Find output:

```
a,b,c = 25, 10.8, 'Hyd'
print(a,b,c, end='---')  — # 25---10.8---Hyd
print(a,b,c, sep=':::')  — # 25:::10.8:::Hyd
print(a,b,c, sep=':::', end='\t\t\t')  25:::<tab>10.8:::<tab>
                                       Hyd:::<tab>
print(a,b,c) # 25 10.8 Hyd
```

Find output:

```
print('Hyd')   — # Hyd
print()        — # Returns None
print('Sec')   — # Sec
print()        — # return None
print('cyb')   — # cyb
```

Find outputs:

```
l = [10,20,30,40]
t = (10,20,30,40)
s = {10,20,30,40}
print(l,t,s)   —— #  [10,20,30,40]
                     (10,20,30,40)
                     {10,20,30,40}
```

Find outpts:

```
a = 25
b = '%.f' % a   — # 25.000000
    format  obj
print(b)
print(type(b))  — # <class 'str'>
x = 10.8        — # 10
y = '%.d' % x   — # <class 'str'>
print(y)
print(type(y))  — [10,20,15,11]
m = [10,20,15,18]
n = '%.s' % m
```

Find output

```
a = 10.9274
print ('%8.2f' %a)
```

<3 spaces > 10.93

```
print ('%9.1f' %a)      — # <1space> 10.92
print ('%.10.2f' %a)    — #
print ('%2f' %a)        — <8space>
print ('%6f', %a)       — <4space>
print ('%f' %a)         —
```

Find outputs

```
a= [10,20,30,40]    — # [10,20,30,40]
print ('%s' %a)      — # [10,20,30,40]
print ('%s' a)       — # syntax Error
print ('%s', %a)     — # syntax error
print ('%l' %a)      — # Syntax Error.
print (a)            — # [10,20,30,40]
```

How to print number of keywords - print(len(keyword.kwlist))
How to print type of kwlist - print(type(keyword.kwlist))
print(kwlist) -# prints all keywords.

```
from mod1 import *
from mod2 import *
print(x)
print(y)
```

```
import mod1, mod2
print(mod1.x)
print(mod2.y)
```

```
# input()
    x = input('Enter input : ') # Rama Roo
                                  ↳ always reads as string
    print(type(x))
    print(x)
# x = int(input('Enter input'))
    - float(input("Enter input)
```

eval():

```
print(eval('25')) — # 25
print(eval('10.8')) — # 10.8
print(eval('False')) — # False
print(eval('3+4j')) — # 3+4j
print(eval('Hyd')) — # Hyd
print(eval(" 'Hyd' ")) — 'Hyd'
print(eval("3+4*5")) — # 23
print(eval('[10,20,15,18]')) — [10,20,15,18]
print(eval('{10,20,15,18,20,12,18}')) — {10,20,15,18,20,12,18}
print(eval('(10,20,30)')) — # (10,20,30)
print(eval("{10:'Hyd', 10:'Sec'}")) — {10:'Sec'}
print(eval(4+5)) — # 9
```

Find output:

```
print (eval(" 'hyd' "))  ——  # hyd
hyd = 'sec'  ——  # sec is assigned
print (eval ('hyd'))  ——  # sec
sec = '25'  ——  # 25
print (eval('sec'))  ——  # 25
print(eval(sec))  ——  # 10.8 as assigned
cyb = 10.8
print (eval ('cyb'))  ——  # 10.8
print(eval (cyb))  ——  # Error.
```

Find output:

```
print (bool('False'))  ——  # True
print (eval('False'))  ——  # False.
print (bool(''))  ——  # False
print (eval(' '' '))  ——
print (eval(''))  ——  # Empty string
print (eval (' '' '))
print (eval (' '))
```

Sep argument

```
a, b, c = 25, 10.8, 'Hyd'
print (a,b,c, sep = ';')  ——  # 25;10.8;Hyd.
print (a,b,c, sep = '\t')  ——  # 25 <tab> 10.8<tab> Hyd <tab>
print (a,b,c, sep = '---')  ——  # 25---10.8---Hyd
print (a,b,c)  ——  # 25 10.8 Hyd.
print (a,b,c, separator = ':')  # 25 : 10.8 : Hyd.
```