

L08 Tasks

L08-T1: Using Python libraries: Math & Random

L08-T2: Continue Python library: Random

L08-T3: Python module fraction

L08-T4: Python library: datetime

L08-T5: try-except

Submission:

Submit **all** the exercises to CodeGrade via Moodle before the deadline.

Note:

- If you see some "Code Structure Tests" hidden in CodeGrade but they are not mentioned in the task description, you don't need to worry about them. They are just there to make sure the code is ok.
- Be especially careful with spaces so that your output follows the sample output. Note that also each input-string in the program is ending with '\n'. The reason for this is that it makes the output in CodeGrade more readable.

L08-T1: Using Python libraries: Math & Random

Write a program that uses different Python libraries in different subroutines:

1. **Calculate the area of a circle with the math library.** The program asks the user for the radius of the circle and then calculates the surface area of the circle. Use the pi value (pi) found in the math library and the power calculation function pow. Finally, print the area rounded to 2 decimals.
2. **Guessing a random number with the random library.** The program selects a random number between 0-1000 with the randint function of the random library. After that, the program asks the user for a guess and tells whether the searched number is larger or smaller until the user guesses the number correctly. When the user guesses the number, the program also informs how many tries the user needs to find the correct number

The program should print "Unknown choice, please try again." if it cannot recognize the user's input.

You should use pseudo-random numbers. To make testing the program possible, **set the seed number** before the number is drawn with the **random.seed(1)** command.

Note: “Code structure tests” will be used in this code to make sure that the program is using the libraries correctly.

See the example run below for more detailed operation of the program.

Example run 1:

```
This program uses libraries to solve tasks.
What do you want to do:
1) Calculate the area of the circle
2) Guess the number
0) Stop
Your choice:
1
Enter the radius of the circle as an integer:
7
With a radius of 7, the area of the circle is 153.94.

What do you want to do:
1) Calculate the area of the circle
2) Guess the number
0) Stop
Your choice:
2
Guess the integer drawn by the program.
Enter an integer between 0 and 1000:
500
The requested number is lower.
Enter an integer between 0 and 1000:
250
The requested number is lower.
Enter an integer between 0 and 1000:
125
The requested number is higher.
Enter an integer between 0 and 1000:
187
The requested number is lower.
Enter an integer between 0 and 1000:
140
The requested number is lower.
Enter an integer between 0 and 1000:
132
The requested number is higher.
Enter an integer between 0 and 1000:
137
Correct! You used 7 tries to guess the correct integer.
```

```
What do you want to do:
1) Calculate the area of the circle
2) Guess the number
0) Stop
Your choice:
0
See you again!
```

L08-T2: Continue Python library: Random

Your goal in this task is to create a random password generator. For the purposes of checking code with CodeGrade, we will be using Python's random package. HOWEVER, in real world, you should use "secrets" package, which gives truly random values (<https://docs.python.org/3/library/secrets.html>)

To have agreements on what the character set is, you need to import string package and agree with the following combinations:

```
LETTERS = string.ascii_letters
```

```
DIGITS = string.digits
```

```
SPECIAL = string.punctuation
```

Then the set of used characters is

```
COMB = LETTERS + DIGITS + SPECIAL
```

Then ask the user for the length of the password. For the purposes of submitting this task to CodeGrade, set Python to use **random.seed(8292)**. That way you get the same result every time. Use **random.choice(COMB)**, where the string COMB is defined as above.

Note: "Code structure tests" will be used in this code to make sure that the program is using the libraries and the given combinations correctly.

Example run 1:

```
Enter the length of the password:
0
Password length must be a positive integer.
Enter the length of the password:
16
Generated password: bQ:<AXWQ7`SZ@*i*
```

L08-T3: Python module fraction

This exercise deals with fraction package. The task is to demonstrate how mathematical operations work on fractions. Write a program that asks the numerator and the denominator of two fractions. The program also asks for an integer as an exponent. The program forms fractions from the given numerators and denominators and performs the usual mathematical operations according to the example below:

Note: “Code structure tests” will be used in this code to make sure that the program is using the module correctly.

