

Example : 5 List

In Prolog list is an ordered sequence of components, where components can be variables, constants, compound terms, numbers or another list.

- **Syntax :** `[components]` - write your components within square brackets.
- **Examples :**

`[1,2,3,4].`
`[nova, james, 2].`
`[X, Y, james].`
`[A, [p, n, c], 4].`
`[likes(X,Y), cat(fur)].`
`[]` - an empty list

- **Head and Tail of the lists :**

head : the 1st element of the list.
tail : all elements of the list except the 1st one
Syntax : `[H | T]`

For example :

- (1) In list `[1,2,3,4]` head is 1 and tail is `[2,3,4]`
- (2) In list `[a]` head is a and tail is `[]`.
- (3) In list `[likes(john, mary), X, 1, 2]` head is `likes(john, mary)` and tail is `[X, 1, 2]`.
- (4) In list `[A, [p, n, c], 4]` head is A and tail is `[[p, n, c],4]`
- (5) In list `[[1, 2], a, b]` head is `[1, 2]` and tail is `[a, b]`

Note :

In above examples tails are always a list.

`[H | T] = [1, 2, 3, 4]`, H is 1 and T is `[2, 3, 4] = [1 | [2, 3, 4]]`

Valid:

`[1 | [2, 3]]`
`[[1,2] | [3, 4]]`

Invalid:

`[1, 2 | 3]`

Examples on next page.

Examples - How do these terms unify ? - Read [Unification Instantiation](#)

(1) $[A \mid [b, c]] = [X \mid Y]$

- Prolog Responses :

```
?- [X|Y] = [A | [b, c]] .
```

```
X = A,
```

```
Y = [b, c].
```

```
?- [A | [b, c]] = [X|Y].
```

```
A = X,
```

```
Y = [b, c].
```

(2) $[B, [p, n, c], 4] = [X \mid Y]$.

- Prolog Response :

```
?- [B, [p, n, c], 4] = [X | Y].
```

```
B = X,
```

```
Y = [[p, n, c], 4].
```

(3) $[\text{likes}(\text{john}, \text{mary}), X, 1, 2] = [\text{Head} \mid \text{Tail}]$.

- Prolog Response :

```
?- [likes(john, mary), X, 1, 2] = [Head | Tail].
```

```
Head = likes(john, mary),
```

```
Tail = [X, 1, 2].
```

(4) Try it yourself. $[[1, 2], a, b] = [\text{Head} \mid \text{Tail}]$.

(5) Try it yourself. $[\text{john}] = [\text{Head} \mid \text{Tail}]$.

Sample Prolog Program using List :

Program :

```
/* Facts */
```

```
list([p, q, r]).
```

```
/* Rules */
```

```
what_is([Head | Tail]):- list([Head|Tail]).
```

Query Prompt :

```
?- what_is([Head|Tail]).
```

```
Head = p,
```

```
Tail = [q, r].
```

Learn More - http://boklm.eu/prolog/page_7.html