```
match expr with
 | pat1 [when cond1] -> expr1
 ...
 | patN [when condN] -> exprN
```

The initial `|` is not required.

```
let funcname param = function
 | pat1 [when cond1] -> expr1
 ...
 | patN [when condN] -> exprN
```

With `|`, being boolean *or*.

Use as.

```
pat as name -> expr
```

```
start .. end -> expr
```

The pattern is matched if the value is in the range.

```
raise (Invalid_argument "message")
```

- In `let` bindings.
- In `fun` expressions.