Functions of datetime module

Function	Example
ctime()	from datetime import date Str1= date(2021, 11, 24).ctime() print("Value as string: ", Str1) Str2= date(2022, 7, 12).ctime() print("Value as string: ", Str2)
fromisocalendar()	from datetime import date iso_today = date.fromisocalendar(today.year ,49,6) print(iso_today) **Discription:** This method returns the ISO calendar date that is equivalent to the given Gregorian calendar date, according to these specified parameters: • **year**: It takes the year number as integer type value. • **week**: It takes the week number, with ranges [1 to 52 or 53], as an argument. • **day**: It also takes the number of the day of the week, with ranges [1 to 7], as an argument.
fromisoformat()	from datetime import date print(date.fromisoformat('2023-09-25'))
isocalendar()	from datetime import date Td = date.today() print(Td) print(Td.isocalendar())
isoformat()	from datetime import date td= date.today() date_str = td.isoformat() print(date_str)
isoweekday()	from datetime import date td = date.today() print("Today's date:", td) dayNumber = td.isoweekday() print("Date:", td, "falls on", dayNumber)
replace()	from datetime import date dt = date(2010, 2, 12) print("Original date : ", dt) New_date = dt.replace(year=2023,month=9,day=25) print("After Modify the year:", New_date)
strftime()	**Example 1:** from datetime import date td = date.today() formatted = td.strftime("%Y-%m-%d") print(formatted)

	Example 2: from datetime import datetime as dt CDT = dt.now() formatted = CDT.strftime("%Y-%m-%d %H-%M-%S") print(formatted)
weekday()	from datetime import date td = date.today() print(td) print(td.weekday())
timetuple()	from datetime import datetime td = datetime.today() tt=td.timetuple() print(tt) for i in tt: print(i)