28th

In [3]:
```
v=1
```

In [4]:
```
v
```

Out[4]: 1

In [5]:
```
id(v)
```

Out[5]: 140724143895464

In [6]:
```
nit=2
```

In [7]:
```
nit
```

Out[7]: 2

In [8]:
```
8 = nit
```

```
  Cell In[8], line 1
    8 = nit
    ^
SyntaxError: cannot assign to literal here. Maybe you meant '==' instead of '='?
```

In [9]:
```
v8 = nit
```

In [10]:
```
v8
```

Out[10]: 2

In [11]:
```
8v = 6
```

```
  Cell In[11], line 1
    8v = 6
    ^
SyntaxError: invalid decimal literal
```

In [12]:
```
nit$ = 10
```

```
  Cell In[12], line 1
    nit$ = 10
       ^
SyntaxError: invalid syntax
```

In [13]:
```
gvk = 9
gvk
```

Out[13]: 9

In [15]:
```
a='vamshi'
b='krishna'
```

In [20]:
```
int.__add__(a,b)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[20], line 1
----> 1 int.__add__(a,b)

TypeError: descriptor '__add__' requires a 'int' object but received a 'str'
```

In [19]:
```
str.__add__(a,b)
```

Out[19]:
```
'vamshikrishna'
```

In [21]:
```
import keyword
keyword.kwlist
```

Out[21]:
```
['False',
 'None',
 'True',
 'and',
 'as',
 'assert',
 'async',
 'await',
 'break',
 'class',
 'continue',
 'def',
 'del',
 'elif',
 'else',
 'except',
 'finally',
 'for',
 'from',
 'global',
 'if',
 'import',
 'in',
 'is',
 'lambda',
 'nonlocal',
 'not',
 'or',
 'pass',
 'raise',
 'return',
 'try',
 'while',
 'with',
 'yield']
```

In [22]:
```
import keyword
len(keyword.kwlist)
```

Out[22]: 35

In [23]:
```
def = 29
def
```

```
  Cell In[23], line 1
    def = 29
        ^
SyntaxError: invalid syntax
```

In [24]:
```
DEF = 29
DEF
```

Out[24]: 29

Rules of Python Variables:

- case sensitive
- keywords cannot be used as variables
- special characters not allowed except _ (underscore)
- Does not start with digit but ends with digit

pythob variable = value value is also called as datatypes Example:

- int
- float
- Boolean
- Complex
- string

In [25]:
```
3+1
2+1
```

Out[25]: 3

In [26]:
```
print(3+1)
print(2+1)
```

```
4
3
```

In [ ]: