

# Grammar Rules

---

## 1. Basic Output Commands

### 1.1 Ask

Prompts the user for **input**.

**Example:**

```
botms
CopyEdit
Ask "What is your name?"
```

---

### 1.2 Reply

Outputs a **fixed, conversational response**.

**Example:**

```
botms
CopyEdit
Reply "Hello, World!"
```

---

### 1.3 Print

Used for **system messages** or **debugging outputs**.

**Example:**

```
botms
CopyEdit
Print "Welcome to BOTMS!"
```

---

## 2. Conditional Statements

### 2.1 Structure

BOTMS uses **If**, **Else If**, and **Else** to perform **conditional checks**.

A **colon (:)** starts the block, with **indentation** defining nested code.

**Example:**

```
botms
CopyEdit
```

```
Ask "Do you like BOTMS?"
If input is "Yes":
    Reply "Great! BOTMS likes you too."
Else:
    Reply "No worries, BOTMS will win your heart soon!"
```

---

## 2.2 Multiline Conditions

Multiple lines under the same condition must be **indented consistently**.

---

## 2.3 Logical Operators

Use **and**, **or**, and **not** for **combining conditions**.

**Example:**

```
botms
CopyEdit
If age is greater than 18 and age is less than 60:
    Reply "You are eligible."
```

---

## 3. Loops

### 3.1 Repeat Loop

Repeats a block for a **fixed number of iterations**.

**Example:**

```
botms
CopyEdit
Repeat 3 times:
    Reply "Hello!"
```

---

### 3.2 Infinite Loop

Runs **forever** until explicitly **broken** using **Break**.

**Example:**

```
botms
CopyEdit
```

```
Repeat forever:
  Ask "Type something:"
  If input is "Stop":
    Reply "Loop ended."
  Break
```

---

### 3.3 For Loop

Loops through a **range of numbers** or **list items**.

**Example:**

```
botms
CopyEdit
For i from 1 to 5:
  Reply "Number: {i}"
```

---

### 3.4 Nested Loops

Loops can be **nested** with **separate indentations**.

---

## 4. Variables and Data Types

### 4.1 Declaration

Variables are assigned using **natural language**.

**Example:**

```
botms
CopyEdit
Set name as "Adi"
Set age as 25
```

---

### 4.2 Shortcut Assignment

BOTMS also supports **technical shortcuts**.

**Example:**

```
botms
CopyEdit
```

Set age = 23

---

#### 4.3 Supported Data Types

Type	Example	Meaning
Text	"Adi"	Words or sentences
Number	25	Integer or Float
Boolean	True or False	Logical values

---

### 5. Arithmetic Operations

#### 5.1 Supported Operators

Operation	English Style	Shortcut	Example
Addition	plus	+	Set sum as a plus b
Subtraction	minus	-	Set diff as a minus b
Multiplication	times	*	Set prod as a times b
Division	divided by	/	Set part as total divided by 4
Modulus	mod	%	Set remainder as a mod b
Exponentiation	to power	**	Set result as 2 to power 3
Floor Division	floor divided by	//	Set portion as total floor divided by parts

---

### 6. Custom Expressions – Connected Expressions™

Connected Expressions™ allow **natural language-based** computations.

#### 6.1 Supported Expressions

Expression	Meaning	Example	Output
half of	Divides by 2	Set x as half of 100	50

<b>square of</b>	Power of 2	Set y as square of 4	16
<b>root of</b>	Square root	Set z as root of 25	5
<b>mean of</b>	Average	Set avg as mean of 5, 10, 15	10
<b>% of</b>	Percentage	Set tax as 10% of total	10% of total

---

## 6.2 Connected Expressions Rule

- When **"of"** follows a keyword like **half** or **mean**, BOTMS automatically enters **Expression Mode**.
- Without **"of"**, the keyword is treated as a **variable name**.
- To force BOTMS to evaluate an expression, **wrap it in parentheses**.

### Example:

```
botms
CopyEdit
Set amount as 100
Set half as (half of amount)
```

---

## 7. Lists and Collections

BOTMS introduces a **Unified List System** for sequential and unique data collections.

### 7.1 List Creation

List Type	Example
Sequential	Set list1 as apple, banana, mango
Unique List	Set unique list as apple, banana, mango
Empty List	Set a as List

---

### 7.2 Adding Items

```
botms
CopyEdit
Add orange to list1
```

---

### 7.3 Removing Items

botms

CopyEdit

```
Remove mango from list1
```

```
Remove item at 2 from list1
```

---

### 7.4 Accessing Items

Command	Example	Output
Get by Index	Get 2 from list1	banana
Get First	Get first from list1	apple
Get Random	Get random from list1	mango or banana

---

### 7.5 Looping Through Lists

botms

CopyEdit

```
Loop through list1
```

```
    Reply item
```

---

### 7.6 Fixed Size Lists

BOTMS allows lists to be created with a **fixed size** using the **List of size** keyword. Once set, the list cannot grow beyond the defined size.

Example:

botms

CopyEdit

```
Set numbers as List of size 3
```

```
Add 10, 20, 30 to numbers
```

```
Add 40 to numbers
```

**Output:**

CopyEdit

BOTMS Error: List is Locked

---

## 7.7 Nesting Lists 🌳

BOTMS supports lists inside lists, making it possible to store collections within collections naturally.

