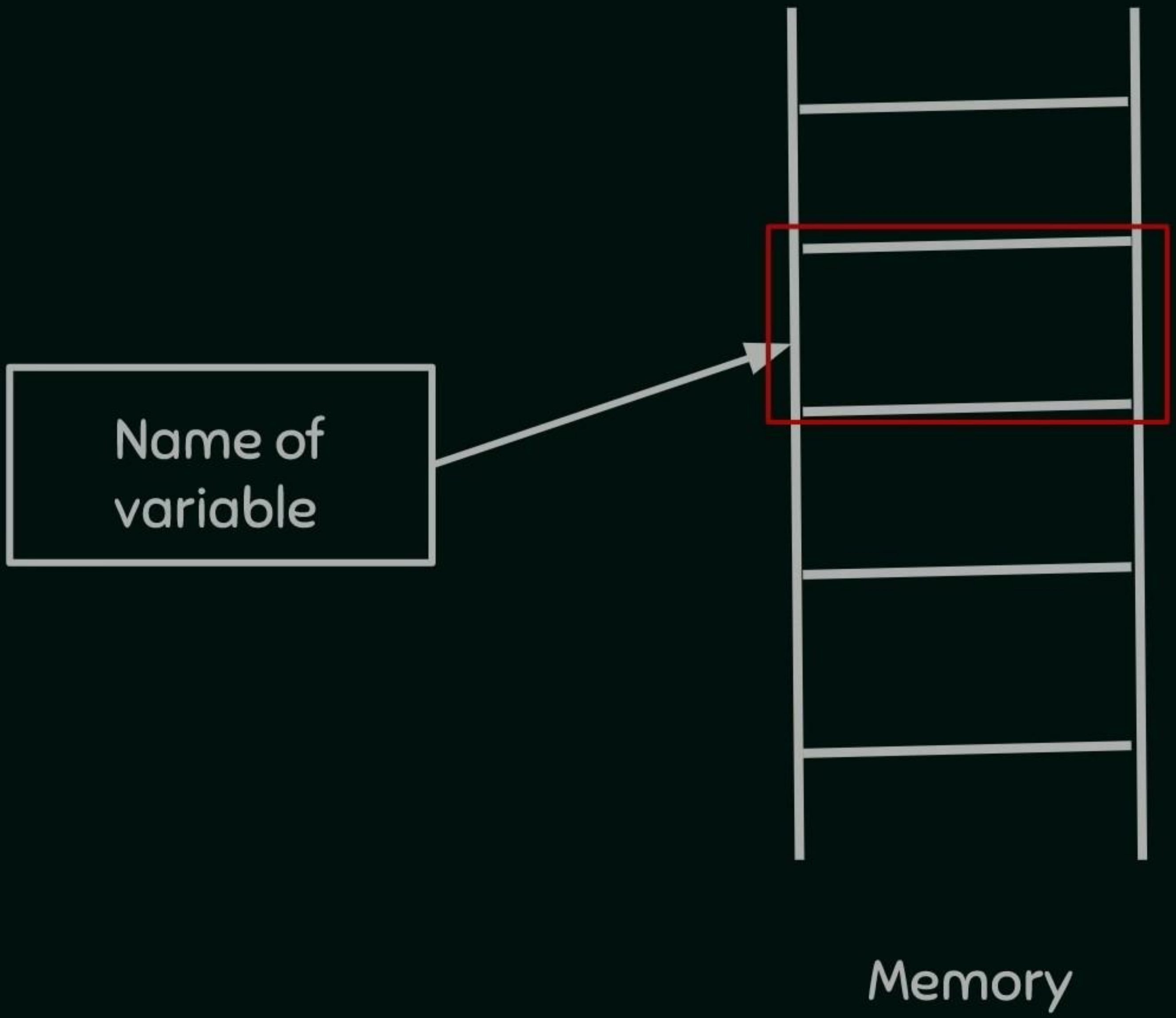

Name of variable
Memory

DECLARATION

Declaration: announcing the properties of variable to the compiler

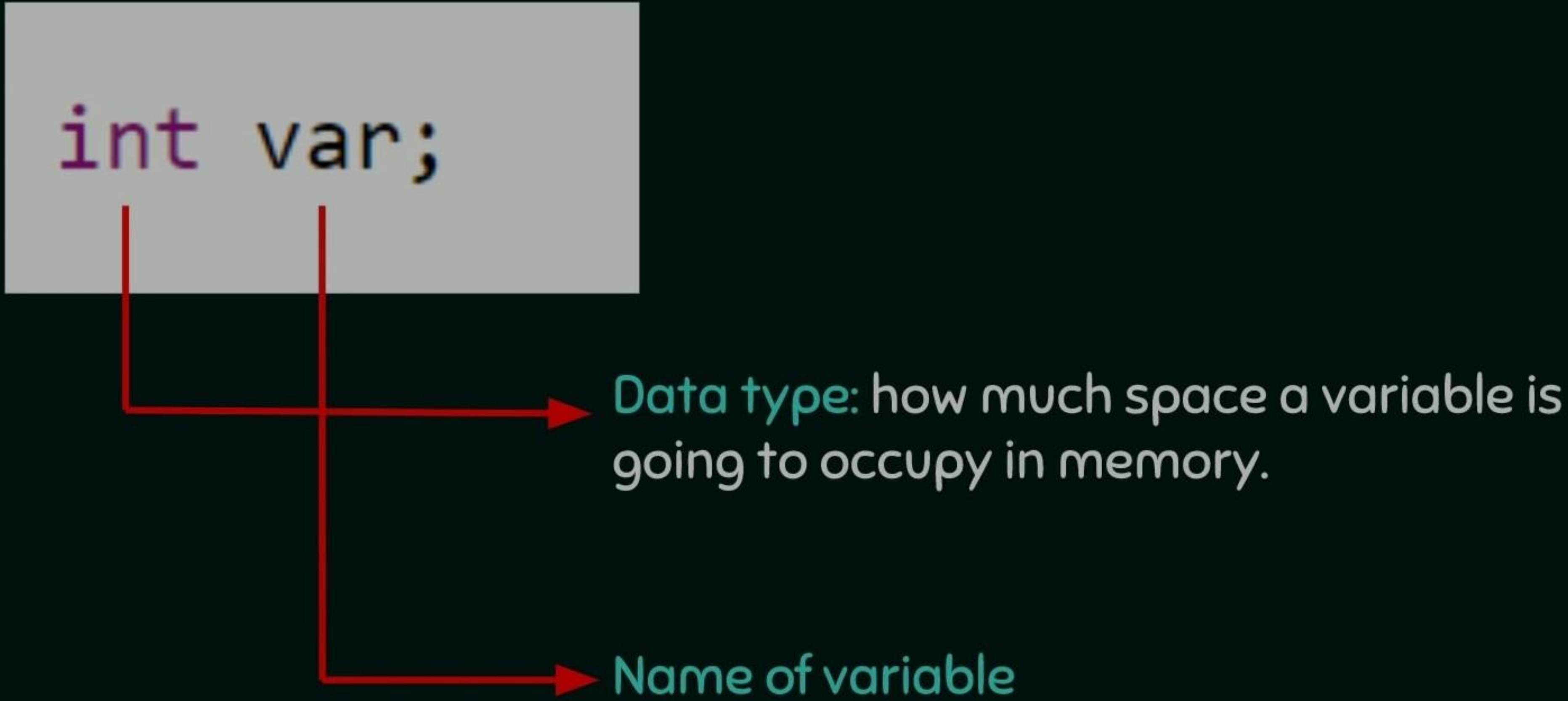
Properties:

1. Size of the variable
2. Name of the variable

Definition

Definition: allocating memory to a variable

Most of the time declaration and definition will be done at the same time.

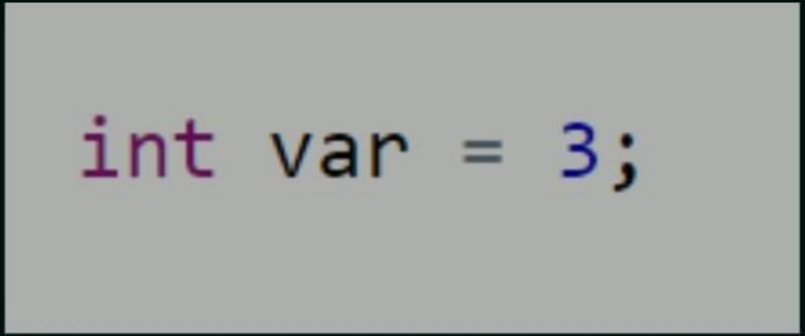


```
int var;
```

Data type: how much space a variable is going to occupy in memory.

Name of variable

Initialization



```
int var = 3;
```

Variable name: composed of letters or combination of letters (both upercase and lowercase) and digits.

Rule #1: don’t start variable name with digit.

globe ✓

2globe ✗

Rule #2: begining with underscore is valid but not recommended.

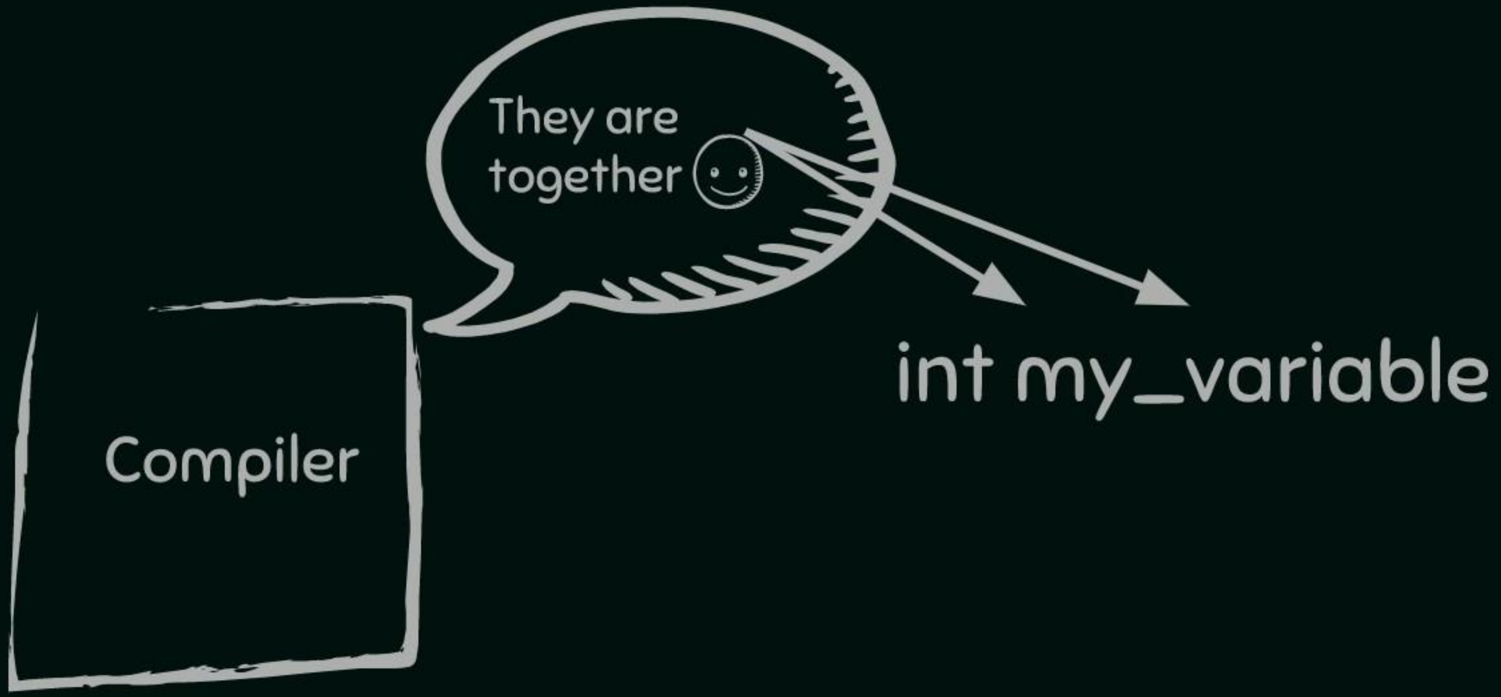
_ (underscore) is treated as letter.

