shopper buys one 3 pound bag of cat food. Her cat eats 3/4 pounds of food each week. How many weeks does one
Presented and solved by:\n \
  - Monica \n \
```

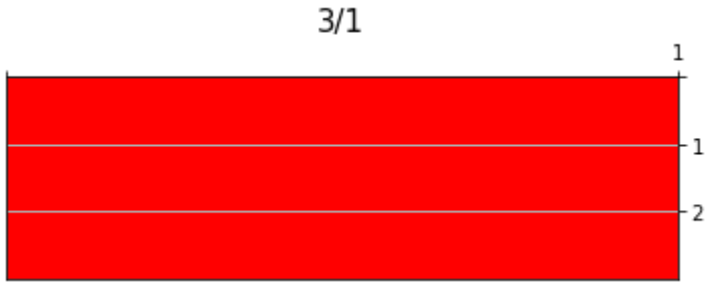
```
  - Jessica\n \
  - Azaria
"""
frac.addTextFrame(text=my_string)

frac.insertImage('girls1.jpeg')

my_string = "First,we start with 3/4 which represents the amount of food the cat eats for each week."
frac.addTextFrame(text=my_string)
frac1 = frac.AddFrac(3, 4, comment="3/4")

my_string = "Second, we show 3 which means how many pounds of cat food we have ."
frac.addTextFrame(text=my_string)
frac.AddFrac(3,1 , comment="3/1")

my_string = "3 divided by 3/4.We want to know how many 3/4's can fit into 3."
frac.addTextFrame(text=my_string)
frac.AddMult(3, 4, 4, comment = '3 / (3/4) = 4. 4 means how many weeks one bag lasts.')


my_string = "Thank you :-)"
frac.addTextFrame(text=my_string)

frac.insertImage("poonie.jpeg")

HTML(frac.CreateVideo(video_name, fps=0.2))
```



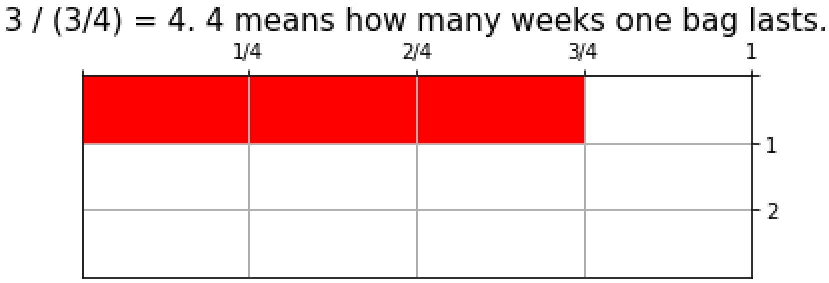
First,we start with 3/4 which represents the amount of food the cat eats for each week.

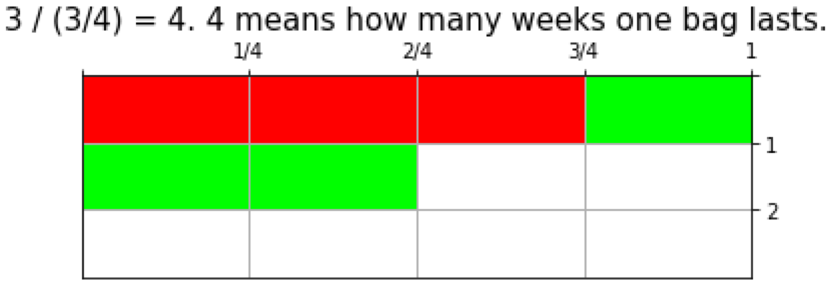


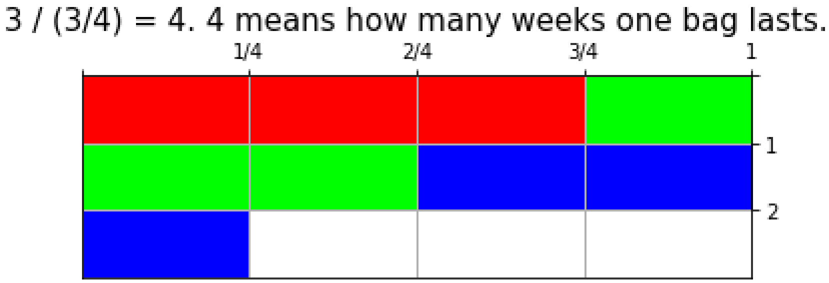
Second, we show 3 which means how many pounds of cat food we have .

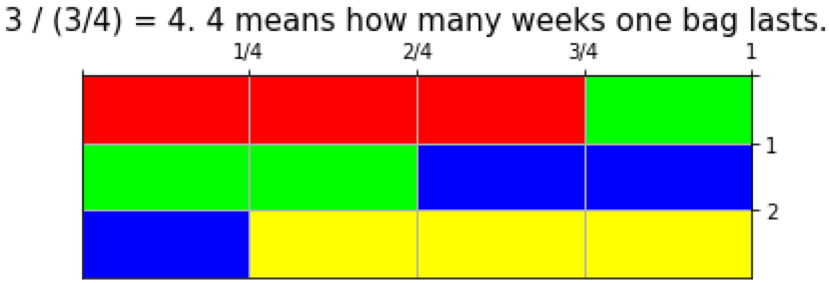


3 divided by 3/4.We want to know how many 3/4's can fit into 3.


3 / (3/4) = 4. 4 means how many weeks one bag lasts.
1/4
2/4
3/4
1
1
2


3 / (3/4) = 4. 4 means how many weeks one bag lasts.
1/4
2/4
3/4
1
1
2


3 / (3/4) = 4. 4 means how many weeks one bag lasts.
1/4
2/4
3/4
1
1
2


3 / (3/4) = 4. 4 means how many weeks one bag lasts.
1/4
2/4
3/4
1
1
2

Thank you :-)

```
Compressed video.mp4 into temp_video.mp4
```

Out[18]:

In [ ]: