타입추론

추론 가능한 타입을 사용해 장황한 코드 방지하기
다른 타입에는 다른 변수 사용하기

추론 가능한 타입을 사용해
장황한 코드 방지하기

모든 변수에 타입을 선언하는 것

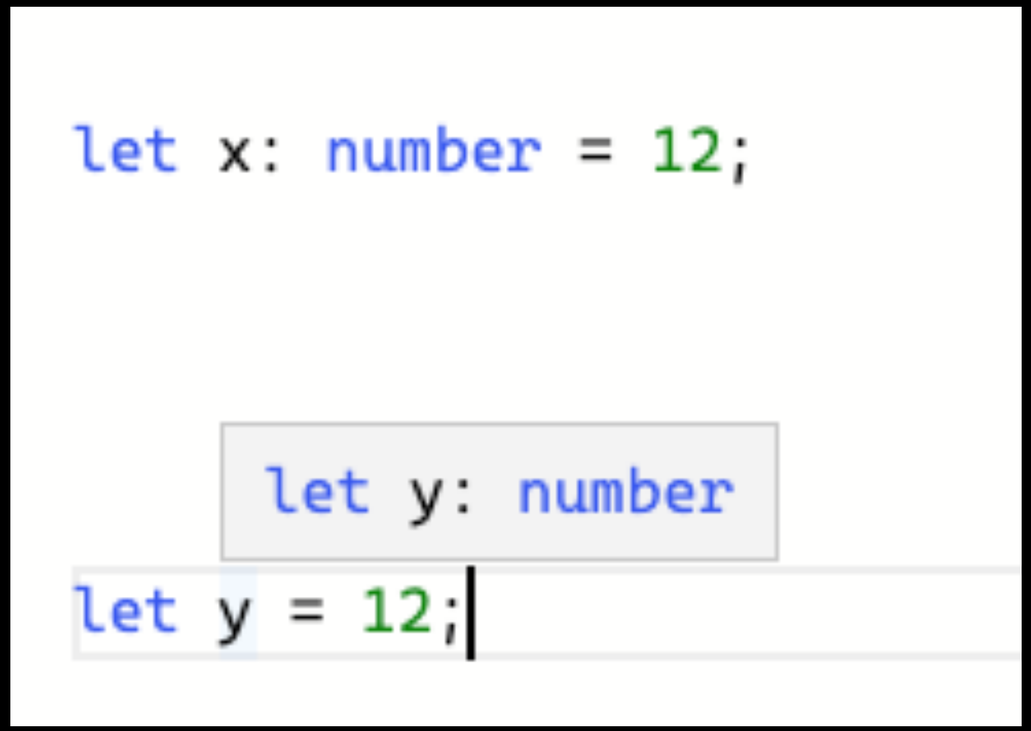
모든 변수에 타입을 선언하는 것

비 생산적

형편없는 스타일

자바스크립트 포팅 시 자주하는 실수

코드의 모든 변수에 타입 선언하기



타입이 추론된다면 타입 구문을 필요하지 않다

타입스크립트는 얼마나 정확하게
타입을 추론할 수 있을까?

타입스크립트는 얼마나 정확하게 타입을 추론할 수 있을까

복잡한 객체의 타입 추론

```
const person: {
  name: string;
  born: {
    where: string;
    when: string;
  };
  died: {
    where: string;
    when: string;
  }
} = {
  name: 'Sojourner Truth',
  born: {
    where: 'Swartekill, NY',
    when: 'c.1797',
  },
  died: {
    where: 'Battle Creek, MI',
    when: 'Nov. 26, 1883'
  }
};
```

```
const person: {
    name: string;
    born: {
        where: string;
        when: string;
    };
    died: {
        where: string;
        when: string;
    };
}
const person = {
  name: 'Sojourner Truth',
  born: {
    where: 'Swartekill, NY',
    when: 'c.1797',
  },
  died: {
    where: 'Battle Creek, MI',
    when: 'Nov. 26, 1883'
  }
};
```

