

A light blue background with a network of white lines and dots, resembling a molecular or digital structure. A single white diagonal line is in the top left corner.

# 기초 웹

22 Winter CNU 기초 스터디

21 남정연

21 박준서

---

# Types of Numbers

In programming, even the humble decimal number system that we all know so well is more complicated than you might think. We use different terms to describe different types of decimal numbers, for example:

- **Integers** are floating-point numbers without a fraction. They can either be positive or negative, e.g. 10, 400, or -5.
- **Floating point numbers** (floats) have decimal points and decimal places, for example 12.5, and 56.7786543.
- **Doubles** are a specific type of floating point number that have greater precision than standard floating point numbers (meaning that they are accurate to a greater number of decimal places).

We even have different types of number systems! Decimal is base 10 (meaning it uses 0–9 in each column), but we also have things like:

- **Binary** — The lowest level language of computers; 0s and 1s.
- **Octal** — Base 8, uses 0–7 in each column.
- **Hexadecimal** — Base 16, uses 0–9 and then a–f in each column. You may have encountered these numbers before when setting [colors in CSS](#).

**Before you start to get worried about your brain melting, stop right there!** For a start, we are just going to stick to decimal numbers throughout this course; you'll rarely come across a need to start thinking about other types, if ever.

The second bit of good news is that unlike some other programming languages, JavaScript only has one data type for numbers, both integers and decimals — you guessed it, `Number`. This means that whatever type of numbers you are dealing with in JavaScript, you handle them in exactly the same way.

# Arithmetic Operators

Operator	Name	Purpose	Example
+	Addition	Adds two numbers together.	<code>6 + 9</code>
-	Subtraction	Subtracts the right number from the left.	<code>20 - 15</code>
*	Multiplication	Multiplies two numbers together.	<code>3 * 7</code>
/	Division	Divides the left number by the right.	<code>10 / 5</code>
%	Remainder (sometimes called modulo)	Returns the remainder left over after you've divided the left number into a number of integer portions equal to the right number.	<code>8 % 3</code> (returns 2, as three goes into 8 twice, leaving 2 left over).
**	Exponent	Raises a <code>base</code> number to the <code>exponent</code> power, that is, the <code>base</code> number multiplied by itself, <code>exponent</code> times. It was first Introduced in EcmaScript 2016.	<code>5 ** 2</code> (returns 25 , which is the same as <code>5 * 5</code> ).

# Increment and Decrement Operators

```
guessCount++;
```

Let's try playing with these in your console. For a start, note that you can't apply these directly to a number, which might seem strange, but we are assigning a variable a new updated value, not operating on the value itself. The following will return an error:

```
3++;
```

So, you can only increment an existing variable. Try this:

```
let num1 = 4;  
num1++;
```

# Assignment Operators

Operator	Name	Purpose	Example	Shortcut for
<code>+=</code>	Addition assignment	Adds the value on the right to the variable value on the left, then returns the new variable value	<code>x += 4;</code>	<code>x = x + 4;</code>
<code>-=</code>	Subtraction assignment	Subtracts the value on the right from the variable value on the left, and returns the new variable value	<code>x -= 3;</code>	<code>x = x - 3;</code>
<code>*=</code>	Multiplication assignment	Multiplies the variable value on the left by the value on the right, and returns the new variable value	<code>x *= 3;</code>	<code>x = x * 3;</code>
<code>/=</code>	Division assignment	Divides the variable value on the left by the value on the right, and returns the new variable value	<code>x /= 5;</code>	<code>x = x / 5;</code>

# Strings

```
const string = 'The revolution will not be televised.';  
console.log(string);
```

```
const badString1 = This is a test;  
const badString2 = 'This is a test;  
const badString3 = This is a test';
```

# Quotes

```
const sgl = 'Single quotes.';  
const dbl = "Double quotes";  
console.log(sgl);  
console.log(dbl);
```

```
const badQuotes = 'What on earth?';
```

```
const sglDbl = 'Would you eat a "fish supper"?';  
const dblSgl = "I'm feeling blue.";  
console.log(sglDbl);  
console.log(dblSgl);
```

# Escaping Characters

```
const bigmouth = 'I\'ve got no right to take my place...';  
console.log(bigmouth);
```



# Concatenating String

```
const greeting = `Hello`;
```

```
const name = 'Chris';  
const greeting = `Hello, ${name}`;  
console.log(greeting); // "Hello, Chris"
```

```
const one = 'Hello, ';  
const two = 'how are you?';  
const joined = `${one}${two}`;  
console.log(joined); // "Hello, how are you?"
```

# Concatenating String

```
const greeting = "Hello";  
const name = "Chris";  
console.log(greeting + ", " + name); // "Hello, Chris"
```

```
const greeting = "Hello";  
const name = "Chris";  
console.log(`${greeting}, ${name}`); // "Hello, Chris"
```

# Concatenating String

```
const myString = '123';  
const myNum = Number(myString);  
console.log(typeof myNum);
```

```
const myNum2 = 123;  
const myString2 = myNum2.toString();  
console.log(typeof myString2);
```

# Multiline Strings

```
const output = `I like the song.  
I gave it a score of 90%.`;  
console.log(output); // I like the song.  
                    // I gave it a score of 90%.
```

```
const output = 'I like the song.\nI gave it a score of 90%.';  
console.log(output); // I like the song.  
                    // I gave it a score of 90%.
```

