

Type	
unit	void
int	integer
float	
bool	stdbool.h
char	'A'
string	"hello"
'a list	head :: tail [1;3;2]
'a array	[[5;7;8]]

N-uplet	
t1...tn	type inféré
let tuple = (a,b,c,d)	
let (a,b,c) = tuple	
fst / snd	

Reference, string, array	
let x=ref 3	variable mutable
x := 4 ;	affectation
!x	lecture
s.[0]	lecture string
t.(0)	lecture tab
t.(0) <- x;	écriture tab
s1 ^ s2	concaténation
! string = immuables	

Enregistrements
type record =
{ field1 : bool;
mutable field2 : int; }
let r = {field1= true ; field2 = 3;}
let r' = { r with field1=false }
r.field2 <-4;
let {field1=var_a;_}=r

Type somme
type enum =
cst
pair of string*int
let c = cstt
let c = Pair ("bar", 3)

Boucles	
while ... do ...done;	!creer variable mutable
for i=... to ... do ...done;	i inaccessible
for i= downto ... do ...done;	à l'envers

Exception	
raise Invalid_argument "blabla"	"blabla"= mess
raise Division_by_zero	
try ...	
with	
Division_by_zero ->	
Invalid_argument mess->	mess est utilisable

