# DataType Conversion-JS Vaishnu

```
javascriptCopy code
let score = 33;
let score1 = "33";
console.log(typeof score);      // number      -> Syntax1
console.log(typeof(score1));     // string      -> Syntax2
```

- Two variables are declared: `score` and `score1`.
- `score` is assigned the numeric value `33`, and `score1` is assigned the string value `"33"`.
- The first `console.log` prints the data type of the variable `score`, which is `number`.
- The second `console.log` prints the data type of the variable `score1`, which is `string`.

```
javascriptCopy code
let score2 = "23";          // String datatype
let valueInNumber = Number(score2);        // String to number conversion using Number
class
console.log(typeof valueInNumber);
```

- A variable `score2` is declared and assigned the string value `"23"`.
- The `Number()` function is used to convert the string value of `score2` to a numeric value and store it in the variable `valueInNumber`.
- The `console.log` prints the data type of `valueInNumber`, which is `number`.

```
javascriptCopy code
let score3 = "23va1";
let valueInNumber1 = Number(score3);
console.log(typeof valueInNumber1);    // number
console.log(valueInNumber1);    // NaN
```

- A variable `score3` is declared and assigned the string value `"23va1"`.

- The `Number()` function attempts to convert the string value of `score3` to a numeric value. Since the string contains non-numeric characters, the conversion results in `NaN` (Not a Number).
- The first `console.log` prints the data type of `valueInNumber1`, which is `number`.
- The second `console.log` prints the value of `valueInNumber1`, which is `NaN`.

```
javascriptCopy code
let score4 = null;
let valueInNull = Number(score4);
console.log(typeof valueInNull);   // number
console.log(valueInNull);              // 0
```

- A variable `score4` is declared and assigned the value `null`.
- The `Number()` function attempts to convert the value of `score4` to a numeric value. Since `null` represents the absence of a value, it is converted to `0`.
- The first `console.log` prints the data type of `valueInNull`, which is `number`.
- The second `console.log` prints the value of `valueInNull`, which is `0`.

```
javascriptCopy code
let score5 = undefined;
let valueInUndefined = Number(score5);
console.log(typeof valueInUndefined);   // number
console.log(valueInUndefined);           // NaN
```

- A variable `score5` is declared but not assigned any value, so it holds the value `undefined`.
- The `Number()` function attempts to convert the value of `score5` to a numeric value. Since `undefined` represents an uninitialized variable, it is converted to `NaN`.
- The first `console.log` prints the data type of `valueInUndefined`, which is `number`.
- The second `console.log` prints the value of `valueInUndefined`, which is `NaN`.

```
javascriptCopy code
let score6 = true;
let valueInBoolean = Number(score6);
console.log(typeof valueInBoolean);   // number
console.log(valueInBoolean);           // 1
```

- A variable `score6` is declared and assigned the boolean value `true`.
- The `Number()` function attempts to convert the boolean value of `score6` to a numeric value. `true` is converted to `1`, and `false` would be converted to `0`.
- The first `console.log` prints the data type of `valueInBoolean`, which is `number`.
- The second `console.log` prints the value of `valueInBoolean`, which is `1`.

```
javascriptCopy code
let score7 = "Vaishnu";
let valueInString = Number(score7);
console.log(typeof valueInString);   // number
console.log(valueInString);          // NaN
```

- A variable `score7` is declared and assigned the string value `"Vaishnu"`.
- The `Number()` function attempts to convert the string value of `score7` to a numeric value. However, since the string contains non-numeric characters, the conversion results in `NaN`.
- The first `console.log` prints the data type of `valueInString`, which is `number`.
- The second `console.log` prints the value of `valueInString`, which is `NaN`.

```
javascriptCopy code
let isLoggedIn = 1;
let booleanIsLoggedIn = Boolean(isLoggedIn);
console.log(booleanIsLoggedIn);        // true

// 1 => true, 0 => false
// "" => false    // emptyString => false
// "Vaishnu"  => true
```

- A variable `isLoggedIn` is declared and assigned the numeric value `1`.
- The `Boolean()` function is used to convert the numeric value of `isLoggedIn` to a boolean value. Since `1` is a truthy value, the conversion results in `true`.
- The `console.log` prints the value of `booleanIsLoggedIn`, which is `true`.

- The comments provide examples of the conversions between numbers and booleans.

```
javascriptCopy code
let someNumber = 33;
let stringNumber = String(someNumber);
console.log(typeof stringNumber);     // string
console.log(stringNumber);              // "33" (but in console it will not show like
 this)
```

- A variable `someNumber` is declared and assigned the numeric value `33`.
- The `String()` function is used to convert the numeric value of `someNumber` to a string value, and it is stored in the variable `stringNumber`.
- The first `console.log` prints the data type of `stringNumber`, which is `string`.
- The second `console.log` prints the value of `stringNumber`, which is `"33"`.

**Notes:**

- When using the `Number()` function to convert a string to a number, it will attempt to parse the string as a numeric value. If the string contains non-numeric characters, the conversion results in `NaN`.
- When using the `Boolean()` function to convert a value to a boolean, the following rules apply: `false`, `0`, an empty string, `null`, `undefined`, and `NaN` are converted to `false`. Any other value, including non-empty strings, numbers other than `0`, and objects, are converted to `true`.