타입스크립트는 얼마나 정확하게 타입을 추론할 수 있을까

함수의 반환 타입 추론

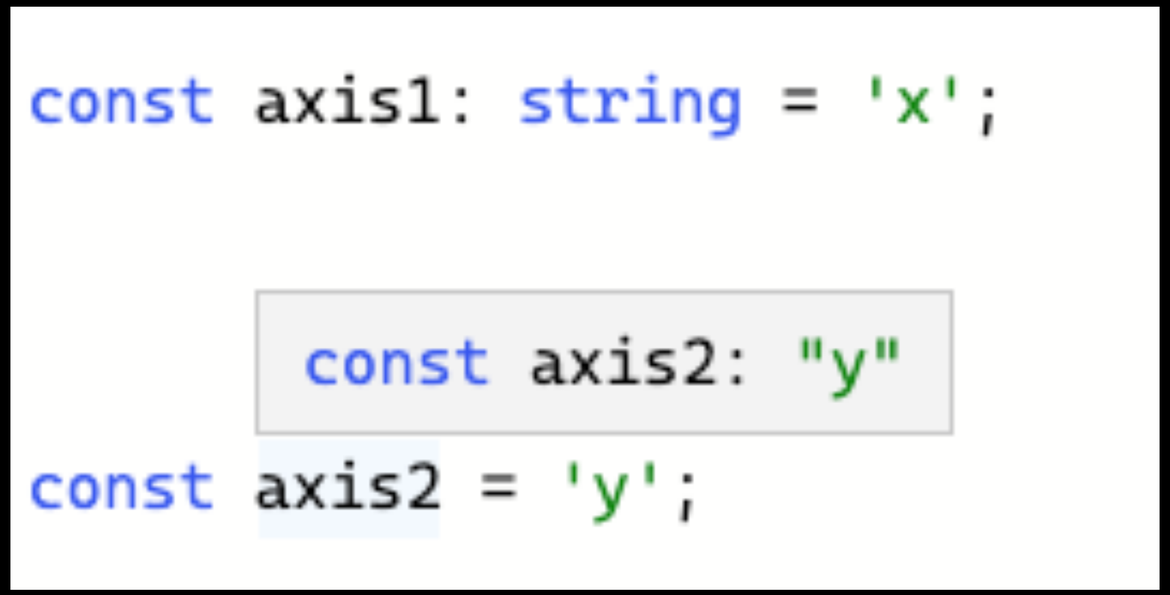
```
function square(nums: number[]) {
  return nums.map(x => x * x);
}

function square(nums: number[]): number[]

const squares = square([1, 2, 3, 4]); // Type is number[]
```

타입스크립트는 얼마나 정확하게 타입을 추론할 수 있을까

const 키워드로 타입 추론



타입추론의 장점

타입 추론의 장점

리팩터링에 용이해진다.

```ts
interface Product {
  id: string;
  name: string;
  price: number;
}

function logProduct(product: Product) {
  const id: number = product.id;
    // ~~ Type 'string' is not assignable to type 'number'
  const name: string = product.name;
  const price: number = product.price;
  console.log(id, name, price);
}

function logProduct2(product: Product) {
  const {id : id1, name: name1, price: price1} = product;
  const {id: id2, name: name2, price: price2}: {id: string; name: string; price: number } = product;

  console.log(id1, name1, price1);
  console.log(id2, name2, price2);
}
```

명시적 타입 구문이 필요한 경우

명시적 타입 구문이 필요할 경우

객체 리터럴을 정의할 경우

```
const elmo: Product = {
  name: 'Tickle Me Elmo',
  id: '048188 627152',
  price: 28.99,
};

const furby = {
  name: 'Furby',
  id: 630509430963,
  price: 35,
};

logProduct(furby);
logProduct(elmo);
```

명시적 타입 구문이 필요할 경우

함수의 반환

반환 타입을 명시해야하는 2가지 이유

- 반환 타입을 명시하면 함수에 대해 명확하게 알 수 있다.
- 명명된 타입을 사용할 수 있다.

결론

추론 가능한 타입을 사용해 장황한 코드 방지하기

1.타입 추론 가능할 경우 타입 구문을 작성하지 않는다.
2.추론될 수 있는 경우라도 객체 리터럴과 함수 반환에는 타입 명시를 고려해야 한다.