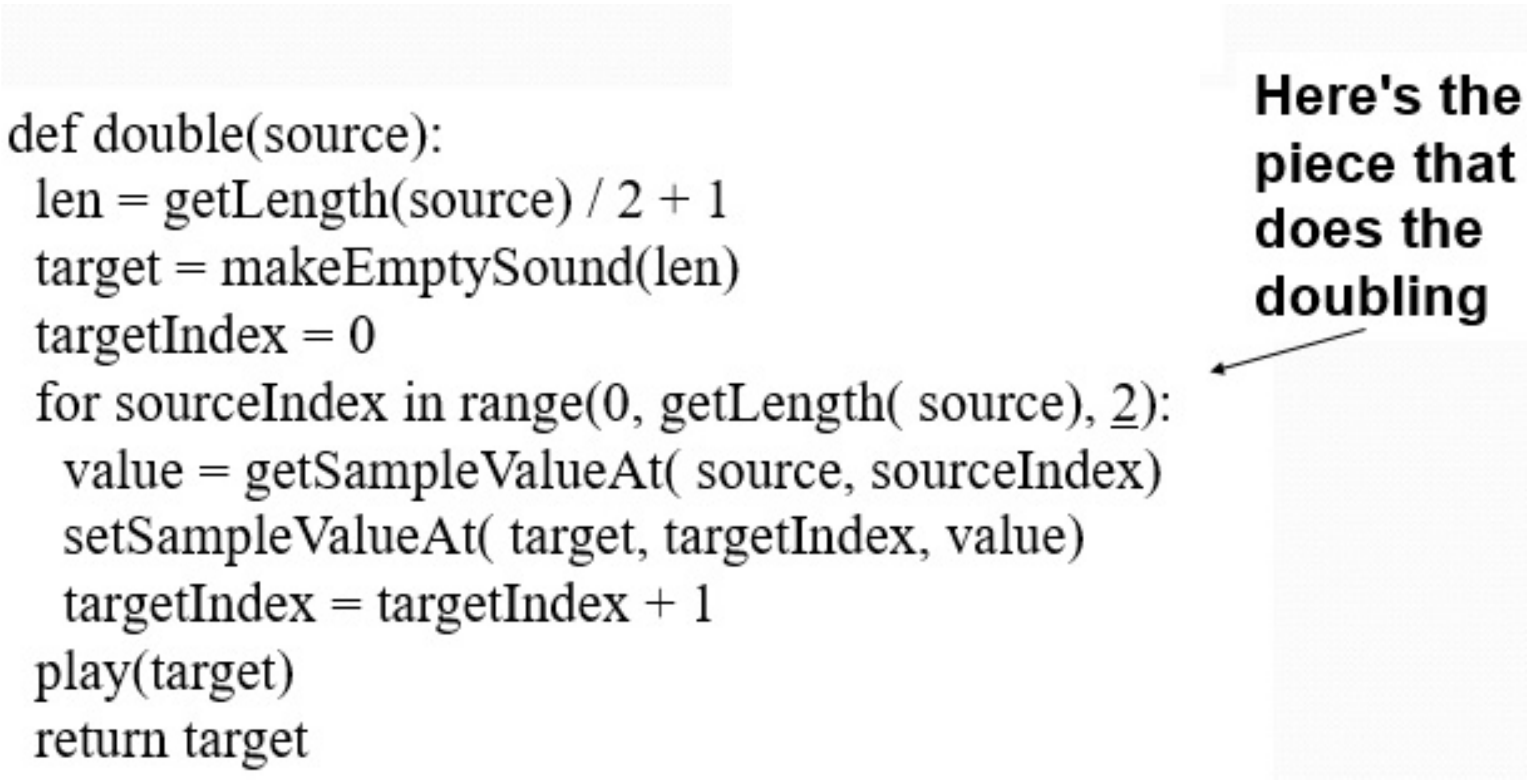
How Sampling Keyboards Work

- They have a huge memory with recordings of lots of different instruments played at different notes
- When you press a key on the keyboard, the recording closest to the note you just pressed is selected, and then the recording is shifted to exactly the note you requested.
- The shifting is a generalization of the half/double functions we saw earlier.

