Chapter 4

Expressions and Data Manipulation

Introduction

In this chapter, we'll unlock one of n8n's most powerful features: expressions. Expressions allow you to dynamically access, transform, and manipulate data as it flows through workflows, turning static automations into intelligent, adaptive systems.

What Are Expressions?

Expression: Code wrapped in double curly braces `{{ }}` that dynamically evaluates to produce a value. Expressions can access data, perform calculations, manipulate strings, and apply JavaScript functions.

Expression Syntax Basics

The Double Curly Braces

All expressions must be wrapped in `{{ }}`. Without these braces, n8n treats content as literal text.

Examples:

• Literal: `Hello, $json.name` → Outputs exactly as written

• Expression: `{{ $json.name }}` → Outputs 'Hello, John Smith'

Basic Data Access

Expression	What It Accesses

`{{ $json }}`	Entire JSON object
`{{ $json.propertyName }}`	Specific property
`{{ $json.nested.property }}`	Nested property
`{{ $json.array[0] }}`	First array item
`{{ $node["Node Name"].json }}`	Data from specific node

Working with Strings

String Methods

Method	Purpose	Example
`.toUpperCase()`	Convert to uppercase	`{{ $json.name.toUpperCase() }}`
`.toLowerCase()`	Convert to lowercase	`{{ $json.email.toLowerCase() }}`
`.trim()`	Remove whitespace	`{{ $json.name.trim() }}`
`.replace()`	Replace text	`{{ $json.phone.replace("-", "") }}`
`.split()`	Split into array	`{{ $json.tags.split(",") }}`

Practical Examples:

• **Create email from name:** `{{ $json.name.toLowerCase().replace(" ", ".") + "@company.com" }}`

• **Extract first name:** `{{ $json.fullName.split(" ")[0] }}`

Working with Numbers

Math Operations

Operation	Example	Result
`Addition`	`{{ $json.price + 5 }}`	`105 (if price = 100)`
`Multiplication`	`{{ $json.quantity * $json.price }}`	`500 (if qty=5, price=100)`
`Percentage`	`{{ $json.total * 0.08 }}`	`8 (8% of 100)`

Common Calculations:

- **Sales tax:** `{{ ($json.subtotal * 0.08).toFixed(2) }}`
- **Discount:** `{{ Math.round((1 - $json.salePrice / $json.originalPrice) * 100) }}%`

Working with Dates and Times

Date Formatting

Expression	Output
`{{ $now.toFormat('yyyy-MM-dd') }}`	`2025-11-21`
`{{ $now.toFormat('MMM dd, yyyy') }}`	`Nov 21, 2025`
`{{ $now.toFormat('HH:mm:ss') }}`	`14:30:00`

Date Calculations

Common Operations:

- **Add 7 days:** `{{ $now.plus({ days: 7 }).toFormat('yyyy-MM-dd') }}`
- **Subtract 30 days:** `{{ $now.minus({ days: 30 }).toFormat('yyyy-MM-dd') }}`

Conditional Logic (Ternary Operator)

Syntax: `{{ condition ? valueIfTrue : valueIfFalse }}`

Examples:

- **Customer status:** `{{ $json.total > 1000 ? "VIP" : "Standard" }}`
- **Time greeting:** `{{ $now.hour < 12 ? "Good morning" : "Good afternoon" }}`

Working with Arrays

Array Methods

Method	Purpose
`.length`	`Number of items`
`.map()`	`Transform each item`
`.filter()`	`Keep items matching condition`
`.join()`	`Combine into string`

Practical Examples:

- **Get all names:** `{{ $json.products.map(p => p.name).join(", ") }}`
- **Calculate total:** `{{ $json.items.reduce((sum, item) => sum + item.price, 0) }}`

Real-World Scenarios

E-commerce Order Processing

Calculate with shipping:

```
{{ ($json.subtotal + ($json.subtotal < 50 ? 5.99 : 0)).toFixed(2) }}
```

Free shipping over $50, otherwise $5.99

Customer Segmentation

Determine tier:

```
{{ $json.lifetimeValue > 10000 ? "Platinum" : $json.lifetimeValue > 5000 ? "Gold" : "Silver"
}}
```

Best Practices

Guidelines:

1. Keep expressions simple—use Set/Code nodes for complex logic

2. Handle null values with optional chaining: $json.user?.profile?.name

3. Test expressions using preview feature

4. Comment complex logic in node notes

5. Use default values: {{ $json.quantity || 1 }}

Common Errors

Error 1: 'Cannot read property of undefined'

Solution: Use optional chaining or check existence first

Error 2: Expression returns '[object Object]'

Solution: Access specific properties or use JSON.stringify()

Practice Exercises

Exercise 1: Email Personalization

Create personalized greeting based on time and customer name

• Extract first name from full name

• Show appropriate greeting for time of day

Exercise 2: Price Calculator

Calculate final price with discounts and tax

• 10% discount if quantity >= 10

• Add 8% sales tax

Key Takeaways

• Expressions {{ }} enable dynamic data access and transformation

• Use JavaScript methods for strings, numbers, arrays, dates

• Ternary operators enable inline conditional logic

• Always handle null/undefined values

• Keep expressions simple; use Set nodes for complexity

• Test expressions before saving

Looking Forward

In Chapter 5, we'll explore workflow control and logic. You'll learn IF nodes, Switch nodes, loops, error handling, and building workflows that make intelligent decisions.