What are some other terms for *disjoint union*?

Tagged union, *variant records*, *algebraic data types*.

What kinds of unions does OCaml support?

exact unions which do not admit new members after initial definition, and *open unions*, which do.

Give the syntax for exact unions.

```
type name =
    name1 [of type1]
  ...
  | nameN [of typeN]
```

Each `namei` is the *constructor* names and must begin with a capital letter.
The optional `typei` is its value.

How are instances of a disjoint union created?

With the syntax of function application give the constructor name followed by an instance of its value type, if applicable.

Give the syntax of open unions.

```
type 'a name =
  [> 'name1 [of type1]
  ...
  | 'nameN [of typeN] ] as 'a
```

What does the type `[> X of Y]` mean?

It means the expression is an instance of a union type that contains the member `X` with value `Y`. More cases that are supported can be added using `|`.

The value of a union instance can be accessed through ...

... pattern matching.

Give the effective definition of `list`.

Why isn't this the actual definition.

```
type 'a list =
    []
  | :: of 'a * 'a list
```

`[]` is not a valid name and requires language support.

Give the definition of `option`.

```
type 'a option =
    None
  | Some of 'a
```

Give another syntax for closed unions.

Same as open unions, except with $<$ instead of $>$.