

Adding 8-bit binary numbers | Part B

Sometimes you might need to only use part of a string, so you could use the slicing operator to take out the part you need. Supposing you have a database of vehicle number plate registrations like this:

```
AB13 CDE
```

The number tells us how old the car is, so we want to take out only this part.

```
numberplate = "AB13 CDE"  
age = numberplate[2:4]  
print (age)
```

The slicing operator works like this:

```
<string>[ start : end ]
```

We plug in the string and then specify the start and end of the part we want. But hold on a minute, 1 is the third character – why have we started at 2? The slices actually work like this:

```
|0 A |1 B |2 1 |3 3 |4   |5 C |6 D |7 E |8
```

The lines represent where the slices will take place. Of course as we already know, computers begin counting at 0 so the first line is 0. Can you see that the line with a number 2 is just before the character 1, and the line with a number 4 is just after the character 3? This is why we started at 2 and not 3.

We can also miss out either the start or the end. If we miss out the start value, the string will be sliced from the start up until the specified end point:

```
food = "strawberries and cream"  
print( food[ :5] )
```

This outputs "straw"

Or, if we want to slice from a specified point to the end, we miss out the end value:

```
print( food[5:] )
```

This outputs "berries and cream".

