

# Syntra Language Specification

*A modern, human-readable programming language inspired by C++.*

## 1. Core Keywords

Concept	Keyword in Syntra	Example
Program start	begin	action begin() { ... }
Function definition	action	action compute() { ... }
Variable declaration	point, distance, name, flag	point x = 10;
Output	show	show "Hello!";
Input	get	get username;
If	when	when (x > 10) { ... }
Else	otherwise	otherwise { ... }
While	loop	loop (x < 5) { ... }
For	repeat	repeat (point i = 0; i < 10; i++) { ... }
Return	give	give result;
Include	use	use math;
Namespace	space	space core;
Comment	note	note This is a comment.

## 2. Data Types

Type		Meaning	Example
point		Integer number	point x = 10;

distance		Floating-point number	distance y = 12.75;
name		String (text)	name user = "Alice";
flag		Boolean (true/false)	flag active = true;

## 3. Syntax Examples

### Hello World

```
use core

action begin() {
    show "Welcome to Syntra!";
}
```

### Variables and Basic I/O

```
use core

action begin() {
    name username;
    point age;

    show "Enter your name: ";
    get username;

    show "Enter your age: ";
    get age;

    show "Hello, " + username + ". You are " + age + " years old.";
}
```

## Condition (If / Else)

```
action begin() {  
    point number = 10;  
  
    when (number > 5) {  
        show "The number is greater than 5.";  
    }  
    otherwise {  
        show "The number is small.";  
    }  
}
```

## For Loop (Repeat)

```
action begin() {  
    repeat (point i = 0; i < 3; i++) {  
        show "Iteration " + i;  
    }  
}
```

## While Loop

```
action begin() {  
    point count = 0;  
  
    loop (count < 5) {  
        show "Count: " + count;  
        count = count + 1;  
    }  
}
```

## Function Example

```
action add(point a, point b) {  
    give a + b;  
}
```

```
action begin() {  
    point result = add(3, 7);  
    show "Result = " + result;  
}
```

## Full Example Program

```
use core
```

```
action multiply(point a, point b) {  
    point result = a * b;  
    give result;  
}
```

```
action begin() {  
    point x = 4;  
    point y = 5;  
    point output = multiply(x, y);  
  
    when (output > 10) {  
        show "Large product: " + output;  
    }  
    otherwise {  
        show "Small product: " + output;  
    }  
}
```