# Integer

print("\*\*\*\*\*\*\*\* Integer \*\*\*\*\*\*\*\*\*")

print("{}".format(10)) # Do not write space between {}

print("{} {}".format(10, 20))

print("{0}".format(10))

print("{0} {1}".format(10, 20))

print("{1} {0}".format(10, 20))

print("{num1}".format(num1=10))

print("{num1} {num2}".format(num1=10, num2=20))

print("{num2} {num1}".format(num1=10, num2=20))

Float

print("\*\*\*\*\*\*\*\* Float \*\*\*\*\*\*\*\*\*")

print("{}".format(10.56))

print("{} {}".format(10.56, 20.42))

print("{0}".format(10.56))

print("{0} {1}".format(10.56, 20.42))

print("{1} {0}".format(10.56, 20.42))

print("{num1}".format(num1=10.56))

print("{num1} {num2}".format(num1=10.56, num2=20.42))

print("{num2} {num1}".format(num1=10.56, num2=20.42))

#String

print("\*\*\*\*\*\*\*\* String \*\*\*\*\*\*\*\*\*")

print("{}".format("apple "))

print("{} {}".format("apple", "banana"))

print("{0}".format("apple"))

print("{0} {1}".format("apple", "banana"))

print("{1} {0}".format("apple", "banana"))

print("{str1}".format(str1="apple"))

print("{str1} {str2}".format(str1="apple", str2="banana"))

print("{str2} {str1}".format(str1="apple", str2="banana"))

# Integer and String

print("Hello My Name is {}".format("alex"))

print("{} {}".format(10, "banana"))

print("{0} {1}".format(10, "banana"))

print("{1} {0}".format(10, "banana"))

print("{num1} {str1}".format(num1=10, str1="banana"))

print("{str1} {num1}".format(num1=10, str1="banana"))

Comma as thousand Separator

print("{:,}".format(1234567890))

#variable

name = "Rahul"

age= 62

print("My name is {} and age {}".format(name, age))

# Expressing a Percentage

a = 50

b = 3

print("{:.2%}".format(a/b))

# Accessing arguments� items

value = (10, 20)

print("{0[0]} {0[1]}".format(value))

# Format with Dict

data1 = {'rahul': 2000, 'sonam': 3000}

print("{0[rahul]:d} {0[sonam]:d}".format(data1))

# Format with Dict

data2 = {'rahul': 2000, 'sonam': 3000}

print("{d[rahul]:d} {d[sonam]:d}".format(d=data2))

# Accessing arguments by name:

data3 = {'rahul': 2000, 'sonam': 3000}

print("{rahul} {sonam}".format(\*\*data3))

# \*\* is a format parameter (minimum field width)

# Integer

print("\*\*\*\*\*\*\*\* Integer \*\*\*\*\*\*\*\*")

print("{}".format(15)) # Printing as string 15

print("{:d}".format(15))

print("{0:d}".format(15))

print("{num:d}".format(num=15))

print("{num:5d}".format(num=15))

print("{num:05d}".format(num=15))

print("{num:+5d}".format(num=15))

print("{num:<5d}".format(num=15)) # Left align

print("{num:\*<5d}".format(num=15)) # Left align with fill

print("{num:\*>5d}".format(num=15)) # Right align with fill

print("{num:^5d}".format(num=15)) # Center align

print("{num:\*^5d}".format(num=15)) # Center align with fill

Float

print("\*\*\*\*\*\*\*\* Float \*\*\*\*\*\*\*\*")

print("{}".format(15.65)) # Printing as string 15.65

print("{:f}".format(15.65))

print("{0:f}".format(15.65))

print("{num:f}".format(num=15.65))

print("{num:8f}".format(num=15.65)) # 15.650000

print("{num:8.3f}".format(num=15.65))

print("{num:+8.2f}".format(num=15.65))

print("{num:<8.2f}".format(num=15.65)) # Left align

print("{num:\*<8.2f}".format(num=15.65)) # Left align with fill

print("{num:\*>8.2f}".format(num=15.65)) # Right align with fill

print("{num:^8.2f}".format(num=15.65)) # Center align

print("{num:\*^8.2f}".format(num=15.65)) # Center align with fill

String

print("\*\*\*\*\*\*\*\* String \*\*\*\*\*\*\*\*")

print("{:8s}".format("apple"))

print("{:<8}".format("apple"))

print("{:\*<8}".format("apple"))

print("{:>8}".format("apple"))

print("{:\*>8s}".format("apple"))

print("{:^8s}".format("apple"))

print("{:\*^8s}".format("apple"))

print("\*\*\*\*\*\*\*\* Truncating String \*\*\*\*\*\*\*\*")

print("{:.3s}".format("ojas"))

print("{:8.3s}".format("ojas"))

print("{:\*<8.3s}".format("ojas"))

print("{:>8.3s}".format("ojas"))

print("{:\*>8.3s}".format("ojas"))

print("{:^8.3s}".format("ojas"))

print("{:\*^8.3s}".format("ojas"))

like 4

# Integer

print("\*\*\*\*\*\*\*\* Integer \*\*\*\*\*\*\*\*\*")

a = 10

b = 20

print(f"{a}") # empty expression not allowed

print(f"{a} {b}")

print(f"{b} {a}")

