

Further notes on *sprintf*

Electronic Music II

Spring 2013

1. Creating an open message for `sfplay~` by using `sprintf` is simpler than laid out in the previous handout.
2. `sprintf` can format its input, based on how the arguments are entered. This is demonstrated below using the following sequence of messages: “open/A/a”, “Daniel/Fr/antz”, and “All/One/Word”.
3. Create a `sprintf` with three `%s` arguments. Type them one after the other, with no spaces.

```
sprintf %s%s%s
```

4. The output of this object will accordingly have no spaces:

Object	Message
1	openAa
1	DanielFrantz
1	AllOneWord

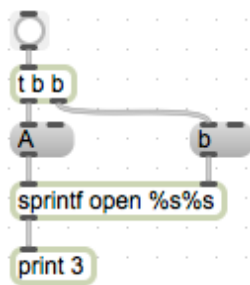
5. Now create another `sprintf`, again with three `%s` arguments. This time, add a space after the first argument (`%s %s%s`, shown below).

```
sprintf %s %s%s
```

6. This will add a space between the first and second symbols sent to `sprintf`, but not between the second and third.

2	open Aa
2	Daniel Frantz
2	All OneWord

7. Keep in mind that an input to the first inlet (representing the first slot in the output list) will output the entire list. Remember to only send this message once the other two inlets (slots) have been set.
8. `sprintf` can also add fixed data to its input, as long as it is created with the arguments. So, we can simplify how we make an open message even further:



9. Here I've added the word “open”, since this is a fixed element in all open messages we will be creating. `sprintf` will combine its output list with this text, as it is formatted in the object. So, the output of this is:

Object	Message
3	open Ab