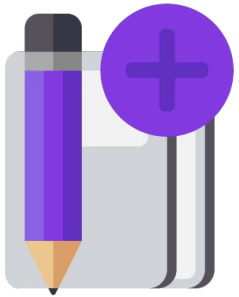
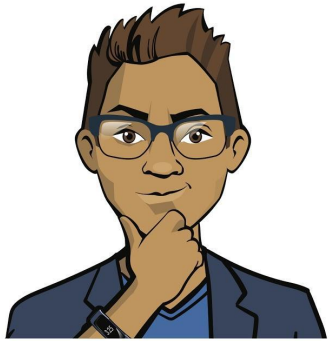
Brainstorm:

The random module



The Python standard library

All the features of the Python language are described in a large set of Python files called a ***library*** .

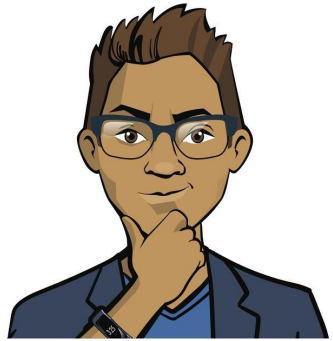


Brainstorm

The Python standard library

All the features of the Python language are described in a large set of Python files called a ***library*** .

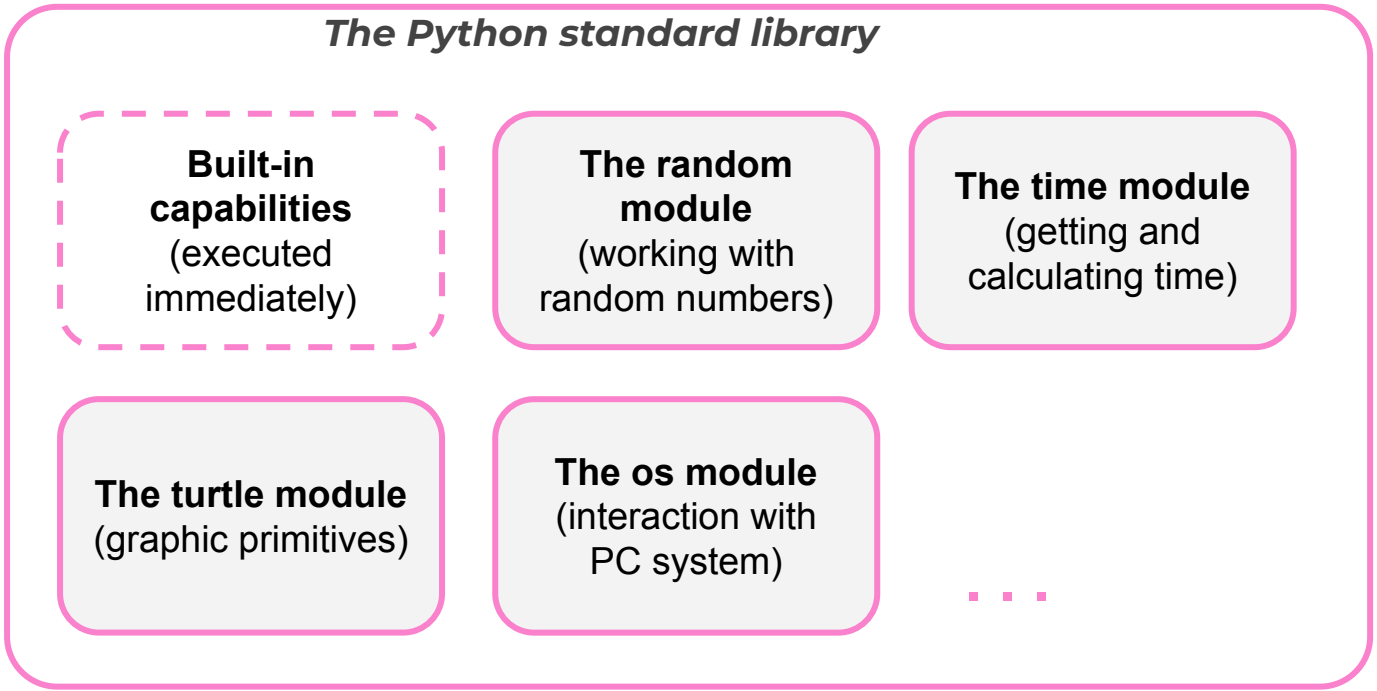
- The interpreter immediately recognizes and executes the simplest commands (e.g., input-output).
- Other commands (e.g., those for random numbers) are used less often, so they are stored in separate library files called ***modules*** .



