



ITEM 4 구조적 타이핑에 익숙해지기

*TypeScript*의 타입 호환성은 구조적 서브타이핑을 기반으로 한다
구조적 타이핑이란 오직 멤버만으로 타입을 관계시키는 방법

타입스크립트의 구조적 타입시스템의 기본 규칙은 y 가 최소한 x 와 동일한 멤버를 가지고 있다면
 x 와 y 는 호환이 된다.

강한 타입 시스템을 통해 높은 가독성과 코드 품질을 지향하는 TS가 왜 타입 호환성을 지원할까?

타입 안정성에는 문제가 없을까?



- protein : 3(g)
- carbohydrates : 62(g)
- fat : 9 (g)

Food

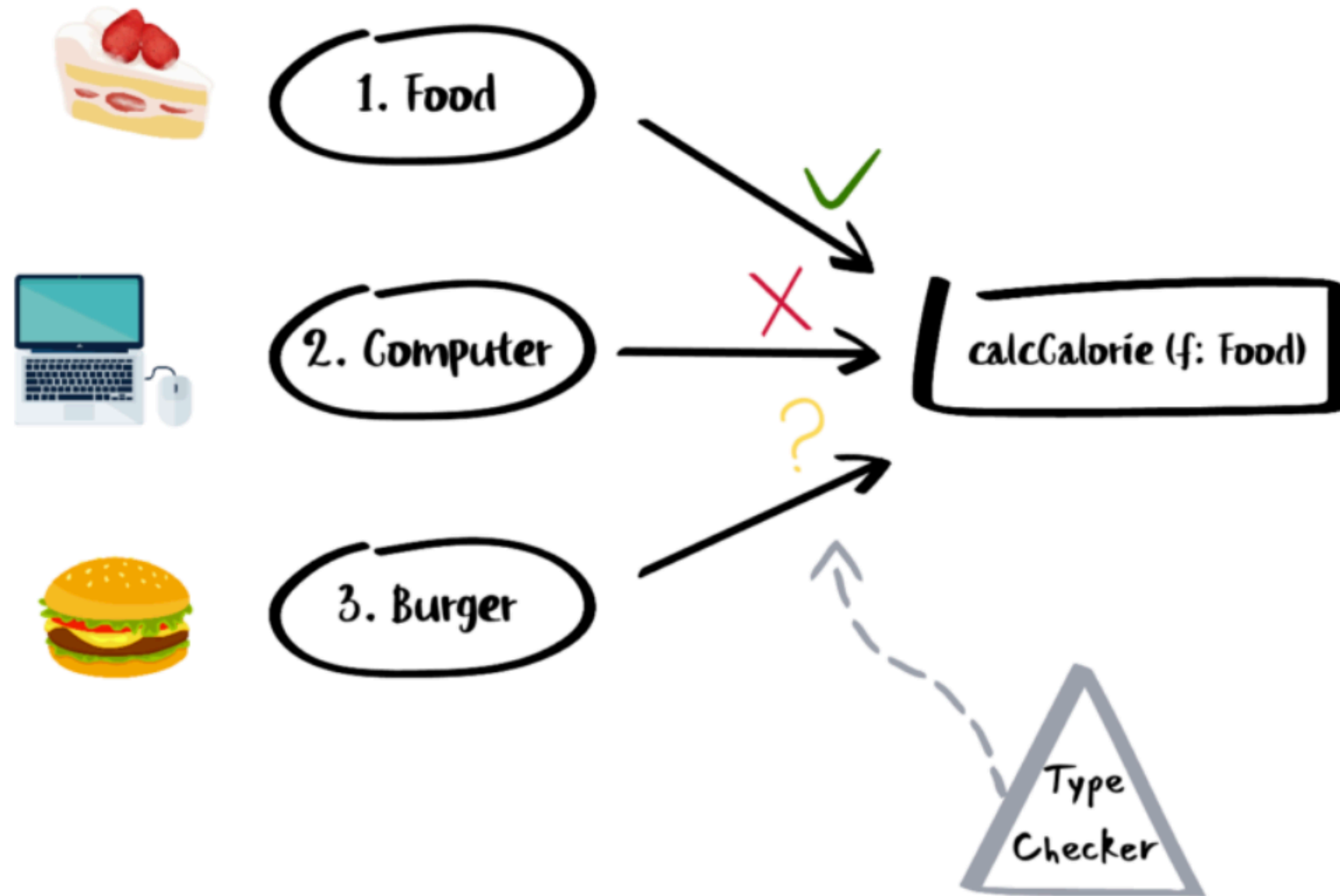


calcCalorie (f: Food)



calorie : 341 (kcal)

```
1  type Food = {  
2    /** 각 영양소에 대한 gram 중량값 */  
3    protein: number;  
4    carbohydrates: number;  
5    fat: number;  
6  }  
7  
8  function calculateCalorie(food: Food){  
9    return food.protein * 4  
10     + food.carbohydrates * 4  
11     + food.fat * 9  
12  }
```


```
type Food = {  
  /** 각 영양소에 대한 gram 중량값 */  
  protein: number;  
  carbohydrates: number;  
  fat: number;  
}  
  
function calculateCalorie(food: Food){  
  return food.protein * 4  
    + food.carbohydrates * 4  
    + food.fat * 9  
}
```

