

What is the syntax of `match` expression?

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

The initial | is not required.

What shortcut is provided for functions whose body is a pattern match on its sole parameter?

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

How can two patterns be composed?

With $|$, being boolean *or*.

Give the value matching a pattern a name.

Use as.

pat as name -> expr

Give the syntax of a pattern range.

`start .. end -> expr`

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

Do the equivalent of throwing an
`IllegalArgumentException`.

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

Where is pattern matching used other than in
`match` expressions?

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