# String Property

```
const browserType = 'mozilla';  
browserType.length;
```

# String Indexing

```
browserType[0];
```

```
browserType[browserType.length-1];
```

# String Methods

```
const browserType = 'mozilla';

if (browserType.includes('zilla')) {
  console.log('Found zilla!');
} else {
  console.log('No zilla here!');
}
```

```
const browserType = 'mozilla';

if (browserType.startsWith('zilla')) {
  console.log('Found zilla!');
} else {
  console.log('No zilla here!');
}
```

```
const browserType = 'mozilla';

if (browserType.endsWith('zilla')) {
  console.log('Found zilla!');
} else {
  console.log('No zilla here!');
}
```

# String Methods

```
const browserType = 'mozilla';  
console.log(browserType.slice(1, 4)); // "ozi"
```

```
browserType.slice(2); // "zilla"
```



# String Methods

```
const radData = 'My NaMe Is MuD';  
console.log(radData.toLowerCase());  
console.log(radData.toUpperCase());
```

# String Methods

```
const browserType = 'mozilla';  
const updated = browserType.replace('moz', 'van');  
  
console.log(updated);    // "vanilla"  
console.log(browserType); // "mozilla"
```

```
let browserType = 'mozilla';  
browserType = browserType.replace('moz', 'van');  
  
console.log(browserType); // "vanilla"
```

# Arrays

```
const shopping = ['bread', 'milk', 'cheese', 'hummus', 'noodles'];  
console.log(shopping);
```

```
const sequence = [1, 1, 2, 3, 5, 8, 13];  
const random = ['tree', 795, [0, 1, 2]];
```

# Array Property

```
const shopping = ['bread', 'milk', 'cheese', 'hummus', 'noodles'];  
console.log(shopping.length); // 5
```

# Array Property

```
const shopping = ['bread', 'milk', 'cheese', 'hummus', 'noodles'];  
console.log(shopping[0]);  
// returns "bread"
```

```
const shopping = ['bread', 'milk', 'cheese', 'hummus', 'noodles'];  
shopping[0] = 'tahini';  
console.log(shopping);  
// shopping will now return [ "tahini", "milk", "cheese", "hummus",
```

```
const random = ['tree', 795, [0, 1, 2]];  
random[2][2];
```

# Array Methods

```
const birds = ['Parrot', 'Falcon', 'Owl'];  
console.log(birds.indexOf('Owl'));    // 2  
console.log(birds.indexOf('Rabbit')) // -1
```

# Array Methods

```
const myArray = ['Manchester', 'Liverpool'];  
myArray.push('Cardiff');  
console.log(myArray);    // [ "Manchester", "Liverpool", "Cardiff" ]  
myArray.push('Bradford', 'Brighton');  
console.log(myArray);    // [ "Manchester", "Liverpool", "Cardiff", "Bradford", "Brighton" ]
```

```
const myArray = ['Manchester', 'Liverpool'];  
const newLength = myArray.push('Bristol');  
console.log(myArray);    // [ "Manchester", "Liverpool", "Bristol" ]  
console.log(newLength);  // 3
```

```
const myArray = ['Manchester', 'Liverpool'];  
myArray.unshift('Edinburgh');  
console.log(myArray);    // [ "Edinburgh", "Manchester", "Liverpool" ]
```

# Array Methods

```
const myArray = ['Manchester', 'Liverpool'];  
myArray.pop();  
console.log(myArray);    // [ "Manchester" ]
```

```
const myArray = ['Manchester', 'Liverpool'];  
const removedItem = myArray.pop();  
console.log(removedItem);    // "Liverpool"
```

```
const myArray = ['Manchester', 'Liverpool'];  
myArray.shift();  
console.log(myArray);    // [ "Liverpool" ]
```



# Array Methods

```
const myArray = ['Manchester', 'Liverpool', 'Edinburgh', 'Carlisle'];
const index = myArray.indexOf('Liverpool');
if (index !== -1) {
  myArray.splice(index, 1);
}
console.log(myArray);    // [ "Manchester", "Edinburgh", "Carlisle" ]
```

```
const myArray = ['Manchester', 'Liverpool', 'Edinburgh', 'Carlisle'];
const index = myArray.indexOf('Liverpool');
if (index !== -1) {
  myArray.splice(index, 2);
}
console.log(myArray);    // [ "Manchester", "Carlisle" ]
```

# Array Methods

```
const birds = ['Parrot', 'Falcon', 'Owl'];

for (const bird of birds) {
  console.log(bird);
}
```

```
function double(number) {
  return number * 2;
}

const numbers = [5, 2, 7, 6];
const doubled = numbers.map(double);

console.log(doubled); // [ 10, 4, 14, 12 ]
```

```
function isLong(city) {
  return city.length > 8;
}

const cities = ['London', 'Liverpool', 'Totnes', 'Edinburgh'];
const longer = cities.filter(isLong);

console.log(longer); // [ "Liverpool", "Edinburgh" ]
```

# String to Array Conversion

```
const myData = 'Manchester,London,Liverpool,Birmingham,Leeds,Carlisle';
```

```
const myArray = myData.split(',');  
myArray;
```

# Array to String Conversion

```
const myNewString = myArray.join(',');  
myNewString;
```

```
const dogNames = ['Rocket','Flash','Bella','Slugger'];  
dogNames.toString(); // Rocket,Flash,Bella,Slugger
```