Type coercion and type conversion

🔄 Type Coercion vs Type Conversion in JavaScript

JavaScript is a **dynamically typed** language — this means variable types are determined at runtime. Because of this, JavaScript often automatically converts data types **behind the scenes** (type coercion), but you can also explicitly convert them (type conversion).

✅ 1. Type Conversion (Explicit)

> You manually convert a value from one data type to another using built-in functions.

🔷 Examples:

```
// String to Number
let str = "123";
let num = Number(str);  // 123

// Number to String
let num2 = 456;
let str2 = String(num2); // "456"

// Boolean to String
String(true);    // "true"
String(false);   // "false"

// Number to Boolean
Boolean(0);      // false
Boolean(1);      // true
```

🧠 Common Functions:

- `String(value)`
- `Number(value)`
- `Boolean(value)`
- `parseInt(value)`
- `parseFloat(value)`

⚠️ 2. Type Coercion (Implicit)

> JavaScript automatically converts one data type to another when performing operations involving different types.

🔷 Examples:

```
// String coercion (number + string)
let result = 5 + "10";   // "510" (number coerced to string)

// Number coercion
let result2 = "5" * 2;   // 10 (string coerced to number)
let result3 = "5" - 1;   // 4

// Boolean coercion
let value = !!"hello";   // true (non-empty string is truthy)
```

🧪 Type Coercion Rules

Operation	Coercion Result
`+`	If one operand is a string → both are coerced to strings
`-`, `*`, `/`	Operands are coerced to numbers
`==`	Allows coercion (use `===` to avoid it)
`===`	Strict, **no coercion**

⚠️ Examples of Weird Coercions

```
console.log(null + 1);       // 1
console.log(undefined + 1);  // NaN
console.log(true + false);   // 1
console.log("5" - "2");      // 3
console.log("5" + 1);        // "51"
console.log(5 + null);       // 5
console.log("5" * "2");      // 10
```

🔍 Equality Comparison: `==` vs `===`

```
console.log("5" == 5);   // true (type coerced)
console.log("5" === 5);  // false (no coercion, types differ)
```

- `==`: Loose equality – performs type coercion
- `===`: Strict equality – does **not** perform coercion

🧠 Summary

Concept	Description
Type Conversion	You manually change the type
Type Coercion	JavaScript changes the type automatically
Safer Practice	Use **explicit conversion** to avoid bugs
Preferred Equals	Use `===` instead of `==` for comparison

Would you like a visual cheatsheet or a real-world code example next?