Example:

```
botms
CopyEdit
Set parent as List
Add apple, banana to parent
Set child as List
Add mango, orange to child
Add child to parent
Loop through parent
  Reply item
```

**Output:**

```
CopyEdit
apple
banana
[mango, orange]
```

---

## 7.8 BOTMS Lock™ 🔒

When a list is declared with **List of size**, BOTMS automatically activates the **BOTMS Lock™**, preventing both additions and removals beyond the fixed limit.

If a locked list exceeds its size, BOTMS throws this error:

```
CopyEdit
BOTMS Error: List is Locked
```

---

## 8. Functions and Modular Design

BOTMS uses **Modular Code Blocks™** to make code **reusable** and **cleaner**.

---

## 8.1 Defining Functions

Functions are defined with the keyword **Function**.

botms

CopyEdit

```
Function greet(name):  
    Reply "Hello, {name}!"
```

---

## 8.2 Calling Functions

botms

CopyEdit

```
Call greet("Adi")
```

**Output:**

CopyEdit

```
Hello, Adi!
```

---

## 8.3 Returning Values

botms

CopyEdit

```
Function square(x):  
    Return x to power 2
```

```
Set result as Call square(5)
```

```
Print result
```

**Output:**

CopyEdit

```
25
```

---

## 8.4 Function with Multiple Parameters

botms

CopyEdit

```
Function add(x, y):  
    Return x plus y
```



```
Set sum as Call add(10, 20)
Print sum
```

**Output:**

CopyEdit  
30

---

## 8.5 Nesting Functions 🌳

Functions can call other functions inside:

botms  
CopyEdit

```
Function square(x):
    Return x to power 2

Function cube(x):
    Set temp as Call square(x)
    Return temp times x

Set result as Call cube(3)
Print result
```

**Output:**

CopyEdit  
27

---

## 8.6 Inline Functions™

BOTMS supports **Inline Functions™** for single-line quick functions.

botms  
CopyEdit

```
Inline Function double(x): x times 2
Set result as Call double(10)
Print result
```

**Output:**

## 8.7 Modular Design

Modular Design encourages writing **independent code blocks** that can be reused across different parts of the code without repetition.

---

How BOTMS Modular Design Works:

- Each **Function** is treated as a standalone **Module**.
  - Modules can be **imported** or **called** anywhere.
  - Functions automatically become **BOTMS Modules™** once defined.
- 

Example

### Without Modular Design:

```
botms
CopyEdit
CopyEdit
Ask "Enter a number:"
Set square as input to power 2
Print square

Ask "Enter another number:"
Set square2 as input to power 2
Print square2
```

---

### With Modular Design:

```
botms
CopyEdit
CopyEdit
Function square(n):
    Return n to power 2

Ask "Enter a number:"
Set result as Call square(input)
```

```
Print result
```

```
Ask "Enter another number:"  
Set result as Call square(input)  
Print result
```


---

## Benefits of Modular Design in BOTMS

Feature	Without Modular Design	With Modular Design
Code Reuse	✗ No	✓ Yes
Clean Code	✗ No	✓ Yes
Easy Maintenance	✗ No	✓ Yes
Nested Functions	✗ No	✓ Yes

---

## 9. Error Handling

BOTMS uses **Human-Friendly Error Messages**  so even beginners will know what's wrong without searching on Google.

### 9.1 BOTMS Try-Catch™

BOTMS has its own custom **Try-Catch™ System** to catch errors without breaking the whole code.

**Example:**

```
botms  
CopyEdit  
Try:  
    Ask "Enter a number:"  
    Set x as input divided by 0  
    Print x  
Catch:  
    Reply "Oops! Something went wrong."
```


**Output:**

```
copyedit
```

CopyEdit  
Oops! Something went wrong.

---

## 9.2 BOTMS Built-in Error Types

Error Name	Meaning	Example
<b>BOTMS Type Error</b>	Wrong data type	Trying to divide text by a number
<b>BOTMS Zero Error</b>	Divide by zero	10 divided by 0
<b>BOTMS Lock Error</b> 	List size exceeded	Adding more items to a locked list
<b>BOTMS Not Found Error</b>	Variable not defined	Trying to <b>Print</b> an undefined variable
<b>BOTMS Expression Error</b>	Wrong custom expression	Writing <b>square of apple</b>

---

## 9.3 BOTMS Error Custom Messages

You can create your own error messages using the **Throw** keyword.

### Example:

```
botms
CopyEdit
Try:
    Ask "Enter your age:"
    If input is not number:
        Throw "Age must be a number!"
Catch:
    Reply error
```

### Output:

```
copyedit
CopyEdit
Age must be a number!
```

---

## 9.4 New Keywords Added in Error Handling Section

Keyword	Meaning	Type	Example
---------	---------	------	---------

<b>Try</b>	Starts an error-catching block	<b>Inline Command</b>	Try:
<b>Catch</b>	Executes if an error happens	<b>Inline Command</b>	Catch:
<b>Throw</b>	Manually raises an error	<b>Inline Function™</b>	Throw "Invalid data"
<b>error</b>	Stores the last error message	<b>BOTMS Variable™</b>	Reply error