# Python Keywords

1. Keywords are Reserved Words in Python
2. We can't use keywords as Variables Name, Functions Name, or any other Identifiers
3. Keywords are Case Sensitive

In [3]:
```
# To Import Keywords
import keyword
```

In [4]:
```
#To Check Keywords
print(keyword.kwlist)
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'fo
r', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'no
t', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

In [5]:
```
#To Check Total Keywords
len(keyword.kwlist)
```

Out[5]: 35

# 2.Python Identifier

1. Identifier is name that given to the entites like class,functions, and variables, etc
2. it helps the identify the one entity to another entity

## Rules for Declare the Identifier

- Identifiers are casesensitive
- Identifiers are can't start with Digit
- Special Characters are not allowed, Except Underscore _
- Keywords are the not allowed as Identifier
- No length Limit as declare as Identifier

# 3.Python Comments

- Comments are the lines, that exits in computer programms that can ignored by the **Compilers and Interpreters**

- By Using Comments, That Program is easily understand and More Readble for humans
- In Python, # is used to write the **Comments**

**Compilers :**

1. Compilers are the Run the Program and convers it the machine code at a time
2. Compilers are take large amount of time to run the code
3. Ex: C,C++, Java

**Interpreter :**

1. Interpretes run the program line by liune or statement wise at a time.
2. Interpreter take the very less time to riun the program
3. Ex : Python, Java Script,Ruby

### Multiline Comments

- If we have required multipule lines commments, we use hash (#), in each line

In [16]:
```
# This is Manikanta
# Preparing the Data Science
# This will Take 180 target
```

In [17]:
```
""" This is Manikanta,
Preparing the Data Science
This will take 180 days target """
```

Out[17]:
```
' This is Manikanta,\nPreparing the Data Science\nThis will take 180 days t
arget '
```

# 4.Python Statements

- Instructions that a python interpreter can execute are called the Statements

In [20]:
```
# Single Statements
a = 10
```

In [21]:
```
# Multiple Lines Statements
a = 1+2+3+4+5+ \
    6+7+\
    8+9
print(a)
```

In [22]:
```
#By using paranthasis
a = (1+2+3+4+5+ \
    6+7+\
    8+9)
print(a)
```

```
45
```

In [23]:
```
#Multiple Statements in SingleLine
a=10 ; b=20; c=30
print(a,b,c)
```

```
10 20 30
```

# 5.Variables

- A Variable is a location in memory used to store the some value or
- The Rules for writing variables are same as the writing the identifiers
- We Don't even to declare the type of variable. This is handled internally according to the type of value we assign to the value.

## Variable Assignments

- we use the = assignment operator to assign the values into variables

In [28]:
```
x = 11
y = 22.22
z = "Manikanta"
```

In [29]:
```
# Assign the Values in Same Line
x,y,z = 10,20.20," Manikanta"
print(x,y,z)
```

```
10 20.2  Manikanta
```

In [30]:
```
#Assign the same value to multiple Variables
a=b=c=d= "Mani"
print(a)
print(b)
print(c)
print(d)
```

```
Mani
Mani
Mani
Mani
```

In [31]:
```
#### Storage Locations
x = 3
id(x)
```

Out[31]: 140711834888696

In [32]:
```
y=3
id(y)
```

Out[32]: 140711834888696

In [33]:
```
y=2
id(y)
```

Out[33]: 140711834888664

# 6.Data Types

- Every Value in Python has a data type
- we use the **type()** function to know witch class a variable or value
- Interger - int
- Float -float
- Complex - complex
- Boolean - bool

In [36]:
```
a=10
print(a,"The Type of a is:", type(a))
```

```
10 The Type of a is: <class 'int'>
```

In [37]:
```
b=10.10
print(b,"The Type of b is:", type(b))
```

```
10.1 The Type of b is: <class 'float'>
```

In [38]:
```
c=10
print(c,"The Type of c is:", type(c))
```

```
10 The Type of c is: <class 'int'>
```

In [39]:
```
d= True
print(d,"The Type of d is:",type(d))
```

```
True The Type of d is: <class 'bool'>
```

## Python Strings

- String is the Sequence of Unicode Characters
- we can use single quotes or double quotes ' ' or " "
- Strings can be indexed and Start from 0.

In [42]:
```
m = "This is Manikanta"
print(type(m))
```

```
<class 'str'>
```

### Type Casting or Datatype Convertion

In [44]:
```
#Integer to Float
b = float(100)
print(b)
type(b)
```

```
100.0
```

Out[44]: float

In [45]:
```
#Flaot to Integer
a = int(100.100)
print(a)
type(a)
```

```
100
```

Out[45]: int

In [46]:
```
#Integer to String
aa = str(100)
print(aa)
type(aa)
```

```
100
```

Out[46]: str

In [47]:
```
#Integer to String with Combination of Letters
bb= str(100PPP)
print(bb)
type(bb)
```

```
  Cell In[47], line 2
    bb= str(100PPP)
              ^
SyntaxError: invalid decimal literal
```

# 7.Python Output & Input

### Output

In [ ]:
```
#Use Print() function to get output data
print("Hello World")
```

In [ ]:
```
a=10
print("The Value of a is:", a)
```

In [ ]:
```
#Multiple Statements in the Single Line
a =10; b=20
print("The value of a is {} and b is {}".format(a,b))
```

In [ ]:
```
#Multiple Statements in Single Line and Specify the Index
a= 100;b=200
print("The Value of a is {0} and b is {1}".format(a,b))
```

In [ ]:
```
#Multiple Statements in Single Line and Specify the Index
a= 100;b=200
print("The Value of b is {1} and a is {0}".format(a,b))
```

In [ ]:
```
#ByUsing the Keyword Argument
print("Hello My {Name}, Age is {Age}".format(Name="Manikanta",Age=29))
```

## Input

In [ ]:
```
# We have take input from the user, then we take input() function used
num = input("Enter the Value")
print(num)
```

In [ ]:

In [ ]: