

## TypeScript - enregistrements et variants

1. (type enregistrement). Considérez la déclaration de type suivante :

```
type Point = { readonly x: number, readonly y: number }
```

- Donner des exemples de valeurs de type Point.
- Ecrire une fonction qui déplace un point horizontalement :  
`function hTrans(p: Point, dx: number): Point`
- Ecrire une fonction qui déplace un point verticalement :  
`function vTrans(p: Point, dy: number): Point`
- Ecrire à nouveau les deux fonctions précédentes en utilisant la syntaxe *spread*.

2. Considérez les déclarations de type suivantes :

```
type Address = {  
  readonly streetNumber: number,  
  readonly streetName: string,  
  readonly postalCode: number,  
  readonly city: string  
}  
type Contact = {  
  readonly firstName: string,  
  readonly lastName: string,  
  readonly phone: string,  
  readonly address: Address  
}
```

- Donner des exemples de valeurs de type Address et Contact.
- Ecrire une fonction qui modifie la ville d'une adresse donnée :  
`function setCity(a: Address, c: string): Address`
- Ecrire une fonction qui modifie l'adresse d'un contact donné :  
`function setAddress(c: Contact, a: Address): Contact`
- Ecrire une fonction qui modifie le code postal de l'adresse d'un contact donné :  
`function setPostalCode(c: Contact, p: number): Contact`

3. (type énuméré). Considérez la déclaration de type suivante :

```
type Month =  
  | 'January'  
  | 'February'  
  | 'March'  
  | 'April'  
  | 'May'  
  | 'June'  
  | 'July'  
  | 'August'  
  | 'September'  
  | 'October'  
  | 'November'  
  | 'December'
```

Ecrire une fonction qui convertit un mois vers le numéro correspondant :

```
function monthToNumber(month: Month): number
```

4. Considérez les déclarations de type suivantes :

```
type State = { readonly count: number }  
  
type Action =  
  | 'reset'  
  | 'increment'  
  | 'decrement'
```

Ecrire une fonction `reducer` qui modifie un état donné en fonction d'une action donnée :

```
function reducer(state: State, action: Action): State
```

5. (**type variant**). Remplacer le type `Action` par le suivant et modifier la fonction `reducer` en conséquence :

```
type Action =  
  | { readonly type: 'reset' }  
  | { readonly type: 'increment' }  
  | { readonly type: 'decrement' }
```

6. Considérez les déclarations de type suivantes :

```
type State = { readonly balance: number }  
  
type Action =  
  | { readonly type: 'deposit', readonly payload: number }  
  | { readonly type: 'withdraw', readonly payload: number }
```

Ecrire une fonction `reducer` qui modifie un état donné en fonction d'une action donnée :

```
function reducer(state: State, action: Action): State
```