Brainstorm

The Python standard library

Many programmers like Python specifically because of this wide range of features in the standard library and modules.



Brainstorm

The random module contains tools for generating random values

Brainstorm

Useful functions:

Function	*Purpose*
`randint(a, b)`	To generate a random integer greater than or equal to a and less than or equal to b
`random()`	To generate a random decimal from 0 (inclusive) to 1 (not inclusive)

Connecting and using the module

Step 1
Step 2

```
import random
number = random.randint(0, 9)
```

Connecting the whole module. When using a command, we need to specify which module it is from.

Step 1
Step 2

```
from random import randint
number = randint(0, 9)
```

Connecting one function from the module. When we call a function, only its name is specified.

Step 1
Step 2

```
from random import *
number = randint(0, 9)
```

Connecting all the functions of the module. When we call a function, only its name is specified.

Brainstorm

The random module contains tools for generating random values

Brainstorm

Useful functions:

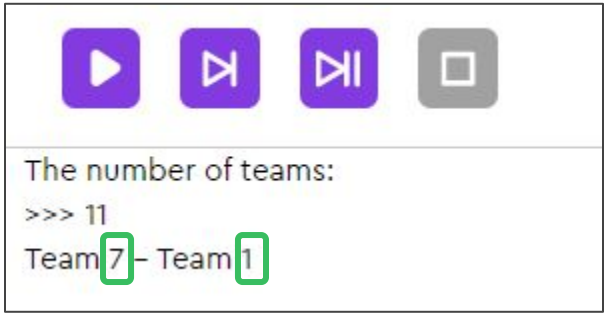
Function	*Purpose*
`randint(a, b)`	To generate a random integer greater than or equal to a and less than or equal to b
`random()`	To generate a random decimal from 0 (inclusive) to 1 (not inclusive)

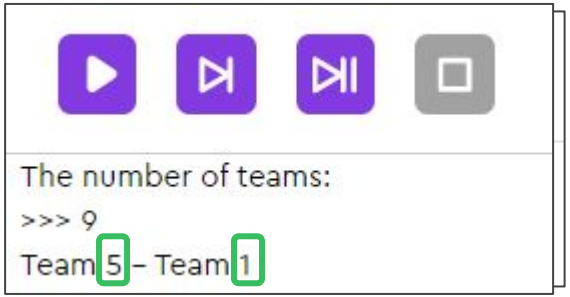
Example:

```
from random import randint

lottery_num = randint(1000, 9999)

print(lottery_num)
```

Let’s go back to the task

Task. Write a program that draws lots for water polo teams. As a result, the program should print a random pair of numbers to indicate which teams will play one another in the qualifying match.





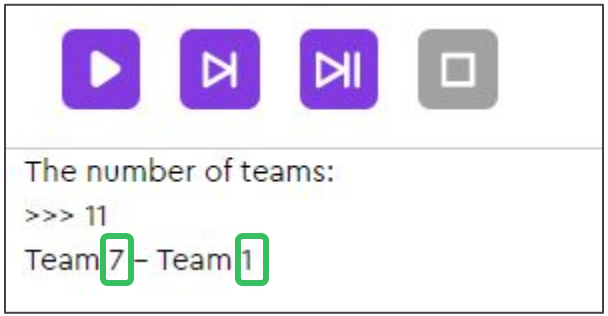
A situation in which team N is assigned to compete against itself shall be considered normal. If this occurs, the referee will simply re-run the program.

