Open in Colab

Prelab 1.2 Python algorithms for datetime and saving into a csv file

2.1 Python datetime function

Python has a special date class, called “date.” A date, like a string, or a number, or a numpy array, has special rules for creating it and methods for working with it.

Here is an example on computing and printing local date and time:

In [2]:
```python
# Import datetime from the datetime module
from datetime import datetime

# Compute the local datetime: local_dt
local_dt = datetime.now()

# Print the local datetime
print(local_dt)
```

```
2026-01-27 20:40:43.370915
```

2.2 Formating date and time

You can also display the datetime function in different ways depending on convenience. The following example illustrates how to personalize the date-time formatting:

In [3]:
```python
# Compute the local datetime: local_dt
local_dt = datetime.now().strftime("%Y-%m-%d %H:%M:%S.%f") #The string inside parenthesis defines the format

# Print the local datetime
print(local_dt)
```

```
2026-01-27 20:40:45.314942
```

Task 2.1

Display the local datetime in the following formats:

- month/day/year
- hour:minute:second

In [ ]:
```python
# Time formatting
local_dt = datetime.now()
print(f"month/day/year: {local_dt.strftime('%m/%d/%Y')}")
print(f"hour:minute:second: {local_dt.strftime('%H:%M:%S')}")
```

```
month/day/year: 01/27/2026
hour:minute:second: 20:40:47
```

You can also perform arithmetics in Python.

In [5]:
```python
# Import date
from datetime import date
# Create Dates
superbowl_dates = [date(2019, 2, 3), date(2020, 2, 2), date(2021, 2, 7), date(2022, 2, 13)]

#Subtracting the third and first date
difference = superbowl_dates[2]-superbowl_dates[0]
```

```
#Print the results as a sentence. You can concat strings by using commas
print('The difference between Superbowls on',superbowl_dates[2],'and',superbowl_dates[0],'is',difference)
```

```
The difference between Superbowls on 2021-02-07 and 2019-02-03 is 735 days, 0:00:00
```

You can also use a time function as well. The following example calculates the time difference before and after printing a string:

In [6]:

```
#Import time library
import time
start = time.time()
print("hello") #Printing the string = hello
end = time.time()
print(end - start,'seconds')
```

```
hello
0.00021529197692871094 seconds
```

Task 2.2

- Calculate how many days are left on this year.
- Determine if the processing time for printing longer strings in Python changes significantly in terms of seconds.

In [38]:

```
# Days left in the year
yr_end = datetime(2027,1,1)
time_to_yr_end = yr_end - local_dt
print(f"Days left in this year: {time_to_yr_end}")

# Processing time
def time_printing(string):
    start = time.time()
    print(string)
    end = time.time()
    processing_time = end - start
    print(f"Processing time to print the string: {processing_time} seconds")
    return processing_time

short_time = time_printing("hi") # short string
long_time = time_printing("hi this is a much longer strinmg than the previous one") # long string

# Compare processing times
if round(long_time, 0) > round(short_time, 0):
    print("The longer string took more time to process.")
elif round(short_time, 0) > round(long_time, 0):
    print("The shorter string took more time to process.")
else:
    print("Both strings took roughly the same number of seconds to process.")
```

```
Days left in this year: 338 days, 3:19:12.716392
hi
Processing time to print the string: 5.9604644775390625e-06 seconds
hi this is a much longer strinmg than the previous one
Processing time to print the string: 3.337860107421875e-06 seconds
Both strings took roughly the same number of seconds to process.
```

2.3 (Preview) Using Python to save csv file

In Lab 1, you will integrate code block as shown below. The following algorithm performs data saving of multiple rows into a csv file. Carefully read all comment lines and become familiar with the code.

Warning: These algorithm will not run on its own. It requires additional lines of code that connect the data with to the algorithm.

Python - use this code block to save data in a csv file on the Raspberry Pi

```
# writing to CSV file using csv package
with open(filename, 'w') as f:
    # creating a CSV wiriting object
    write = csv.writer(f)

    # writing header on the first row
    write.writerow(header)

    #Use a 'while' or 'for' cycle to extract data row by row.
    while (declare the conditional within these parentheses):
        i += 1              # indicator + 1 for each row in a loop
                            # write your timestamp for measurement
                            #extract information from the sensor and save it as a variable


        data = [timestamp, str(var1), str(var2), str(var3)] # data list, all elements are
string data type

        write.writerow(data) # writing measured data

        # printing indicator, timestamp, var1, var2, var3 (this section is optional)
        print("{}th measurement, {}:\n var1={:.4f}units, var2={:.4f}units, var3=
{:.4f}unit\n".format(i, timestamp, var1, var2, var3))

        time.sleep(1)

# gently close the CSV file object
f.close()
```

Get back to Lab Index Page