

Effective Typescript

해당 분야의 용어로 타입 이름 짓기

Item 36

컴퓨터 과학에서 어려운 일은 단 두 가지 뿐이다. 캐시 무효화와 이름 짓기

- 필 칼튼

```
interface Animal {  
  name: string;  
  endangered: boolean;  
  habitat: string;  
}  
  
const leopard: Animal = {  
  name: 'Snow Leopard',  
  endangered: false,  
  habitat: 'tundra',  
};
```

- name은 매우 일반적인 용어입니다.
동물의 학명인지 일반적인 명칭인지 알 수 없습니다.
- endangered 속성이 멸종 위기를 표현하기 위해 boolean 타입을 사용한 것이 이상합니다.
이미 멸종된 동물을 true로 해야 하는지 판단 할 수 없습니다.
- 서식지를 나타내는 habitat 속성은 너무 범위가 넓은 string 타입(Item 33)일 뿐만 아니라 서식지라는 뜻 자체도 불분명하기 때문에 다른 속성들보다도 훨씬 모호 합니다.
- 객체의 변수명이 leopard이지만, name 속성의 값은 'Snow Leopard'입니다.
객체의 이름과 속성의 name이 다른 의도로 사용된 것인지 불분명합니다.

속성에 대한 정보가 모호하기 때문에 해당 속성을 작성한 사람을 찾아서 의도를 물어봐야 합니다.

```

interface Animal {
  commonName: string;
  genus: string;
  species: string;
  status: ConservationStatus;
  climates: KoppenClimate[];
}
type ConservationStatus = 'EX' | 'EW' | 'CR' | 'EN' | 'VU' | 'NT' | 'LC';
type KoppenClimate = |
  'Af' | 'Am' | 'As' | 'Aw' |
  'BSh' | 'BSk' | 'BWh' | 'BWK' |
  'Cfa' | 'Cfb' | 'Cfc' | 'Csa' | 'Csb' | 'Csc' | 'Cwa' | 'Cwb' | 'Cwc' |
  'Dfa' | 'Dfb' | 'Dfc' | 'Dfd' |
  'Dsa' | 'Dsb' | 'Dsc' | 'Dwa' | 'Dwb' | 'Dwc' | 'Dwd' |
  'EF' | 'ET';
const snowLeopard: Animal = {
  commonName: 'Snow Leopard',
  genus: 'Panthera',
  species: 'Uncia',
  status: 'VU', // vulnerable
  climates: ['ET', 'EF', 'Dfd'], // alpine or subalpine
};

```

- name은 commonName, genus, species 등 더 구체적인 용어로 대체
- endangered는 동물 보호 등급에 대한 IUCN의 표준 분류 체계인 ConservationStatus 타입의 status로 변경
- habitat은 기후를 뜻하는 climates로 변경되었으며, 쾨펜 기후 분류를 사용

타입 정보를 찾기 위해 사람에 의존할 필요가 없습니다.

타입, 속성, 변수에 이름을 붙일 때 명심해야할 세 가지 규칙

1. 동일한 의미를 나타낼 때는 같은 용어를 사용해야 합니다.
2. 모호하고 의미 없는 이름은 피해야 합니다.
3. 이름을 지을 때는 포함된 내용이나 계산 방식이 아니라 데이터 자체가 무엇인지를 고려해야 합니다.

1. 동일한 의미를 나타낼 때는 같은 용어를 사용해야 합니다.

하루에 필요한 에너지의 양은 하루 동안의 총 열량 소모량인 대사량으로 구한다. 그중 기초 대사량은 생존에 필수적인 에너지로, 쾌적한 온도에서 편히 쉬는 동물이 공복 상태에서 생성하는 열량으로 정의된다. 이때 체내에서 생성한 열량은 일정한 체온에서 체외로 발산되는 열량과 같다. 기초 대사량은 개체에 따라 대사량의 60~75%를 차지하고, 근육량이 많을수록 증가한다.

기초 대사량은 직접법 또는 간접법으로 구한다. ㉠ 직접법은 온도가 일정하게 유지되고 공기의 출입량을 알고 있는 호흡실에서 동물이 발산하는 열량을 열량계를 이용해 측정하는 방법이다. ㉡ 간접법은 호흡 측정 장치를 이용해 동물의 산소 소비량과 이산화 탄소 배출량을 측정하고, 이를 기준으로 체내에서 생성된 열량을 추정하는 방법이다.

19세기의 초기 연구는 체외로 발산되는 열량이 체표 면적에 비례한다고 보았다. 즉 그 둘이 항상 일정한 비(比)를 갖는다는 것이다. 체표 면적은 (체중)^{0.67}에 비례하므로, 기초 대사량은 체중이 아닌 (체중)^{0.67}에 비례한다고 하였다. 어떤 변수의 증가율은 증가 후 값을 증가 전 값으로 나눈 값이므로, 체중이 W에서 2W로 커지면 체중의 증가율은 $(2W)/(W) = 2$ 이다. 이 경우에 기초 대사량의 증가율은 $(2W)^{0.67} / (W)^{0.67} = 2^{0.67}$, 즉 약 1.6이 된다.

1930년대에 클라이버는 생쥐부터 코끼리까지 다양한 크기의 동물의 기초 대사량 측정 결과를 분석했다. 그래프의 가로축 변수로 동물의 체중을, 세로축 변수로 기초 대사량을 두고, 각 동물별 체중과 기초 대사량의 순서쌍을 점으로 나타냈다.

- 동일한 의미를 나타낼 때는 같은 용어를 사용해야 합니다.
- 일례로, 가령, 예컨대, 예를 들어, 가정해본다면, ~이러한 상황을 생각해보자, 어떤 사례로 보자면
- 글을 쓸때는 동의어를 사용하면 지루함을 피할 수 있어서 좋지만 코드에서는 좋지 않습니다.
- 정말로 의미적으로 구분이 되어야 하는 경우에만 다른 용어를 사용해야 합니다.

2. 모호하고 의미 없는 이름은 피해야 합니다.

- data, info, thing, item, object, entity 같은 모호하고 의미 없는 이름은 삼가해야 합니다.
만약 entity라는 용어가 해당 분야에서 특별한 의미를 가진다면 괜찮습니다.
- 귀찮다고 무심코 의미 없는 이름을 붙여서는 안 됩니다.

3. 이름을 지을 때는 데이터 자체가 무엇인지를 고려해야 합니다.

- 이름을 지을 때는 포함된 내용이나 계산 방식이 아니라 데이터 자체가 무엇인지를 고려해야 합니다.
예를 들어, INodeList 보다는 Directory가 더 의미있는 이름입니다.
이는 구현의 측면이 아니라 개념적인 측면에서 디렉터리를 생각하게 합니다.
- 좋은 이름은 추상화의 수준을 높이고 의도치 않은 충돌의 위험성을 줄여 줍니다.

마무리

1. 가독성을 높이고, 추상화 수준을 올리기 위해서 **해당 분야의 용어를 사용**해야 합니다.
2. 같은 의미에 다른 이름을 붙이면 안됩니다. **특별한 의미가 있을 때만 용어를 구분**해야 합니다.