

WRITING CUSTOM PYTHON SCRIPTS IN OMNIVIEW PART I

Recently our support team wrote up a helpful tip on indexing the array of absorbance spectrum to get the desired wavelength in a custom python script which is executed by [OmniView Software](#). For information on overwriting negative answers see [Writing Custom Python Scripts in OmniView part II](#).

As shown in the image below, each channel on the spectrometer can be individually configured with a starting and stopping wavelength. Additionally, the step size can be adjusted.

If we assume that the starting wavelength will be 1000 nm and the ending wavelength will be 2100 nm with a 1 nm step size, then we could hard code in values.

For example, if we want to get the absorbance value at 1430nm then we could simply call

```
au = getAu(scan)

absorbance1430 = au[1430]

absorbance1450 = au[1450]

answer = absorbance1430 / absorbance1450
```

However, this will only work as long as the channel's configuration does not change. If in the future someone adjusts the starting position or the step size then the position inside of the array will change and the above code will not work. To properly determine the index at which to find absorbance at a particular wavelength, e.g. 1430nm, the above code would have to calculate the index as follows:

```
au = getAu(scan)

wl = getWL(scan)

step = wl[1] - wl[0]

index1430 = float(1430 - wl[0])/step
```

```
absorbance1430 = au[index1430]
```

(And this code would need to be repeated for the absorbance at 1450.)

A simpler alternative is to use the `getAuAt` function which achieves the same thing as the above code in a single line:

```
answer = getAuAt(scan, 1430) / getAuAt(scan, 1450)
```

Interested in learning more? Check out the training videos we have on our [youtube channel](#).