

Aiken New Functions

Here are some examples of newly developed functions and their code snippets. Aiken is a modern language, and the exact functions and syntax might vary with newer versions. For the most accurate and detailed examples, refer to the latest Aiken documentation or release notes.

1. Enhanced Type Inference

****Code Example:****

```
```aiken
// No explicit type needed; inferred from usage
let add = (a, b) => a + b
let result = add(5, 3) // result is inferred to be an Int
```
```

2. Pattern Matching

****Code Example:****

```
```aiken
// Pattern matching with a sum type
type Shape = Circle(Float) | Rectangle(Float, Float)

let area = (shape: Shape) =>
 match shape with
 | Circle(radius) => 3.14 * radius * radius
 | Rectangle(width, height) => width * height
```
```

3. Concurrency Primitives

****Code Example:****

```
```aiken
// Async/await example for asynchronous computation
let fetchData = async (url: String) =>
 let response = await http.get(url)
```

```

 response.body

// Using async function
let processData = async () =>
 let data = await fetchData("http://example.com/data")
 // Process data
...

```

## ### 4. Meta-Programming Support

```

Code Example:
```aiken
// Example of simple meta-programming using code generation
macro multiply(a: Int, b: Int) => a * b

let result = multiply(4, 5) // Expands to 4 * 5
...

```

5. Enhanced Error Handling

```

**Code Example:**
```aiken
// Custom error type and handling
type Error = NotFound | InvalidInput(String)

let handleError = (error: Error) =>
 match error with
 | NotFound => "Error: Not found"
 | InvalidInput(msg) => "Error: " ++ msg

let result = try
 // Code that might throw an error
 let value = someFunction()
 Ok(value)
catch e => Err(handleError(e))
...

```

## ### 6. Improved Standard Library

**\*\*Code Example:\*\***

```
``aiken
// Using new standard library functions
let sortedList = List.sort([3, 1, 4, 1, 5, 9]) // [1, 1, 3, 4, 5, 9]
``
```

### ### 7. Immutable Data Structures

**\*\*Code Example:\*\***

```
``aiken
// Working with immutable lists
let numbers = [1, 2, 3]
let newNumbers = numbers ++ [4, 5] // [1, 2, 3, 4, 5]
``
```

### ### 8. Optimized Compilation

**\*\*Code Example:\*\***

```
``aiken
// This feature is more about how code is compiled rather than specific syntax, but you might
// write more efficient code as follows:
let optimizedFunction = (x: Int) =>
 let y = x * x
 y + x
``
```

### ### 9. Interoperability with Other Languages

**\*\*Code Example:\*\***

```
``aiken
// Calling a C function (hypothetical example)
foreign import c_function "c_library.h" (Int) -> Int

let result = c_function(10) // Calls C function with 10
``
```

### ### 10. New Syntax Features

**\*\*Code Example:\*\***

```
``aiken
// Example of new syntax features
let greet = (name: String) => f"Hello, {name}!"
let message = greet("World") // "Hello, World!"
``
```

These code snippets reflect some of the new functions and features in Aiken