

# Unit 1 - Module 1

---

## Meaning of Sound

---

We first need to understand underlying concepts behind sound. As stated on page 22 of your *Let's Play the Ukulele* textbook:

When a rock falls into water, the water gets pushed and pulled around, creating little waves that spread out. This also happens in air, which is how we hear sound. When something like an ukulele string moves (vibrates), it squishes and stretches the air around it. This squishing and stretching of air then spreads arounds as a *sound wave*. When the air squishes and stretches quickly and the wave reach our ears just right, we call this *sound*.

The *frequency* of a sound describes how often the air squishes and stretches in one second, and its unit is called a hertz (Hz). Hypothetically, if you could hear a sound of 1 Hz, then your ears are noticing that the air is squished and stretched 1 time each second. If you could hear 9 kHz (kilohertz), then your ears are noticing that the air is squished and stretched 9,000 times per second. In music, some specific frequencies are called notes, and we give the names like C, B flat, and F sharp.

You might have heard people talk about sounds that are "high pitch" or "low pitch". Like *frequency*, *pitch* describes how many times a *sound wave* is squished and stretched in one second (measured in Hz), which is why we hear different "high" and "low" sounds. We can say that *frequency* describes what the wave is physically doing, and *pitch* is how this *frequency* feels to our ears. In this book, we use the scientific term *frequency* to talk about the *sound wave*.

## Tasks

---

You will work on multiple activities from your textbook, but will use the Jupyter notebook to document and submit your answers. Some tasks will be modified to introduce new programming concepts.

The title of your Jupyter notebook (as indicated by the # symbol for a first level heading) should be Unit 1 - Module 1. Your answers should be under ## Task 1 . This is the structure of your notebook:

## # Unit 1 – Module 1

Some descriptiong here maybe

### ## Task 1

Your answers and calculations.

### ## Task 2

Your answers and calculations.

## Task 1

---

Complete **Activity 1.2 - Octaves I** in your textbook. The fundamental concept introduced here is that of an **octave**. "When the frequencies are multiples of each other by a power of two, we call them *octaves*. For example, 3 Hz and 24 Hz are octaves, because you multiply 3 Hz by a power of two ( $8 = 2^3$ ) to get 24 Hz. 3 Hz and 25 Hz are not octaves."

In python, you can write exponents using two multiplication symbols.

```
print(2**3) # should output 8
print(4**-1) # should output 0.25
```

## Task 2

---

You will complete a modified version of **Activity 1.3 - Octaves II** for your second task. Instead of just calculation the three frequencies, you are going to write a program that let's you input 2 numbers, the frequency and the number of octaves (higher or lower) that you want to calculate.

For example, your program should ask me to enter a frequency and the number of octaves, so I enter 200 and then 1, and the computer should tell me that the frequency that is 1 octave higher than 200 Hz is 400 Hz.

You already know how to calculate this and how to print to the screen, but you don't how to get input from the user. To get an input from the user, you can simply use the built-in `input()` function.

For example, you can ask someone to insert a number and assign that input to the variable a:

```
a = input("Enter a number: ")    # get input
print(a)                        # prints value of a
print(type(a))                  # prints a's data type
```

What do you notice? Are you getting a number (int or float) or something else?

You can convert strings to numbers (assuming that you've only entered numeric characters of course) using the built-in `int()` function or the `float()` function, depending on what you want to convert to.

For example:

```
a = input(Please enter an integer: )
myInt = int(a)
print(myInt)
print(type(myInt))
```

What happens when you input a `-` sign. Does it still work?

Can you do the same for floats?

Here is a recommended pseudocode to complete your task:

```
# ask for an input for the frequency
# convert the input to a float
# ask for an input for the number of octaves
# convert the input to an int
# calculate the frequency
# print the result
```

You should use string concatenation to print your result nicely. In other words, build one single result string that prints your answer. You can combine strings using the `+` operator. To convert ints or floats to strings, use the `str()` function.

For example:

```
a = 14
b = 2
c = 14/2
resultOutput = "When I divided " + str(14) + " by " + str(b) + ", I get " + str(c) + "
print(resultOutput)
```

## Extra

---

If you finish early, take a look at some of the other string operators, such as

- `s.upper()`
- `s.lower()`
- `s.capitalize()`
- `len(s)`
- `s.strip()`
- `s.replace(s2, s3)`
- `s.isnumeric()`
- `s[0]`
- `s[1]`
- `s[:]`
- `s[::-1]`

In the given examples, replace `s` with the name of your string variable. You can view many more useful functions in this [Python string cheat sheet](#). Don't try to remember them all, just understand how to use them.