The `append` Predicate

```
?- append([1,2],[3,4],Z).

Z = [1, 2, 3, 4]

Yes
```

- Library predicate **`append(X,Y,Z)`** succeeds if and only if **`Z`** is the result of appending the list **`Y`** onto the end of the list **`X`**

Not Just A Function

```
?- append(X,[3,4],[1,2,3,4]).

X = [1, 2]

Yes
```

- **append** is **flexible**:
- can be used with any pattern of instantiation (that is, with variables in any positions)

Not Just A Function

```
?- append(X,Y,[1,2,3]).

X = []
Y = [1, 2, 3] ;

X = [1]
Y = [2, 3] ;

X = [1, 2]
Y = [3] ;

X = [1, 2, 3]
Y = [] ;

No
```

An Implementation

```
append([], B, B).
append([Head|TailA], B, [Head|TailC]) :-
  append(TailA, B, TailC).
```

Other Predefined List Predicates

Predicate	Description
`member(X,Y)`	Provable if the list **`Y`** contains the element **`X`**.
`select(X,Y,Z)`	Provable if the list **`Y`** contains the element **`X`**, and **`Z`** is the same as **`Y`** but with one instance of **`X`** removed.
`nth0(X,Y,Z)`	Provable if **`X`** is an integer, **`Y`** is a list, and **`Z`** is the **`X`**th element of **`Y`**, counting from 0.
`length(X,Y)`	Provable if **`X`** is a list of length **`Y`**.

- All flexible, like **`append`**
- Queries can contain variables anywhere