```
/*
JavaScript Type Conversions
In programming, type conversion is the process of
converting data of one type to another. For example:
converting String data to Number.
There are two types of type conversion in JavaScript.

Implicit Conversion - automatic type conversion
Explicit Conversion - manual type conversion

JavaScript Implicit Conversion
In certain situations, JavaScript automatically converts
one data type to another (to the right type). This is
known as implicit conversion.

Example 1: Implicit Conversion to String
// numeric string used with + gives string type
let result;

result = '3' + 2;
console.log(result) // "32"

result = '3' + true;
console.log(result); // "3true"

result = '3' + undefined;
console.log(result); // "3undefined"

result = '3' + null;
console.log(result); // "3null"

Note: When a number is added to a string, JavaScript
converts the number to a string before concatenation.

Example 2: Implicit Conversion to Number
```

```
// numeric string used with - , / , * results number
type

let result;

result = '4' - '2';
console.log(result); // 2

result = '4' - 2;
console.log(result); // 2

result = '4' * 2;
console.log(result); // 8

result = '4' / 2;
console.log(result); // 2

Example 3: Non-numeric String Results to NaN

// non-numeric string used with - , / , * results to NaN

let result;

result = 'hello' - 'world';
console.log(result); // NaN

result = '4' - 'hello';
console.log(result); // NaN

Example 4: Implicit Boolean Conversion to Number

// if boolean is used, true is 1, false is 0

let result;
```

```
result = '4' - true;
console.log(result); // 3

result = 4 + true;
console.log(result); // 5

result = 4 + false;
console.log(result); // 4

Note: JavaScript considers 0 as false and all non-zero
number as true. And, if true is converted to a number,
the result is always 1.

Example 5: null Conversion to Number

// null is 0 when used with number
let result;

result = 4 + null;
console.log(result);  // 4

result = 4 - null;
console.log(result);  // 4

Example 6: undefined used with number, boolean or null

// Arithmetic operation of undefined with number,
boolean or null gives NaN

let result;

result = 4 + undefined;
console.log(result);  // NaN
```

```
result = 4 - undefined;
console.log(result);  // NaN

result = true + undefined;
console.log(result);  // NaN

result = null + undefined;
console.log(result);  // NaN
```

JavaScript Explicit Conversion

You can also convert one data type to another as per your needs. The type conversion that you do manually is known as explicit type conversion.

In JavaScript, explicit type conversions are done using built-in methods.

Here are some common methods of explicit conversions.

1. Convert to Number Explicitly

To convert numeric strings and boolean values to numbers, you can use Number(). For example,

```
let result;

// string to number
result = Number('324');
console.log(result); // 324

result = Number('324e-1')
console.log(result); // 32.4

// boolean to number
result = Number(true);
console.log(result); // 1
```

```
result = Number(false);
console.log(result); // 0

In JavaScript, empty strings and null values return 0.
For example,

let result;
result = Number(null);
console.log(result);  // 0

let result = Number(' ')
console.log(result);  // 0

If a string is an invalid number, the result will be
NaN. For example,

let result;
result = Number('hello');
console.log(result); // NaN

result = Number(undefined);
console.log(result); // NaN

result = Number(NaN);
console.log(result); // NaN

Note: You can also generate numbers from strings using
parseInt(), parseFloat(), unary operator + and
Math.floor(). For example,

let result;
result = parseInt('20.01');
console.log(result); // 20
```

```
result = parseFloat('20.01');
console.log(result); // 20.01

result = +'20.01';
console.log(result); // 20.01

result = Math.floor('20.01');
console.log(result); // 20

2. Convert to String Explicitly
To convert other data types to strings, you can use
either String() or toString(). For example,

//number to string
let result;
result = String(324);
console.log(result);  // "324"

result = String(2 + 4);
console.log(result); // "6"

//other data types to string
result = String(null);
console.log(result); // "null"

result = String(undefined);
console.log(result); // "undefined"

result = String(NaN);
console.log(result); // "NaN"

result = String(true);
console.log(result); // "true"

result = String(false);
```

```
console.log(result); // "false"

// using toString()
result = (324).toString();
console.log(result); // "324"

result = true.toString();
console.log(result); // "true"
```

Note: String() takes null and undefined and converts them to string. However, toString() gives error when null are passed.

