

Inferred Type Predicates (TypeScript 5.5)

typescript-5.5-beta.ts

- □ ×

```
const persons: (Person|null)[] = []  
  
const personNames = persons  
.filter(person => !person)  
.map(person => person.name)  
    // ^? Person
```

Because of Inferred Type Predicates, this code doesn't throw a compile error anymore in TypeScript 5.5!

In previous versions this was the case, which threw the error '**'person' is possibly 'null'**'. This was the case because the `person` was inferred as Person | null.

With TS 5.5 this is inferred as Person.



timdeschryver.dev/bits

Inferred Type Predicates (TypeScript 5.5)

ts typescript-4.5.ts

```
const personNames = persons
  .filter((person): person is Person => !!person)
  .map(person => person.name)
```

< TypeScript 5.5

ts typescript-5.5-beta.ts

```
const personNames = persons
  .filter((person) => !!person)
  .map(person => person.name)
```

>= TypeScript 5.5



timdeschryver.dev/bits

Inferred Type Predicates (TypeScript 5.5)

typescript-4.5.ts

```
// ⌛ Type predicate is required
function isFish(pet: Fish | Bird): pet is Fish {
    return 'swim' in pet;
}

let pet: Fish|Bird = getPet();

if (isFish(pet)) {
    pet.swim();
} else {
    pet.fly();
}
```

>= TypeScript 5.5

< TypeScript 5.5

typescript-5.5-beta.ts

```
// ⌛ Type predicate is not required anymore
function isFish(pet: Fish | Bird) {
    return 'swim' in pet;
}

let pet: Fish|Bird = getPet();

if (isFish(pet)) {
    pet.swim();
} else {
    pet.fly();
}
```



timdeschryver.dev/bits