


1. Datatypes (Updated)

plaintext


 Copy

 Edit


```
integer    → for whole numbers (like 10, -3)
decimal    → for decimal numbers (like 3.14)
string     → for text (like "Rashika")
```

2. Variable Assignment (Updated)

Use `===` instead of `=` for assignment.

 **Syntax:**

plaintext

 Copy

 Edit

```
integer a === 5;
decimal pi === 3.14;
string name === "Rashika";
```

+ 3. Arithmetic Expressions (Same)

Supports:

plaintext

Copy

Edit

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

✓ Expressions stay the same, only variables use `==`.

plaintext


Copy

Edit

```
integer sum == a + b;  
decimal area == 3.14 * r * r;
```

4. Print Statements (Same)

plaintext


 Copy

 Edit


```
print a;  
print "Hello Rashika!";
```

5. Conditional Statements (Updated)

Use quoted conditions in `if` and `else` blocks.

 **Syntax:**

plaintext

 Copy


 Edit

```
if "a > b" {  
    print "a is greater";  
} else {  
    print "b is greater";  
}
```

6. Loops (Updated)

while Loop:

plaintext


 Copy

 Edit

```
integer i == 0;
while "i < 5" {
    print i;
    i == i + 1;
}
```

for Loop:

plaintext

 Copy

 Edit

```
for "integer i == 0, i < 5, i++" {
    print i;
}
```

For loop uses one quoted string for initialization, condition, and increment (like C-style but quoted).

Feature	Example
Datatypes	<code>integer, decimal, string</code>
Assignment	<code>integer x === 10;</code>
Arithmetic	<code>x + y * 2</code>
Print	<code>print x;</code>
If Condition	<code>if "x > 10" { ... }</code>
Else	<code>else { ... }</code>
While Loop	<code>while "x < 5" { ... }</code>
For Loop	<code>for "integer i === 0, i < 5, i++" { ... }</code>