You can access an element from a tuple the same way as accessing an element from a list, which is by providing the index of the element using square brackets. Here is an example:

```
C:\Users\Paul McJannet>python3
Python 3.12.7 (tags/v3.12.7:0b05ead, Oct  1 2024, 03:06:41) [MSC v.19
41 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information
.
>>> timmy = ("Tim", "Stutzle", 18)
>>> print(timmy[2])
18
```

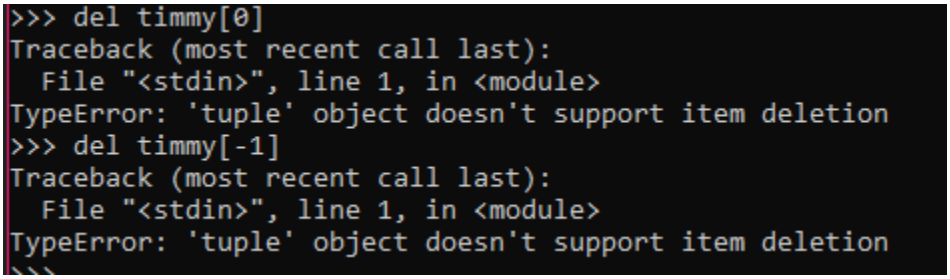
Since indexing is zero-based, this means that we expect timmy[0] to be “Tim”, timmy[1] to be “Stutzle” and timmy[2] to be 18 (which it is).

Tuples also allow for negative indexing the same as lists do. Using an index of [-1] means grabbing the very last element in the tuple, [-2] grabs the penultimate element, [-3] retrieves the third last element, etc. This example gets timmy[-2], and we should expect to get the string “Stutzle” back:

```
>>> timmy = ("Tim", "Stutzle", 18)
>>> print(timmy[2])
18
>>> print(timmy[-2])
Stutzle
```

Unlike indexing, tuples are not the same as lists when it comes to mutability, meaning we cannot delete an element from a tuple. “Tuples are unchangeable, meaning that we cannot change, add or remove items after the tuple has been created” (W3Schools, 2024, para. 4).

This can be seen when trying to delete any element in “timmy”. The result is always an error that reads: “**TypeError: 'tuple' object doesn't support item deletion**”. See below:



W3Schools. (2024). *Python Tuples.* W3Schools. https://www.w3schools.com/python/python_tuples.asp