3. Convert to Boolean Explicitly

To convert other data types to a boolean, you can use Boolean().

In JavaScript, undefined, null, 0, NaN, '' converts to false. For example,

```
let result;
result = Boolean('');
console.log(result); // false

result = Boolean(0);
console.log(result); // false

result = Boolean(undefined);
console.log(result); // false

result = Boolean(null);
console.log(result); // false

result = Boolean(NaN);
console.log(result); // false
```

```
All other values give true. For example,

result = Boolean(324);
console.log(result); // true

result = Boolean('hello');
console.log(result); // true

result = Boolean(' ');
console.log(result); // true

JavaScript Type Conversion Table
The table shows the conversion of different values to
String, Number, and Boolean in JavaScript.

Value   String Conversion   Number Conversion   Boolean
Conversion
1            "1"                  1                tru
e
0            "0"                  0                fal
se
"1"          "1"                  1                tru
e
"0"          "0"                  0                tru
e
"ten"       "ten"                NaN               tru
e
true        "true"                1                tru
e
false       "false"               0                fal
se
null        "null"                0                fal
se
```

```
undefined   "undefined"           NaN                fal
se
''          ""                     0                 fal
se
' '         "
"                   0                true
You will learn about the conversion of objects and
arrays to other data types in later tutorials.
 */

// //number
// //example
// let result;

// result = '4' - '2';
// console.log(result);

// //example
// //string
// let result1;

// result1 = 'abc' - 'efg';
// console.log(result1);

// //example
// let result3;

// result3 = '4' - 2;
// console.log(result3);

// //example
// let result4;

// result3 = '2' - 'efg';
// console.log(result3);
```

```
//boolean

// let res;
// res = '4' - true;
// console.log(res);

// let res;
// res = '4' + true;
// console.log(res);

// let res;
// res = 4 + true;
// console.log(res);

// let res;
// res = 4 + false;
// console.log(res); //4

//null

// let res;
// res = 4 + null;
// console.log(res); //4

// let res;
// res = 4-null;
// console.log(res); //4

//Explicity

//Number

// let res;
```

```
// res = Number("113");
// console.log(res);

// res = Number(true);
// console.log(res);

// res = Number(false);
// console.log(res);

// res = Number(null);
// console.log(res);

// res = Number(' ');
// console.log(res);

// res = Number('hello');
// console.log(res);

// res = Number(undefined);
// console.log(res);

//string

// let res;

// res = String(324);
// console.log(res);

// res = String(2+6);
// console.log(res);

// res = String(null);
// console.log(res);

// res = String(undefined);
```

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

// res = String(true);
// console.log(res);

// res = String(false);
// console.log(res);

// res = String(NaN);
// console.log(res);

//boolean

// let result;
// result = Boolean('');
// console.log(result); // false

// result = Boolean(0);
// console.log(result); // false

// result = Boolean(undefined);
// console.log(result); // false

// result = Boolean(null);
// console.log(result); // false

// result = Boolean(NaN);
// console.log(result); // false

//spread operator:

// let arr = [1, 2, 3];

// let arr1 = [5, 4, 6, ...arr];
// console.log(arr1);
```

```
// let arr1 = [...arr, 5, 4, 6];
// console.log(arr1);

// let arr1 = [5, 4, ...arr, 6,];
// console.log(arr1);

// let arr2 = [4, 5, 6];
// let arr3 = [...arr, ...arr2];
// console.log(arr3);

//rest paramaters:

// function sum(...args) {
//   console.log(args);
// }
// sum();
// sum(1);
// sum(1, 2, 3, 4);

function sum(...args) {
  let sum = 0;
  for (let i = 0; i < args.length; i++) {
    sum = sum + args[i];
  }
  console.log(sum);
}
sum();
sum(1);
sum(1, 2, 3, 4);
```