







UTILISATION D'UN TYPE <> DANS UN SYSTÈME DE TYPE M

**LINK TO THE PATH**











```
type direction = | North | East | South | West
```

```
let label d = match d with  
| North -> "north"  
| East -> "east"  
| South -> "south"  
| West -> "west"
```

```
let _ = print_endline(label North);;  
let _ = print_endline(label East);;  
let _ = print_endline(label South);;  
let _ = print_endline(label West);;
```

MetaLangage (1970) trouve toi un acronyme original Machine Learning





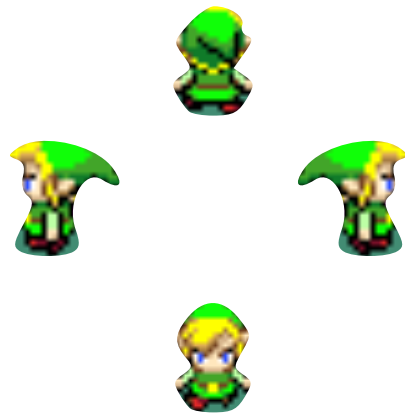
**oCaml**

MAGNET 2-DUALS - THOUGHTS

# LINK TO THE PATH

## UTILISATION D'UN TYPE « OU » DANS UN SYSTÈME DE TYPE ML

MetaLangage (1970) trouve toi un acronyme original Machine Learning



```
type direction = | North | East | South | West

let label d = match d with
| North -> "north"
| East -> "east"
| South -> "south"
| West -> "west"

let _ = print_endline(label North);;
let _ = print_endline(label East);;
let _ = print_endline(label South);;
let _ = print_endline(label West);;
```



# APARTÉ



## POURQUOI OCAML

French tech : made in INRIA

Qui l'utilise ? Facebook, Docker, Citrix, Tarides, Bloomberg, Janestreet, LexiFi, Tezos, ANSSI, Dassault Systèmes, Shiro Games, DGFIP, Airbus, Microsoft, ...

Très utilisé en finance, blockchain, infrastructure management, sécurité, compilateurs

Multi paradigme : fonctionnel, impératif, objet

Multi target : natif, byte code, js, unikernel ... php, java, Erlang

Un ML : très utile pour ce que je veux illustré et inspiration pour des langages objets moderne (Scala, Kotlin, Swift)

Vous en avez fait en M1

... Le meilleur langage du monde

... Les mauvaises langues diront qu'il manque de Higher Kinded Types ... et c'est vrai