**Walrus Operator (:=)**
Walrus operator is introduced in python 3.8 version.
Walrus operator is also called assignment expression operator.

| | |
|---|---|
| = | -- assignment operator |
| := | -- walrus operator or assignment expression |
| == | -- relational operator or equal |

**Example:**
```
>>> a=10+20+b=10-5
SyntaxError: cannot assign to expression
>>> a=10
>>> b:=20
SyntaxError: invalid syntax
>>> c=(a:=10+20)-(b:=10-5)
>>> print(a,b,c)
30 5 25
>>> c=(a=10+20)-(b=10-5)
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
```

**Assignment Operators or Update Operators or Augmented assignment statements**

One operator perform two operations
1. Binary Operation
2. Assignment

| | |
|---|---|
| += | A=10<br>A=A+5 → A+=5 |
| -= | A=10<br>A=A-2 → A-=2 |
| *= | A=5<br>A=A*3 → A*=3 |
| /= | A=6<br>A=A/2 → A/=2 |
| //= | A=4<br>A=A//3 → A//=3 |
| %= | A=5<br>A=A%2 → A%=2 |
| **= | A=5 |

| | |
|---|---|
| | A=A**2 → A**=2 |
| >>= | A=6<br>A=A>>2 → A>>=2 |
| <<= | A=10<br>A=A<<3 → A<<=3 |
| &= | A=10<br>B=5<br>A=A&B → A&=B |
| \|= | A=10<br>B=5<br>A=A\|B → A\|=B |
| ^= | A=10<br>B=5<br>A=A^B → A^=B |

1. Arithmetic Operators → +,-,*,/,//,%,**
2. Relational Operators → >,<,>=,<=,!=,== (Comparison Operators)
3. Logical Operators → and, or, not
4. Conditional Operators → if,else
5. Membership Operators → in, not in
6. Identity Operators → is, is not
7. Bitwise Operators → <<,>>,&,|,^,~
8. Walrus Operator → :=
9. Assignment Operator → +=,-=,*=,/=,%=,//=,**=,>>=,<<=,&=,|=,^=

**Control Statements or Control Structures**
Control statements are used to control the flow of execution of program.
Default flow of execution of program is sequential (line by line).
Control statements are classified into 3 categories

1. Conditional Control Statements
    a. If statement
    b. match statement

2. Looping Control Statements
    a. While
    b. for
3. Branching Statements
    a. Break
    b. Continue
    c. Return
    d. Pass

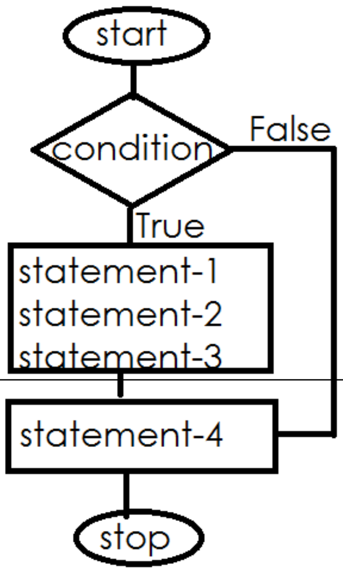
**Conditional Control Statements**
Conditional control statements or conditional statements are used to execute block of statements based on condition or test.

1. If
2. match (python 3.10 version)

**if statement**
if is a conditional control statement. This statement is used to execute block of statements based on condition.

1. Simple if
2. If..else
3. If..elif..else (if..else ladder)
4. Nested if

**simple if**
if without else is called simple if.

| | |
|---|---|
| **Syntax**<br><br>**if <condition>/<test>:**<br>    **statement-1**<br>    **statement-2**<br>    **statement-3**<br>**statement-4**<br><br>if condition is True, PVM executes |  start<br>condition<br>False<br>True<br>statement-1<br>statement-2<br>statement-3<br>statement-4<br>stop |

| | |
|---|---|
| statement-1,statement-2,statement-3 and statement-4<br>if condition is False, PVM execute statement-4 | |

Python does not allow empty block.

**Example:**

```
if 10>5:
   print("Hello")
print("Bye")

#Output:  Hello Bye
if 10<5:
    print("Hello")
print("Bye")

# Output: Bye

if 10<5 or 10>5:
    print("Hello")
print("Bye")

# Output: Hello Bye

if True:
    print("True")
print("False")

# Output: True False
```

**Output:**

```
Hello
Bye
Bye
Hello
Bye
True
False
```

**What is indentation in python?**
Indentation refers to the spaces at the beginning of a code line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code.

Default indent is 4 spaces.

**What is pass statement or keyword?**
The pass statement is used as a placeholder for future code. When the pass statement is executed, nothing happens, but you avoid getting an error when empty code is not allowed.
It is used to create empty blocks.