#Float

print("\*\*\*\*\*\*\*\* Float \*\*\*\*\*\*\*\*\*")

c = 10.56

d = 20.42

print(f"{c}")

print(f"{c} {d}")

print(f"{d} {c}")

#String

print("\*\*\*\*\*\*\*\* String \*\*\*\*\*\*\*\*\*")

f\_name = "alex"

l\_name = "albert"

print(f"{f\_name}")

print(f"{f\_name} {l\_name}")

print(f"{l\_name} {f\_name}")

# Integer and String

name = "alex albert"

age = 10

print(f"Hello My Name is {name}")

print(f"{name} {age}")

print(f"{age} {name}")

Integer

print("\*\*\*\*\*\*\*\* Integer \*\*\*\*\*\*\*\*")

num = 15

print(f"{num}") # Treated as String becoz no type mention

print(f"{num:d}")

print(f"{num:5d}")

print(f"{num:05d}")

print(f"{num:+5d}")

print(f"{num:<5d}") # Left align

print(f"{num:\*<5d}") # Left align with fill

print(f"{num:\*>5d}") # Right align with fill

print(f"{num:^5d}") # Center align

print(f"{num:\*^5d}") # Center align with fill

like 1

Float

print("\*\*\*\*\*\*\*\* Float \*\*\*\*\*\*\*\*")

num = 15.65

price = 15.65123456789

print(f"{num}") # Treated as String becoz no type mention

print(f"{num:f}")

print(f"{num:8f}") # 15.650000

print(f"{price:.20f}")

print(f"{num:8.3f}")

print(f"{num:+8.2f}")

print(f"{num:<8.2f}") # Left align

print(f"{num:\*<8.2f}") # Left align with fill

print(f"{num:\*>8.2f}") # Right align with fill

print(f"{num:^8.2f}") # Center align

print(f"{num:\*^8.2f}") # Center align with fill

#String

print("\*\*\*\*\*\*\*\* String \*\*\*\*\*\*\*\*")

name = "alexa"

print(f"{name}")

print(f"{name:s}")

print(f"{name:8s}")

print(f"{name:<8}")

print(f"{name:\*<8}")

print(f"{name:>8}")

print(f"{name:\*>8s}")

print(f"{name:^8s}")

print(f"{name:\*^8s}")

print("\*\*\*\*\*\*\*\* Truncating String \*\*\*\*\*\*\*\*")

name = "alexalbert"

print(f"{name:.3s}")

print(f"{name:8.3s}")

print(f"{name:\*<8.3s}")

print(f"{name:>8.3s}")

print(f"{name:\*>8.3s}")

print(f"{name:^8.3s}")

print(f"{name:\*^8.3s}")

# Thousand Separator

price = 1234567890

print(f"{price:,}")

print(f"{price:\_}")

#Variable

name = "Rahul"

age= 62

print(f"My name is {name} and age {age}")

# Expression

print(f"{10\*8}")

# Expressing a Percentage

a = 50

b = 3

print(f"{a/b:.2%}")

# Accessing arguments� items

value = (10, 20)

print(f"{value[0]} {value[1]}")

# Format with Dict

data = {'rahul': 2000, 'sonam': 3000}

print(f"{data['rahul']:d} {data['sonam']:d}")

like 2

Calling Function

name= "alexalbert"

print(f"{name}")

print(f"{name.upper()}")

# Using object created from class

#print(f"{obj.name}")

# Curly Braces

print(f"{10}")

print(f"{{10}}")

# Date and Time

from datetime import datetime

today = datetime(2019, 10, 5)

print(f"{today}")

print(f"{today:%d-%b-%Y}")

print(f"{today:%d/%b/%Y}")

print(f"{today:%b/%d/%Y}")

# Datetime Directive https://docs.python.org/3.7/library/datetime.html#strftime-and-strptime-behavior

like 2

datetime — Basic date and time types — Python 3.7.13 documentation

print("%d" % 432)

print("%d %d" % (432, 345))

print("%f" %432.123)

print("%f %f" %(432.123, 10.3))

print("%f" %432.123456)

print("%f" %432.12345651)

print("%s" % "alexalbert")

print("%s %s" % ("Hello", "alexalbert"))

print("%d %s" % (432, "alexalbert"))

#print("%s %d" % (432, "alexalbert")) TypeError

print("%(nm)s %(ag)d" % {'ag':432, 'nm':"alexalbert"})

print("% d" % 432)

print("% d" % 432)

print("%+d" % 432)

print("%8d" % 432)

print("%08d" % 432)

print("%.3f" %432.123)

print("%.2f" %432.123)

print("%.2f" %432.128)

print("%9.2f" %432.128)

print("%09.2f" %432.123)

print("%9.2f" %4388453232.124)