

## Exploring Limits and Infinity with Music

One morning, Charlie was listening to birds sing outside their window. They tried to mimic the sounds and realized the sounds were not just the notes they were used to hearing in songs on the radio. Charlie started to wonder – how many possible sounds are there?

Microtones are notes between the standard notes we might hear on a piano. It turns out, there are an infinite number of microtones! They just can't all be heard by the human ear.

A musical scale you might be familiar with is: Do, Re, Mi, Fa, Sol, La, Ti, Do – this is known as the Tonic Sol-fa scale. Imagine hearing two notes, such as Do and Re in a musical scale. Now imagine playing a note in between those two notes, then playing a note between that note and Do, and repeating this over and over again.

Charlie decides to use computer code to create a list of numbers that continue to split in half – just like when you imagined playing a note directly between two previous notes. Explore Charlie's code and see what kinds of patterns you notice as you continue to split the numbers in half, and then by other amounts.

1. Copy this exact code into a Python console, such as [Google Colab](https://cscircles.cemc.uwaterloo.ca/console/) or <https://cscircles.cemc.uwaterloo.ca/console/>.

```
print ('Number of Splits \t Microtone')
for i in range (1,6):
    splitNumber = i
    microtone = 1/2**splitNumber
    print (splitNumber, '\t\t\t', microtone)
```

When you execute the code by running it, your output should look like this:

Number of Splits	Microtone
1	0.5
2	0.25
3	0.125
4	0.0625
5	0.03125

To avoid making an error in your code, look carefully to make sure your code is exactly as shown. Have a classmate look it over to be sure.

**Number  
Unit 1 Line Master 5b****Exploring Limits and Infinity  
with Music (cont'd)**

To split the tone in half, the note halfway between the two original notes would be played. After 1 split, it is 0.5 between the two notes – a semitone. If we continue splitting the notes, after 5 splits, we end up with the number 0.03125.

What if we continue splitting the notes 10 times?

Let's check!

2. Alter the code so that the range is (1,11), which means we will see 10 splits – up to but not including 11:

```
print ('Number of Splits \t Microtone')
for i in range (1,11):
    splitNumber = i
    microtone = 1/2**splitNumber
    print (splitNumber, '\t\t\t', microtone)
```

As we continue to split the notes, over and over again, we will eventually approach a **limit**, but will never reach it, since there are an infinite number of sounds.

If we add all the microtones, we can determine the limit of the sum of these microtones!

3. Alter your code to determine the sum of the microtones:

```
sum = 0
print ('Number of Splits \t Microtone')
for i in range (1,11):
    splitNumber = i
    microtone = 1/2**splitNumber
    print (splitNumber, '\t\t\t', microtone)
    sum = sum + microtone
print ('\n')
print ('Sum:', sum)
```

## Exploring Limits and Infinity with Music (cont'd)

Your output should look like this:

Number of Splits	Microtone
1	0.5
2	0.25
3	0.125
4	0.0625
5	0.03125
6	0.015625
7	0.0078125
8	0.00390625
9	0.001953125
10	0.0009765625
Sum: 0.9990234375	

What number does the sum of the microtone values seem to be approaching?

**Hint:** Round up! This is the limit of the sum of the numbers in this case.

Let's have the code round the number for us.

4. Alter the code as shown below:

```
sum = 0
print ('Number of Splits \t Microtone')
for i in range (1,11):
    splitNumber = i
    microtone = 1/2**splitNumber
    print (splitNumber, '\t\t\t', microtone)
    sum = sum + microtone
print ('\n')
print ('Sum:', sum)
print ('Limit of the sum:', round(sum,2))
```

**Note:** `round(sum,2)` will round the sum to two decimal places.

**Number  
Unit 1 Line Master 5d****Exploring Limits and Infinity  
with Music (cont'd)**

Your output should look like this:

Number of Splits	Microtone
1	0.5
2	0.25
3	0.125
4	0.0625
5	0.03125
6	0.015625
7	0.0078125
8	0.00390625
9	0.001953125
10	0.0009765625

Sum: 0.9990234375  
Limit of the sum: 1.0

Instead of splitting the notes in half each time, let's split them in thirds!  
Do you think the sum of the numbers will be higher or lower than when we split the notes in half? Let's try it...

5. Alter the code as shown below, then run it.

```
sum = 0
print ('Number of Splits \t Microtone')
for i in range (1,11):
    splitNumber = i
    microtone = 1/3**splitNumber
    print (splitNumber, '\t\t\t', microtone)
    sum = sum + microtone
print ('\n')
print ('Sum:', sum)
print ('Limit of the sum:', round(sum,2))
```

What is the limit of the sum when we divide the tones by 3 each time?  
What would this number be in fraction form?

**Exploring Limits and Infinity  
with Music (cont'd)****Reflect and Connect**

6. Predict what the limit of the sum would be if we divided the notes by 4.

Alter the code to check your prediction!

How could this number be represented as a fraction?

7. Alter the code to split the notes by 5 and by 6.

8. Complete the table below using your code and the output for the limit of the sums when we divide the tones by various values:

<b>Splitting Value (i.e., Number to Divide By)</b>	<b>Actual Sum</b>	<b>Limit of Sum Rounded to Two Decimal Places</b>	<b>Limit of Sum in Fraction Form</b>
2			$\frac{1}{1}$
3	0.49999153245609573		
4			$\frac{1}{3}$
5		0.25	
6	0.19999999669236565		$\frac{1}{5}$

9. What pattern do you notice in the limits of the sum as we increase the splitting value by one each time?