

Y9 Python Help Sheet – Working with Lists

A list is created in Python using square brackets []. It is very similar to an array that you will encounter in other languages.

Lists are a good way of storing lots of related values using just one variable name

A typical list might be:

```
classlist = ["Joe", "Ann", "Sam", "Fred", "Mary"]
```

Like strings they are stored in positions starting at 0 (zero).

classlist	0	1	2	3	4
	Joe	Ann	Sam	Fred	Mary

Selecting single items

If I entered:

```
print(classlist[2])
```

it would print `Sam` as it is the item in position 2.

Selecting groups of items

You can select groups by using the colon : but there are a couple of things to remember. If you entered

```
print(classlist[0:2])
```

it will print `"Joe", "Ann"` , the items in positions 0 to 2 **but not including position 2**.

```
print(classlist[2:4])
```

would print `"Fred", "Mary"` the items from 2 to 4 but **not including position 4**.

Selecting items from the end of the list

If you use a negative number in the square brackets it will return items counting from the end of the list so

```
print(classlist[-2])
```

would print `Fred` as it is the second item from the end of the list

Selecting characters up to or after a certain point

If you wanted to print everything after a certain position you can do this by choosing a position and then a colon so

```
print(classlist[2:])
```

would print `"Sam", "Fred", "Mary"` as that is what is in position 2 onwards

If you wanted to print everything up to a certain position you put the colon first so

```
print(classlist[:2])
```

would print "Joe", "Ann" as they are the items up to but **not including position 2**.

Selecting characters from the list

You can select parts of each item by using another set of square brackets.

```
print(classlist[2][2])
```

would print `m` because it is character position 2 in list item number 2 (Sam)

```
print(classlist[3][2:4])
```

would print `ed` because it is character positions 2 to 4 (but not including 4) in list item number 3 (Fred)

In Python, lists can contain strings, numbers or symbols or even a mixture of different data types.

Questions

Complete the table

```
shopping = ["Bread", "Milk", "Cheese", "Chocolate"]
```

Code	Output
<pre>print(shopping[2])</pre>	<code>Cheese</code>
<pre>print(shopping[1:3])</pre>	<code>Milk, Cheese</code>
<pre>print(shopping[0:2])</pre>	<code>Bread, Milk</code>
<pre>print(shopping[-2])</pre>	<code>Cheese</code>
<pre>print(shopping[-2:])</pre>	<code>Cheese, Chocolate</code>
<pre>print(shopping[:1])</pre>	<code>Bread</code>
<pre>print(shopping)</pre>	<code>(All items)</code>
<pre>print(shopping[3][5:])</pre>	<code>late</code>