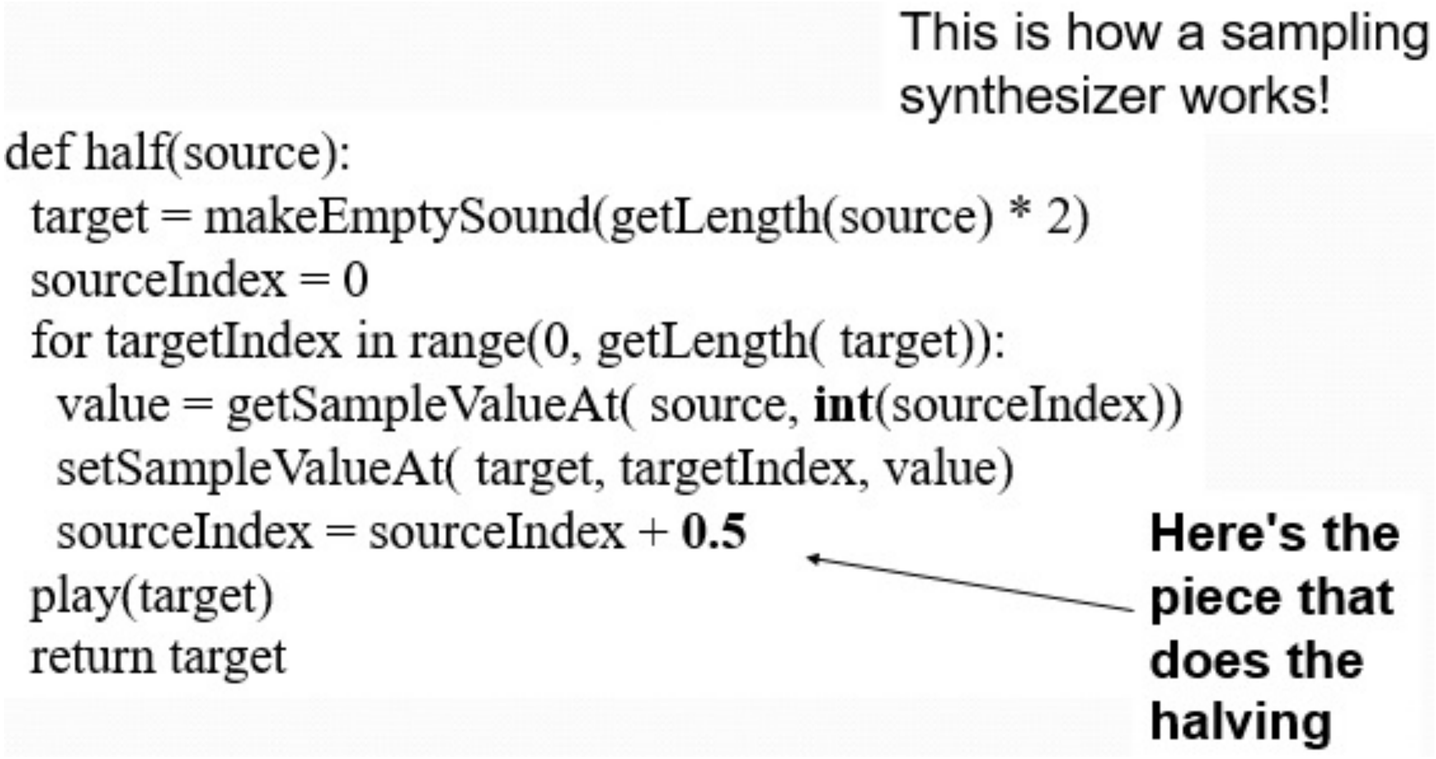
Doubling the Frequency



```
def double(source):
  len = getLength(source) / 2 + 1
  target = makeEmptySound(len)
  targetIndex = 0
  for sourceIndex in range(0, getLength( source), 2):
    value = getSampleValueAt( source, sourceIndex)
    setSampleValueAt( target, targetIndex, value)
    targetIndex = targetIndex + 1
  play(target)
  return target
```

