

# Pattern matching

**Tristan Crolard**

Department of Computer Science  
CEDRIC lab / SYS team

**`tristan.crolard@cnam.fr`**

**`cedric.cnam.fr/sys/crolard`**

# Pattern matching on lists – `is_empty`

## Case analysis:

- ▶ A list is either empty or not.

```
>>> def is_empty[A](l: list[A]) -> bool:
      match (l):
        case []:
          return True
        case _:
          return False
```

```
>>> is_empty([])
```

True

```
>>> is_empty([1, 2, 3])
```

False

# Pattern matching on lists – hd & tl

## Case analysis:

- ▶ A list is either empty or not.
- ▶ If it is not empty, it is composed of some **head**  $h$  (the first element) and some **tail**  $t$  (the remaining elements)

```
>>> def hd[A](l: list[A]) -> A:  
      match (l):  
        case [h, *t]:  
          return h  
        case _:  
          raise ValueError
```

```
>>> hd([1, 2, 3])
```

1

```
>>> hd([])
```

```
Traceback (most recent call last):
```

```
...
```

```
ValueError
```

```
>>> def tl[A](l: list[A]) -> list[A]:  
    match (l):  
        case [h, *t]:  
            return t  
        case _:  
            raise ValueError
```

```
>>> tl([1, 2, 3])
```

```
[2, 3]
```

```
>>> tl([])
```

```
Traceback (most recent call last):
```

```
...
```

```
ValueError
```