

# Summary

## Introduction

[Notation](#)

## Language Constructs

- [Lexical structure](#)
  - [Keywords](#)
  - [Identifiers](#)
  - [Comments](#)
  - [Whitespace](#)
  - [Punctuation](#)
- Modules and source files
- [Items](#)
  - Use declarations
  - Functions
    - Implicit arguments
  - Type aliases
  - Structs
  - Enums
  - Constant items
  - Traits
  - Implementations
- Names
  - Paths
  - Namespaces
  - Visibility
  - [Prelude](#)
  - [Naming conventions](#)
- [Statements](#)
  - [Let statement](#)
  - [Assignment statement](#)
  - [Item statement](#)

- Expression statement
- Expressions
  - Literal expressions
  - Path expressions
  - Block expression
  - Operator expressions
    - Negation operators
    - Arithmetic and logical operators
    - Equality operators
    - Comparison operators
    - Boolean operators
    - Error propagation operator
  - Parenthesized expressions
  - Function calls
  - Method calls
  - Field access expressions
  - Tuple expressions
  - Array expressions
  - Struct expressions
  - If expressions
  - Match expressions
  - For loop expressions
  - Return expressions
- Patterns
- Type system
  - Types
    - Boolean types
    - Felt type
    - Integer types
    - String types
    - Unit type
    - Never type
    - Tuple types
    - Struct types
    - Enum types

- Array types
- Slice types
- Error type
- Generics
- Type layout
  - Packed structs
- Hints

## Language Semantics

- Memory model
- Constant evaluation
- Application binary interface
- Runtime

## Appendices

- [link:Full Grammar](#)