Here's the piece that does the doubling

Halving the Frequency

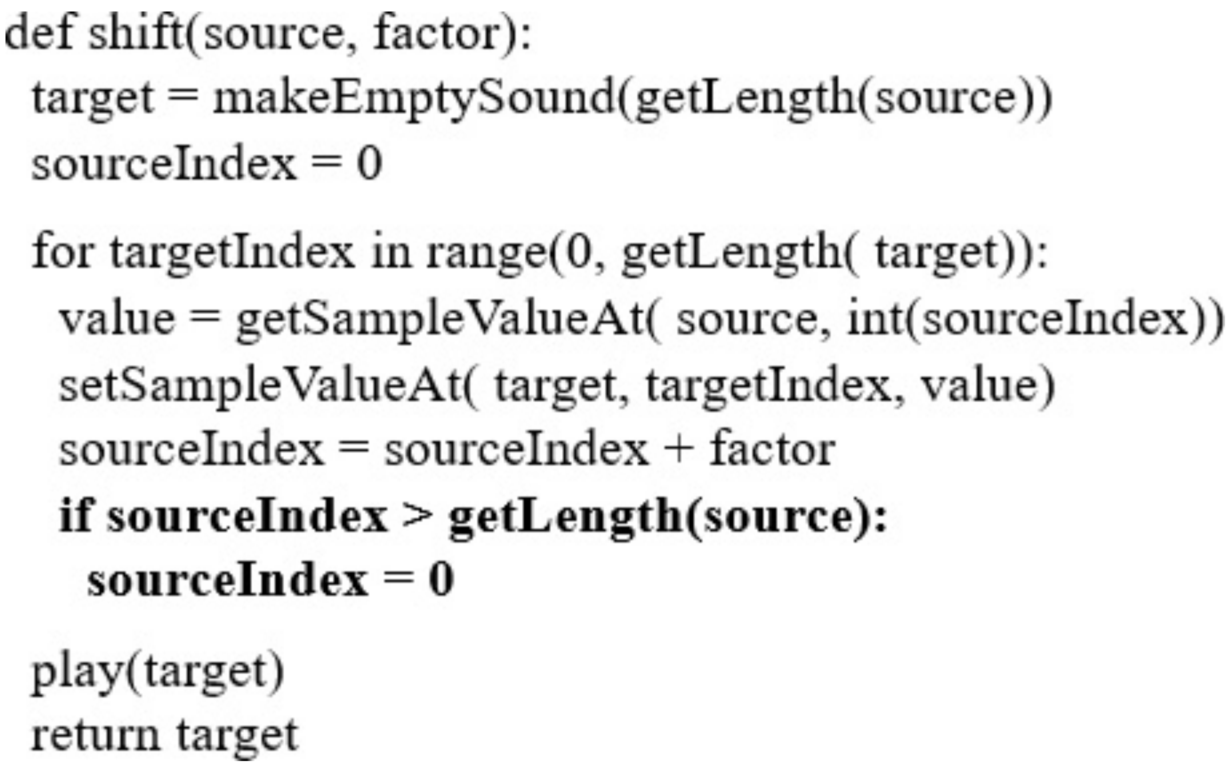


This is how a sampling synthesizer works!

```
def half(source):
  target = makeEmptySound(getLength(source) * 2)
  sourceIndex = 0
  for targetIndex in range(0, getLength( target)):
    value = getSampleValueAt( source, int(sourceIndex))
    setSampleValueAt( target, targetIndex, value)
    sourceIndex = sourceIndex + 0.5
  play(target)
  return target
```

Here's the piece that does the halving

We Need to Prevent Going Past the End of the Sound



```
def shift(source, factor):
  target = makeEmptySound(getLength(source))
  sourceIndex = 0

  for targetIndex in range(0, getLength( target)):
    value = getSampleValueAt( source, int(sourceIndex))
    setSampleValueAt( target, targetIndex, value)
    sourceIndex = sourceIndex + factor
    if sourceIndex > getLength(source):
      sourceIndex = 0

  play(target)
  return target
```

Now We have the Basics of a Sampling Synthesizer

For a desired frequency *f* we want a **sampling interval** like this:

$$samplingInterval = \left(size\left(sound\right)\right)\frac{f}{sampling - rate}$$

How the Original Sound Synthesizers Worked

- What if we added pure sine waves?
  - We can generate a sound that is just a single tone (see the book)
  - We can then add them together (perhaps manipulating their volume) to create sounds that don't exist in nature
- Don't have to use just sine waves
  - Waves that are square or triangular (seriously!) can be heard and have interesting dynamics
  - We can add together waves of lots of types to create unique sounds that can't be created by physical instruments
- We call this additive synthesis
  - Additive synthesis as-is isn't used much anymore

Sampling as an Algorithm

- Think about the similarities between:
  - Halving the sound’s frequency and scaling a picture larger.
  - Doubling the sound’s frequency and scaling a picture smaller.