

Number Formatting

```
number = 210
# decimal places
print( f"number: {number:.2f} ")
# hex conversion
print( f"hex: {number:#0x} ")
# binary conversion
print( f"binary: {number:
r:b} ")
# octal conversion
print( f"octal: {number:o} ")
# scientific notation
print( f"scientific: {number:
r:e} ")
# total number of characters
print( f"Number: {number:0 -
9} ")
# use comma separator
apple_marketcap = 2.626 *
10e12
print( f"{apple_marketcap =
:,} ") # comma separator
#set 2 decimal places and add a
percentage sign to the end
percentage = 10.39439423
print( f"{percentage = :.2%} ")
# percentage
#set 4 decimal places
number= 10.39439423
print( f"{number:.4f} ")
#Output:
number: 420.00
hex: 0x1a4
binary: 110100100
octal: 644
scientific: 4.200000e+02
Number: 000000420
apple_marketcap = 26,260,000,000.0
```

Number Formatting (cont)

```
percentage = 1039.44%
10.3944
```

Repr & Str

```
# __repr__ = developer friendly
# __str__ = user friendly
# write !r to tell Python to
print out the repr method
instead.
from dataclasses import
dataclass
@dataclass
class Person:
    name : str
    age : int
    def __str__(self) ->
str:
        return f"{s -
elf.name} is {self.age} years
old"
Elon = Person ("Elon Musk", 51)
print( f"{Elon} ") # str
print( f"{Elon !r} ") # repr
#output:
Elon Musk is 51 years old
Person (name= 'Elon Musk',
age=51)
```

Date formatting

```
import datetime
today = datetime.datetime.now()
print( f"datetime :
{today}\n ")
# no microseconds
print( f"datetime:
{today:%m/%d/%Y %H:%M:%S} ")
# date only
print( f"date:
{today:%m/%d/%Y} ")
# time only
print( f"time:
{today:%H:%M:%S.%f} ")
```

Date formatting (cont)

```
# time with AM/PM
print( f"time:
{today:%H:%M:%S %p} ")
# 24-hour format
print( f"time:
{today:%H:%M} ")
# Locale's appropriate date and
time representation
print( f"locale appropriate:
{today:%c} ")
# weekday
print( f"weekday:
{today:%A} ")
# day of the year
print( f"day of year:
{today:%j} ")
# how far are we into the year?
day_of_year = f"{today:%j}"
print( f"progress % year:
{int(day_of_year)/365 *
100:.2f}% ")
#output
datetime : 2022-09-13 05:44: -
17.546036
date time: 09/13/2022 05:44:17
date: 09/13/2022
time: 05:44:17.546036
time: 05:44:17 AM
time: 05:44
locale appropriate: Tue Sep 13
05:44:17 2022
weekday: Tuesday
day of year: 256
progress % year: 70.14%
```

Debugging

```
# works in Python 3.8+
x = 10
y = 20
print( f"x = {x}, y = {y}")
print( f"{x = }, {y = }")
# math operations
print( f"{x * y = }")
#output:
x = 10, y = 20
x = 10, y = 20
x * y = 200
```

Multi-line f-string

```
company_name = "Tesla"
employee_count = 100000
mission = "To accelerate the
world's transition to sustain -
able energy "
print( f"""
Company: {company_name}
# of employees: {employee_count},
Mission: {mission}
" " ")
#output:
Company: Tesla
# of employees: 100,000
Mission: To accelerate the
world's transition to sustain -
able energy
```

Alignment

```
number = 4
# n stands for the width of
space to print the variable
number starting from the string
"is" (inclusive of the variable
itself)
print( f"number is {number -
r:4}")
#number is 4
for number in range(1, 5):
    print( f"the number is
{number r:{ number }}")
# the number is 1
# the number is 2
# the number is 3
# the number is 4
left = "left text"
center = " center text!"
right = " right text"
# left:>20 means given a width
of 20 characters print out the
string "left text" starting from
the left.
print( f"{left :>20}") # left
align
# left text
# center:^20 that means to leave
whatever space is left on the
left and right. Since the string
"center text!" is 12 charac -
ters, the left and right would
have four characters of white
space.
print( f"{center : ^20}") #
center align
# center text!
print( f"{right :<20}") #
right align
#right text
```

Alignment (cont)

```
# If we put all three strings
together with their formatting
options, we would have a width
of 60 to place the left, center,
and right string variables.
print( f"{left : <20 }{center :
^20}{right : >20 }")
#left text center text! right
text
```