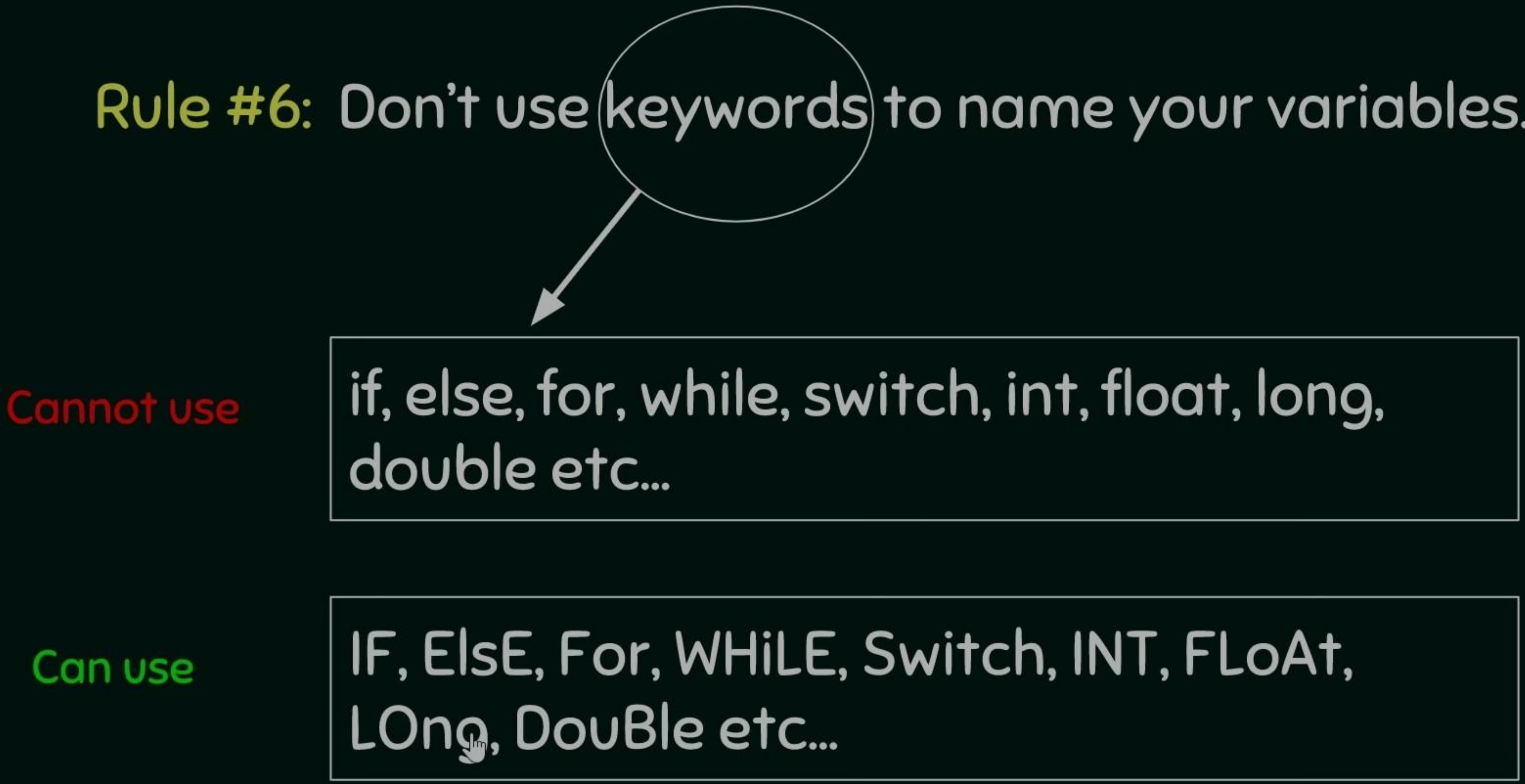
Rule #3: C langauge is case sensitive. Uppercase letters are different from lowercase letters.

var, Var, VaR, vAR, vAr, VAR all are different.

