

# Tuples

Tuples are a data structure that allows you to collect together elements of different types. You can have any type you want, for example, if you wanted to represent a playing card you could create a pair tuple:

```
favCard :: (Char , String)
favCard = ('Q', "Clubs")
```

Tuples have a fixed size after creation and the constructor for a tuple is `(,)`. Having one comma will make you a pair tuple, adding more commas will create bigger tuples with more slots for all your favourite data types (but the max tuple size is 62):

```
(,)      :: a -> b -> (a, b)
(,,)     :: a -> b -> c -> (a, b, c)
(,,,)    :: a -> b -> c -> d -> (a, b, c, d)
```

The type of the pair tuple constructor `(,)` says “yo gimme an `a`, and a `b` and I’ll bung them in a tuple for you”. It is like ordering lots of presents for yourself from the same online shop and having them sent to you in the one parcel to save on postage. For bigger tuples, the alphabet is used to enumerate subsequent slots you wish to have in your tuple. What happens when the alphabet runs out though?! Don’t worry you aren’t limited to 26 slots in a tuple, Haskell has an inspired naming system for slots after slot `z`. If you’re curious just ask GHCi for the type of `(,,,,,,,,,,,,,,,,)`.