Cheatography

TypeScript 2.1.4 Cheat Sheet

by Denis Bednov (denisbednov) via cheatography.com/33000/cs/10252/

Variables

var	Same as ES5 (function-scoped).
let	Block-scoped (like ES6).
const	Same as `let`, but readonly.

Base types

Boolean	`boolean: true | false`
Number	`number: decimal | hex | binary | octal`
String	`string: '...' | "..." | `...${...}...` `
Array	`type[] | type[][] | Array<type>`
Tuple	`[type, type, ...]`
Enum	`enum X { A = 1, B, C, ... }`
Any	`any`
Void	`function f(): void`
Null	`let n: null = null;`
Undefined	`let u: undefined = undefined;`
Never	Use for **functions**, which **never** ends.

--stri ctN ull Checks - check for assignement.

Destructuring

Array	`i = [1, 2];`
	`[f, s] = i;`
	`[f, s] = [s, f];`
	`f([f, s]: [number, number])`
	`[f, s, ...r] = [1, 2, 3, 4];`
	`[f] = [1, 2, 3, 4];`
	`[, s, , f] = [1, 2, 3, 4];`

Object	`o = { a: " f", b: 1, c: " b" }`
	`{ a, b } = o;`
	`({ a, b } = { a: " b", b: 1 });`
	`{ a, ...p } = o;`

Property renaming	`{ a: n1, b: n2 } = o;` `{ a, b }: { a: string, b: number } = o;`
Default values	`{ a, b = 1 } = o;`

Destructuring (cont)

Function declarations	`f(o: { a: string, b?: number })`

Spread	`let f = [1, 2];`
	`let s = [3, 4];`
	`let b = [0, ...f, ...s, 5];`
	`> [0, 1, 2, 3, 4, 5]`

Interface

```
interface LabelledValue {
        ...
}
```

Optional parameters

`name?: <ty pe>;`

Readonly properties

```
interface Point {
    readonly x: number;
        ...
}
```

ReadonlyArray<T>

Variables use **const** whereas properties use **readonly**.

By **Denis Bednov** (denisbednov)

cheatography.com/denisbednov/

Not published yet.
Last updated 16th December, 2016.