Rule #4: special characters (@, #, %, ^, &, *...) not allowed in the name of variable.

Rule #5: Blanks or white spaces not allowed.





Rule #6: Don't use keywords to name your variables.

Cannot use

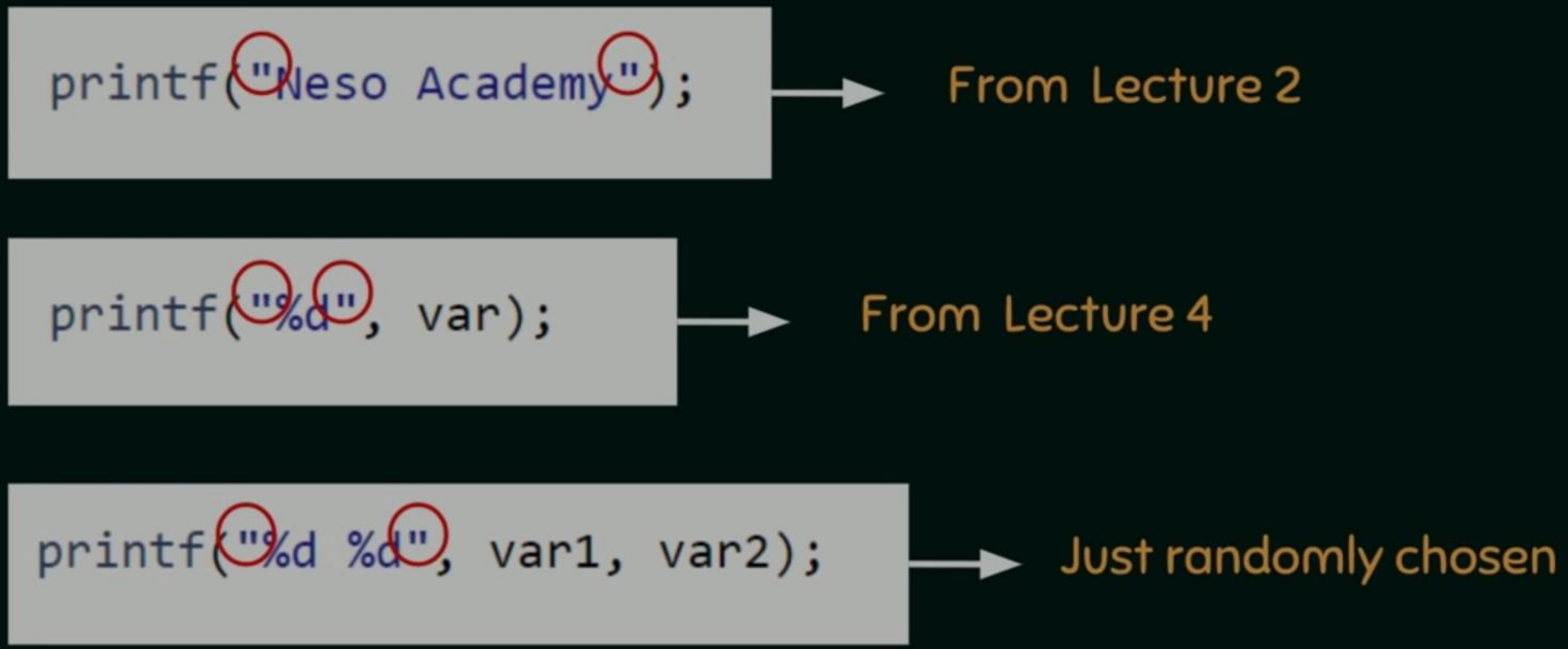
if, else, for, while, switch, int, float, long, double etc...

Can use

IF, ElsE, For, WHiLE, Switch, INT, FLoAt, LOng, DouBle etc...

Do not use loooooonnngggg... names for your variables.

Recall these code snippets



```
printf("Neso Academy");
```

From Lecture 2

```
printf("%d", var);
```

From Lecture 4

```
printf("%d %d", var1, var2);
```

Just randomly chosen

%d is a placeholder for variable

"d" means decimal