006 TypeScript - union type

TypeScript Kata List – Blog Page

[TypeScript Kata List on GitHub](https://github.com/robertdunaway/katas-typescript)

# Duration

5 minutes

# Brief

How to use a union type.

### For More Information

BING/GOOGLE: “TypeScript union type”

# Instructions

Get tutorial folder or the entire katas-typescript repo.

Open the [before/\*.sln] file and execute the kata.

Feel free to execute this kata multiple times because repetition creates motor memory.

# Github

* Before (start kata with this)
  + https://github.com/robertdunaway/katas-typescript/tree/master/006%20TypeScript%20-%20union%20type/before
* After
  + https://github.com/robertdunaway/katas-typescript/tree/master/006%20TypeScript%20-%20union%20type/after

# Kata

Enter the following code to start off this kata.

// declare your union type variable for 3 different types.

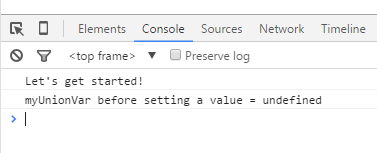
var myUnionVar: string | number | boolean;

console.log('myUnionVar before setting a value = ' + myUnionVar);

myUnionVar = 5;

console.log('typeof myUnionVar = ' + typeof myUnionVar);

When you execute this code you’ll find that, at this point, the myUnionVar is undefined.

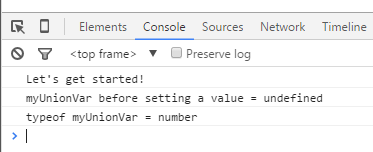


Set the myUnionVar to the number 5 and then use the “typeof” operator to determine the type of the union type variable.

[highlight below for one possible answer]

myUnionVar = 5;

console.log('typeof myUnionVar = ' + typeof myUnionVar);



A common scenario, ideal for union types, is when you don’t know if you are receiving a “thing” or an array of “things”.

Create a Union Type variable that can receive a single number or an array of numbers. Execute a simple “if” statement against the new variable and send output to the console window.

[highlight below for one possible answer]

var myUnionNumber: number | number[];

myUnionNumber = 100;

if (typeof myUnionNumber === 'number') {

console.log('myUnionNumber is a number');

}

else {

console.log('myUnionNumber is now an object');

}

myUnionNumber = [100, 200, 300];

if (typeof myUnionNumber === 'object') {

console.log('myUnionNumber is now an object');

}

else {

console.log('myUnionNumber is a number');

}

While we are at it, let’s go ahead and save the “typeof” result to another variable and then use that variable in a “switch” statement and output the result again. Why not!

[highlight below for one possible answer]

// use a switch statement around the typeof command.

var myVarType: string = typeof myUnionNumber;

switch (myVarType) {

case 'number':

{

console.log('myUnionNumber is a number');

break;

}

case 'object':

{

console.log('myUnionNumber is now an object');

break;

}

}

# Next

Take a few minutes and imagine more examples.