Fourward Programming Language Documentation

Introduction

Fourward is a simple, beginner-friendly programming language made to teach fundamental programming concepts. Its clean and minimal syntax reduces complexity, making it easy for new coders.

Language Syntax

1. **Statement Structure**: Statements end with a semicolon (`;`).
2. **Indentation**: Not required but encouraged for readability.
3. **Comments**: Use `#` for single-line comments.

```
# This is a comment
let x = 5;
```

Reserved Keywords

Fourward reserves the following words:

```
let, const, if, else, while, for, function, return, print, input, true,
false, null
```

Data Types

- **int**: Whole numbers (e.g., `5`, `-3`)
- **float**: Decimal numbers (e.g., `3.14`, `-0.5`)
- **str**: Strings enclosed in double quotes (e.g., `"Hello"`)
- **bool**: Boolean values (`true`, `false`)
- **null**: Represents the absence of a value

Arithmetic Operators

- `+` Addition
- `-` Subtraction
- `*` Multiplication
- `/` Division
- `%` Modulus

Comparative Operators

- `==` Equal to
- `!=` Not equal to
- `>` Greater than
- `<` Less than
- `>=` Greater than or equal to
- `<=` Less than or equal to

Control Flow Statements

Conditional Statements

```
if (x > 10) {
    print("x is large");
} else {
    print("x is small");
}
```

Loops

While Loop:

```
while (x < 10) {
    print(x);
    x = x + 1;
}
```

For Loop:

```
for (let i = 0; i < 5; i++) {
    print(i);
}
```

Functions

Functions are defined using the `function` keyword. Use the `return` keyword to return a value.

Example:

```
function add(a, b) {
    return a + b;
}
```

Input and Output

- `print`: Displays output to the console.
- `input`: Reads input from the user.

Token Identification

Tokens are identified by scanning the input stream and noting their **line number** and **column position**. The **symbol table** records each token's type and location for use during parsing and compilation.

Example Token Log:

```
Line 1, Col 1: let (keyword)
Line 1, Col 5: x (identifier)
Line 1, Col 7: = (operator)
Line 1, Col 9: 5 (integer)
```
