



meaning of name
↳ identifier (name)

int(a) → not identifier
 ↳ variable ✓
 abc → variable ✓
 ↳ not identifier

Local

```
#include <stdio.h>

int total_students = 50;
float avg_marks = 85.6;

void displayResults() {
    printf("Total Students: %d\n", total_students);
    printf("Average Marks: %.2f\n", avg_marks);
}

int main() {
    int a;
    displayResults();
    return 0;
}
```

Global Variable

function

variable (∞) identifier

Identifier

local variable

Rules :

① variable can start letter's (or) underscore (_)
valid : _age, age_, age, bank_id
Invalid : 1age, #temp, @age (all marked with a red X)

② can contain letters, digits and underscore's
valid : age1, Bank_id1
Invalid : 1age, Bank-id (both marked with a red X)

③ Case Sensitive
valid : age
↳ while calling age

Invalid age
↳ while calling Age (marked with a red X)

④ No keywords are allowed

⑤ variable name length should not exceed the length more than 31 characters

⑥ must declare before use
↳ mandatory declare

```
[int a;  
char ch;]
```

C language

↳ 32 keywords

int
char
float
double

→ data type
(not) keywords