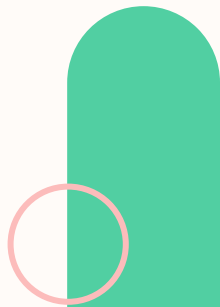


Full Stack Web Development

Exercise

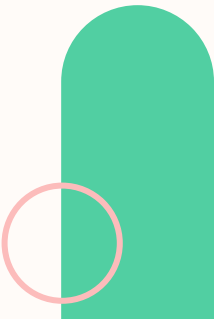
Exercise

- Create a function to check if two objects are equal
- Example
 - Input: `{ a: 2 } & { a: 1 }`
 - Output: **false**
- Example
 - Input: `{ a: "Hello" } & { a: 1 }`
 - Output: **false**
- Example
 - Input: `{ a: 1 } & { a: 1 }`
 - Output: **true**



Exercise

- Create a function to get the intersection of two objects
- Example
 - Input: `{ a: 1, b: 2 } & { a: 1, c: 3 }`
 - Output: `{ a: 1 }`



Exercise

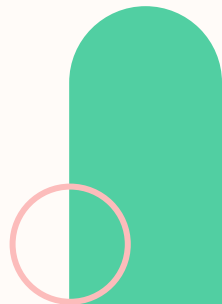
- Create a function to merge two array of student data and remove duplicate data
- Student data : name & email
- Example :

```
Array1 → [  
  { name: 'Student 1', email : 'student1@mail.com' },  
  { name: 'Student 2', email : 'student2@mail.com' }  
]
```

```
Array2 → [  
  { name: 'Student 1', email : 'student1@mail.com' },  
  { name: 'Student 3', email : 'student3@mail.com' }  
]
```

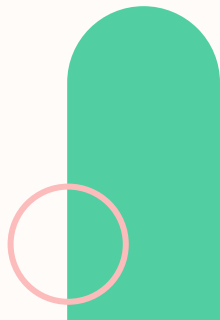
- Result :

```
ArrayResult → [  
  { name: 'Student 1', email : 'student1@mail.com' },  
  { name: 'Student 2', email : 'student2@mail.com' },  
  { name: 'Student 3', email : 'student3@mail.com' }  
]
```



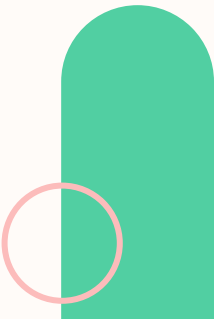
Exercise

- Create a function that can accept input as an array of objects and switch all values into property and property into value
- Example :
 - Input: `[{ name: 'David', age: 20 }]`
 - Output: `[{ David: 'name', 20: 'age' }]`



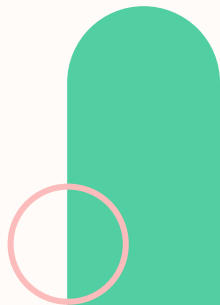
Exercise

- Create a function to find a factorial number using recursion
- Example
 - Input : 5
 - Output: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$



Exercise - Shooting Game

- **Specifications :**
 - Create a shooting game between two player
 - Each player has three properties : name, health and power
 - Each player will take turns to shooting
 - Before shooting, players get a chance to get random items (health +10 or power +10)
 - The game will continue until one of the players has health < 0
- **Requirements :**
 - Create **ShootingGame & Player** class
 - **ShootingGame** class :
 - **constructor(player1, player2)** → player objects as a parameter
 - **getRandomItem()** → return { **health: 0 or 10, power: 0 or 10** }
 - **start()** → start shooting games
 - **Player** class :
 - **Property** → **name, health (default 100), power (default 10)**
 - **hit(power)** → subtract player health
 - **useItem(item)** → apply item to player (increase health or power, based on result from **getRandomItem()**)
 - **showStatus()** → show player status (ex : *"Player A (Health => 100, Power => 10) "*)
 - ShootingGame **start()** function flow :
 - In every turn :
 - Show each player status before shooting
 - Get random item for each player before shooting
 - Show each player status after shooting
 - Show winner name



Thank You!