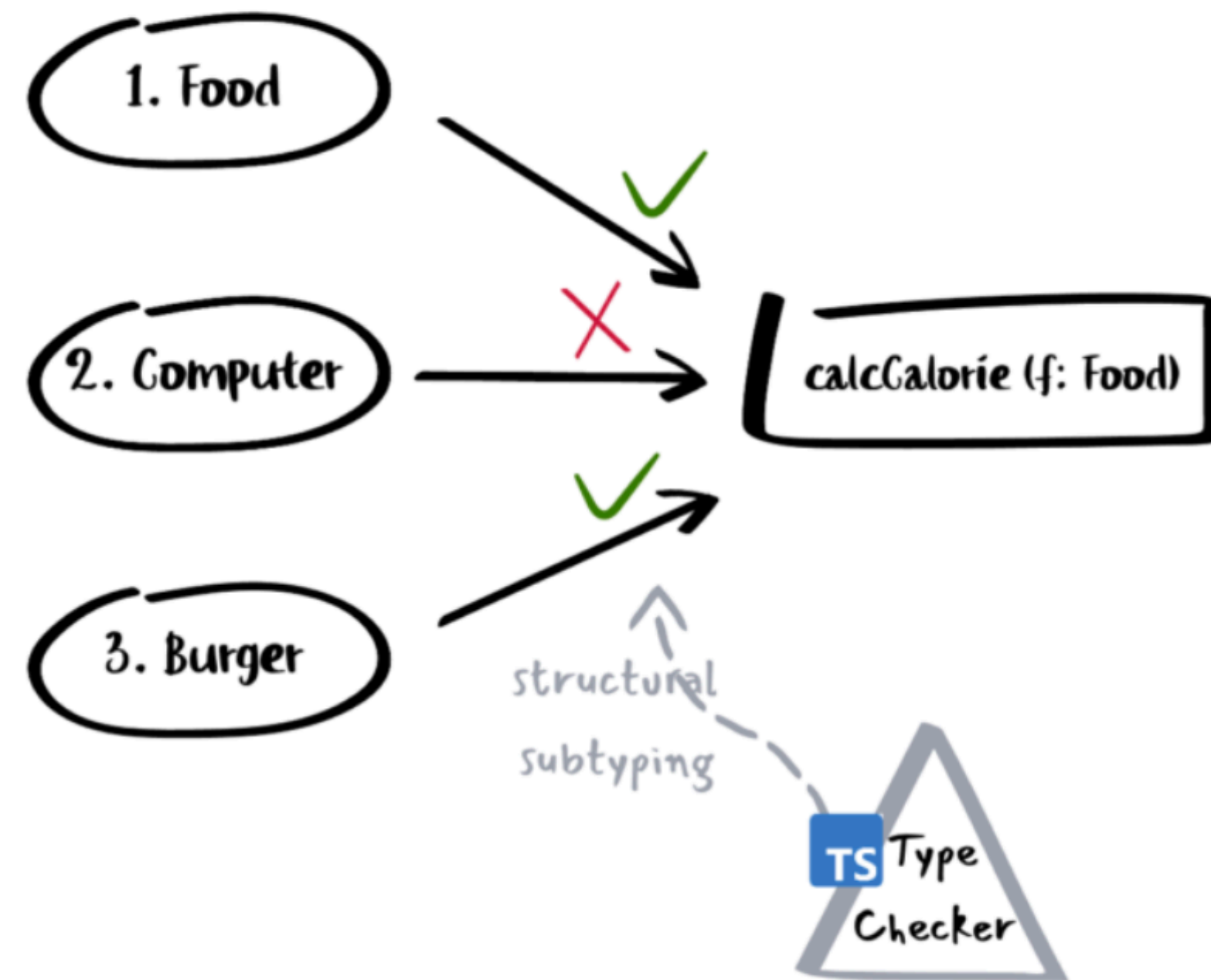
```
const burger = {  
  protein: 29,  
  carbohydrates: 48,  
  fat: 13,  
  burgerBrand: '버거킹'  
}
```


명목적 서브 타이핑

```
1  /** 상속 관계 명시 */
2  type Burger = Food & {
3      burgerBrand: string;
4  }
5
6  const burger: Burger = {
7      protein: 29,
8      carbohydrates: 48,
9      fat: 13,
10     burgerBrand: '버거킹'
11 }
12
13 const calorie = calculateCalorie(burger)
14 /** 타입검사결과 : 오류없음 (OK) */
15
```


구조적 서브타이핑

```
1  const burger = {  
2    protein: 29,  
3    carbohydrates: 48,  
4    fat: 13,  
5    burgerBrand: '버거킹'  
6  }  
7  
8  const calorie = calculateCalorie(burger)  
9  /** 타입검사결과 : 오류없음 (OK) */
```

source: <https://toss.tech/article/typescript-type-compatibility>