Example run 1:

```
Give the first fraction.  
Give numerator (top):  
26  
Give denominator (bottom):  
37  
Give the second fraction.  
Give numerator (top):  
17  
Give denominator (bottom):  
41  
Give an exponent:  
3  
Sum: 26/37 + 17/41 = 1695/1517  
Subtract: 26/37 - 17/41 = 437/1517  
Multiply: (26/37) * (17/41) = 442/1517  
Divide: (26/37) / (17/41) = 1066/629  
Power: (26/37)**3 = 17576/50653
```

L08-T4: Python library: datetime

We will continue practicing the use of datetime library. The program is menu-based, see the example runs below. The selected operation should call a function doing the required task.

The tasks are the following:

Instance variables of the datetime object. The program asks the user for the date and time as a string in the format dd.mm.yyyy hh:mm and converts the string into a datetime object. From this object, it is easy to use instance variables: date, month, year, hour, minute to print them to the screen according to the example run.

Calculation of the length of the period. The program asks the user for date of birth in the format dd.mm.yyyy and calculates how old he/she was at the beginning of this year, that is, on January 1, 2024, and prints the answer in days. By using timedelta objects you can deal with periods of time.

Printing the names of the days of the week. With the strftime function, you can print the information contained in the datetime object in a versatile way. Use this function to print the names of the days of the week on the screen.

1. Start by creating a datetime object using Monday's date (any Monday's date will do).
2. After that, go through all the days of one week with the repeat structure and move to the next day by using timedelta. Please note that the language settings in the computer settings may affect the language of the printout, but in CodeGrade the days of the week are printed in English with the strftime function.

Printing the names of the months of one year as abbreviations. The task is to print the names of the 12 months correctly. Note that the transition must always fall on the next month, but not necessarily on the same day. The names of the months should be printed as *abbreviations*.

See the example run below for more detailed operation of the program and the programming manual describes the most important functions of the datetime module. Implement each of the above functions as its own subroutine, typically less than 10 lines per subroutine.

The program should print “Unknown choice, please try again.”, if it cannot recognize the user’s input.

Note: “Code structure tests” will be used in this code to make sure that the program is using the libraries correctly.

Example run 1:

```
This program uses the datetime library to deal with time.
What do you want to do:
1) Identify the components of a time object
2) Calculate age in days
3) Print the days of the week
4) Print the months
0) Stop
Your choice:
1
Enter the date and time in the format 'dd.mm.yyyy hh:mm':
24.12.2024 19:25
You gave year 2024
You gave month 12
You gave day 24
You gave hour 19
You gave minute 25

What do you want to do:
1) Identify the components of a time object
```

```
2) Calculate age in days
3) Print the days of the week
4) Print the months
0) Stop
Your choice:
2
Enter your birthday as dd.mm.yyyy:
11.11.2000
On January 1, 2024, you were 8451 days old.
```

```
What do you want to do:
1) Identify the components of a time object
2) Calculate age in days
3) Print the days of the week
4) Print the months
0) Stop
Your choice:
3
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
```

```
What do you want to do:
1) Identify the components of a time object
2) Calculate age in days
3) Print the days of the week
4) Print the months
0) Stop
Your choice:
4
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
```

```
What do you want to do:
1) Identify the components of a time object
2) Calculate age in days
3) Print the days of the week
4) Print the months
0) Stop
Your choice:
0
See you again!
```

L08-T5: try-except

You can use the datetime module to check whether a given date is valid. You can attempt to create a datetime object with the provided date, and if an exception is raised, it means the date is not valid.

What is an **exception**? We will consider exceptions properly in Lecture 10, but here is a simple Python example of using a try-except block to catch a ZeroDivisionError exception:

```
try:
    result = 10 / 0 # This is not allowed
except:
    print("Error: Division by zero")
```

In this example, the "try" block attempts to perform a division operation that would result in division by zero. Since this is an exceptional event, a ZeroDivisionError is raised, and the program gracefully handles the error in the "except" block by printing an error message. The advantage of the try-except structure is that an error does not crash the program

Write a program that asks the user to enter a date in YYYY-MM-DD format. Whether the date is correctly formed, it prints the answer according to the examples below.

Note: "Code structure tests" will be used in this code to make sure that the program is using the libraries/modules correctly.

Example run 1:

```
Enter a date in YYYY-MM-DD format:
2023-02-29
2023-02-29 is not a valid date.
```

Example run 2:

```
Enter a date in YYYY-MM-DD format:
2023-09-31
2023-09-31 is not a valid date.
```

Example run 3:

Enter a date in YYYY-MM-DD format:

2023-12-31

2023-12-31 is a valid date.