Type conversion in JavaScript

Type conversion in JavaScript, also known as type coercion, is the process of converting a value from one type to another. JavaScript is a loosely-typed language, which means that you can convert types explicitly and implicitly. Here's a detailed explanation with examples for each type of conversion you mentioned:

Converting Strings to Numbers

1. **Using `Number()` function:**

   ```
   let str = "123";
   let num = Number(str);
   console.log(num); // 123
   ```

2. **Using `parseInt()` and `parseFloat()`:**

   ```
   let strInt = "123";
   let strFloat = "123.45";
   let numInt = parseInt(strInt);
   let numFloat = parseFloat(strFloat);
   console.log(numInt); // 123
   console.log(numFloat); // 123.45
   ```

3. **Using the unary + operator:**

   ```
   let str = "123";
   let num = +str;
   console.log(num); // 123
   ```

Converting Numbers to Strings

1. **Using `String()` function:**

   ```
   let num = 123;
   let str = String(num);
   console.log(str); // "123"
   ```

2. **Using `toString()` method:**

   ```
   let num = 123;
   let str = num.toString();
   console.log(str); // "123"
   ```

3. **Using template literals:**

   ```
   let num = 123;
   let str = `${num}`;
   console.log(str); // "123"
   ```

Converting Dates to Numbers

1. **Using `getTime()` method:**

```
let date = new Date();
let num = date.getTime();
console.log(num); // Number of milliseconds since January 1, 1970
```

2. **Using unary + operator:**

```
let date = new Date();
let num = +date;
console.log(num); // Number of milliseconds since January 1, 1970
```

Converting Numbers to Dates

1. **Using `new Date()` constructor:**

```
let num = 1628502023245; // Example timestamp
let date = new Date(num);
console.log(date); // Date object representing the timestamp
```

Converting Booleans to Numbers

1. **Using `Number()` function:**

```
let boolTrue = true;
let boolFalse = false;
console.log(Number(boolTrue)); // 1
console.log(Number(boolFalse)); // 0
```

2. **Using unary + operator:**

```
let boolTrue = true;
let boolFalse = false;
console.log(+boolTrue); // 1
console.log(+boolFalse); // 0
```

Converting Numbers to Booleans

1. **Using `Boolean()` function:**

```
let num1 = 1;
let num0 = 0;
console.log(Boolean(num1)); // true
console.log(Boolean(num0)); // false
```

2. **Using double negation `!!` operator:**

```
let num1 = 1;
let num0 = 0;
console.log(!!num1); // true
console.log(!!num0); // false
```

Additional Concepts

- **Implicit Conversion (Coercion):** JavaScript often converts types implicitly. For example, when using the == operator, JavaScript converts types to make the comparison.

  ```
  console.log("123" == 123); // true (string to number conversion)
  console.log(true == 1);    // true (boolean to number conversion)
  ```

- **NaN (Not-a-Number):** When a string that cannot be converted to a number is passed to Number(), it returns NaN.

  ```
  let str = "abc";
  let num = Number(str);
  console.log(num); // NaN
  ```

Summary

Type conversion is an essential part of JavaScript programming. Understanding how and when to convert types can help prevent bugs and make your code more predictable. Explicit conversions are generally more readable and less error-prone compared to implicit conversions.