

# TyJS

## Requirements

Install node modules:

```
npm install
```

## How to use

Include the following in your code:

```
const Type = require("./tyjs.js");
```

Check some atom types!

```
const Type = require("./tyjs.js");

let tyJs = new Type("boolean | string");
console.log(tyJs.checks(true)); // true
console.log(tyJs.checks("hi")); // true
console.log(tyJs.checks(NaN)); // false
console.log(tyJs.checks(5)); // false
```

Check some more complex combinations using regex, values and so on:

```
tyJs = new Type("{...2*/a+/: boolean, \"def\": string, ...}");
console.log(tyJs.checks({ "aaa": true, "aaaa": true, def: "abc", "xyz": "abc" })); // true
console.log(tyJs.checks({ "aaa": true, "aaaa": true, "aaaaa": true, def: "abc", "xyz": "abc" })); // false
```

Define your custom checker functions:

```
tyJs = new Type("$0, [...$1]", [(value => value % 2 == 0), (value => value == 5)]);
console.log(tyJs.checks([2, [5, 5, 5]])); // true
console.log(tyJs.checks([2, [5, 4, 5]])); // false
```

Define custom class checkers:

```
tyJs = new Type("Set<boolean>");
tyJs.classChecker(Set, (values, typeCheckers) => {
  const [typeChecker] = typeCheckers;
  return typeCheckers.length == 1 && Array.from(values).every((value) => typeChecker(value));
});
console.log(tyJs.checks(new Set([true, false]))); // true
console.log(tyJs.checks(new Set([true, "abc"]))); // false
console.log(tyJs.checks([true, "abc"])); // false
console.log(tyJs.checks(new Set())); // true
```

## Optional steps

Recompile grammar:

```
nearleyc grammar.ne > grammar.js
```