# List Constructors
## When given two lists...
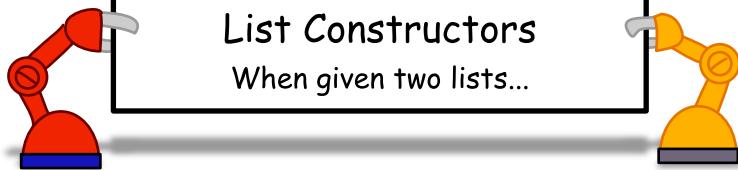
cons
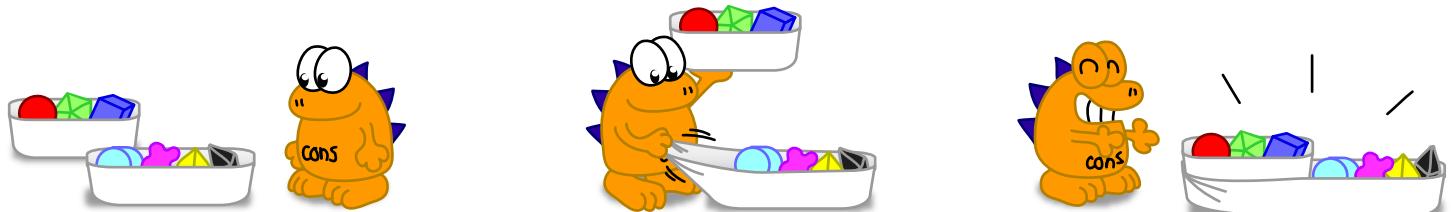

```
> (cons '(r g b) '(c m y k))
((r g b) c m y k)
```

append
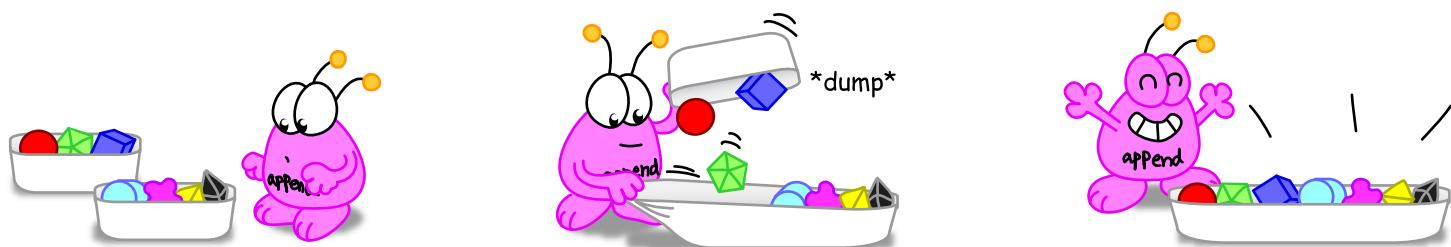

```
> (append '(r g b) '(c m y k))
(r g b c m y k)
```

list
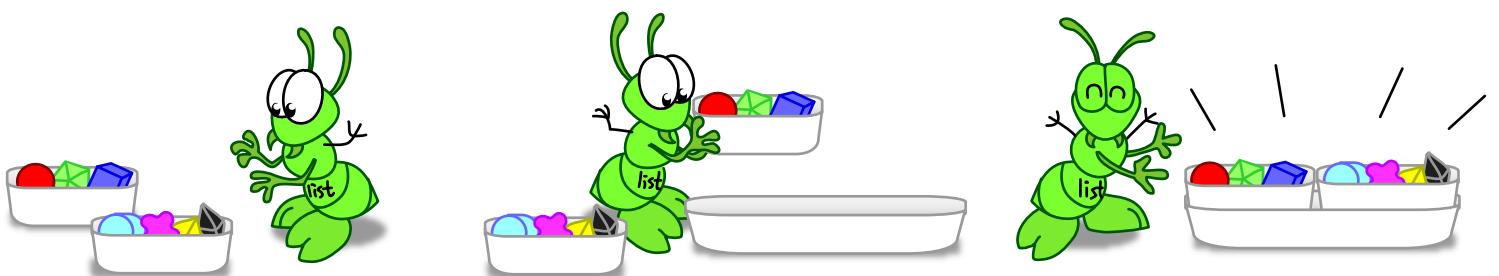

```
> (list '(r g b) '(c m y k))
((r g b) (c m y k))
```

Keep in mind that the list constructors don't change the input given to them. They make copies of the input and work with the copies.

```
> (arguments 'cons)
2
> (arguments 'append)
many
> (arguments 'list)
many
```