

Solution 1 : Affichage **pizzaMax: { nom: 'cheese', prix: 16 }**

1. // Define the price calculation function once
2. const calculatePrice = (pizza) =>
3. pizza.ing.reduce((acc, ing) => acc + prices.get(ing), 0);
- 4.
5. const max = (pizzas) =>
6. pizzas.reduce((max, p) => {
7. const prix = calculatePrice(p)
8. return prix > max.prix ? { nom: p.name, prix } : max;
9. }, { nom: null, prix: 0 });
- 10.
11. console.log("pizzaMax:", max(pizzas));

**pizzaMax: { nom: 'cheese', prix: 16 }**

[code](#)

Solution 1-B : **Pointe vers la pizza**

**pizzaMax: { name: 'queen', ing: [ , , ,  ] }**

1. // Define the price calculation function once
2. const calculatePrice = (pizza) =>
3. pizza.ing.reduce((acc, ing) => acc + prices.get(ing), 0);
- 4.
5. const max = (pizzas) =>
6. pizzas.reduce((max, p) => {
7. const prix = calculatePrice(p)
8. return prix > max.prix ? { nom: p.name, prix } : max;
9. });

[code](#)

Solution 2 : Créer une nouvelle structure

1. `const pizzasWithPrices = pizzas.map(pizza => ({`
2. `name: pizza.name, // ou ...pizza`
3. `price: pizza.ing.reduce((acc, ing) => acc + prices.get(ing), 0)`
4. `});`
- 5.
6. `const max = pizzasWithPrices.reduce((acc, pizza) =>`
7. `acc.price > pizza.price ? acc : pizza`
8. `);`

Affichage

