

TypeScript 入門

■

- [// @ts-nocheck](#)
- [// @ts-check](#)
- [// @ts-ignore](#)
- [// @ts-expect-error](#)
- [JSDoc](#)
 - [@typedef](#)
 - [@type](#)
 - [@param](#)
 - [@return](#) [@returns](#)
 - [@extends](#) [_____](#)

TypeScript 入門

■ “ ” JS 入門

```
// @ts-nocheck
```

```
// @ts-nocheck _____  
■
```

TypeScript 入門

JavaScript

```
// @ts-nocheck
```

```
const element = document.getElementById(123);
```

```
document.getElementById(123)  
_____
```

```
// @ts-check
```

JavaScript // @ts-check

checkJs

```
// @ts-check
let isChecked = true;

console.log(isChceked); //
```

JavaScript // @ts-check TypeScript

// @ts-ignore

// @ts-ignore TypeScript

JavaScript

```
let x:number;

x = 0;

// @ts-ignore
x = false; //
```

 x number // @ts-

ignore

// @ts-expect-error

// @ts-expect-error TypeScript

```
function doStuff(abc: string, xyz: string) {
  assert(typeof abc === "string");
  assert(typeof xyz === "string");
  // do some stuff
}

expect(() => {
  // @ts-expect-error
```

```
doStuff(123, 456);  
}).toThrow();
```

@typedef

@typedef

TypeScript

```
/**
 * @typedef {(number | string)} NumberLike
 */
```

NumberLike

number string

TypeScript

```
type NumberLike = string | number;
```

@type

@type

```
/**
 * @type {string}
 */
let a;
```

@type

a

string

@type

@typedef

```
/**
 * @typedef {(number | string)} NumberLike
 */

/**
 * @type {NumberLike}
 */
let a = 0;
```

@type

TypeScript

```
/**@type {true | false} */
let a;

/** @type {number[]} */
let b;
```

```

/** @type {Array<number>} */
let c;

/** @type {{ readonly x: number, y?: string }} */
let d;

/** @type {(s: string, b: boolean) => number} */
let e;

```

@param

@param

```

/**
 * @param {string} x
 */
function foo(x) {}

```

```

/**
 * @param {string} [x]
 */
function foo(x) {}

```

```

/**
 * @param {string} [x="bar"]
 */
function foo(x) {}

```

x

bar

@return @returns

@return @returns

```

/**
 * @return {boolean}
 */
function foo() {
    return true;
}

/**
 * @returns {number}
 */
function bar() {
    return 0;
}

```

@extends

@extends 

```

/**
 * @extends {Base}
 */
class Derived extends Base {
}

```

@public  @protected  @private 

@readonly 

```

class Base {
    /**
     * @public
     * @readonly
     */
    x = 0;

    /**
     * @protected
     */
    y = 0;
}

```

□□ #2

□ □□ □□ 2023-07-09 12:07:22 UTC

□ □□ □□ 2023-12-16 14:51:21 UTC