Brainstorm

Let’s go back to the task

Task. Write a program that draws lots for water polo teams. As a result, the program should print a random pair of numbers to indicate which teams will play one another in the qualifying match.

```
from random import randint


total = int(input('The number of teams:'))
participant_1 = randint(1, total)
participant_2 = randint(1, total)
print('Team', participant_1, '-', 'Team', participant_2)
```

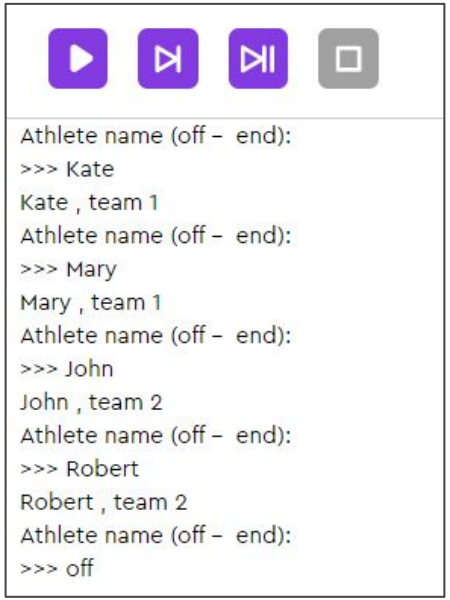


A situation in which team N is assigned to compete against itself shall be considered normal. If this occurs, the referee will simply re-run the program.

Brainstorm

Let’s go over one more task

Task. Write a program for the distribution of players into two teams during practice. The user enters the names of the athletes one by one (“off” – to stop input), and the program prints a random team for every person.



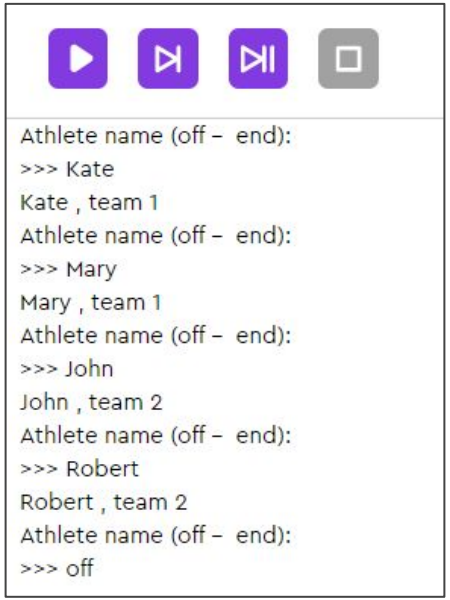
We will assume that the coach is responsible for the numerical equality of the teams, not the program.

Brainstorm

Let’s go over one more task

Task. Write a program for the distribution of players into two teams during practice. The user enters the names of the athletes one by one (“off” – to stop input), and the program prints a random team for every person.

```
from random import randint
name = input('Athlete name (off -  end):')
while name != 'off':
    team_num = randint(1, 2)
    print(name, ', team', team_num)
    name = input('Athlete name (off -  end):')
```

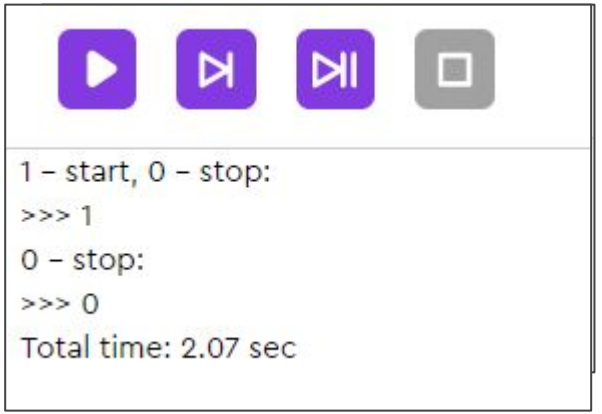


We will assume that the coach is responsible for the numerical equality of the teams, not the program.

Brainstorm

Let’s go over a task

Task. Write a stopwatch program to calculate the crawl swimming race time in seconds. At startup, the program prompts you to enter “1” to start counting. Entering “0” stops the stopwatch and displays the number of seconds.



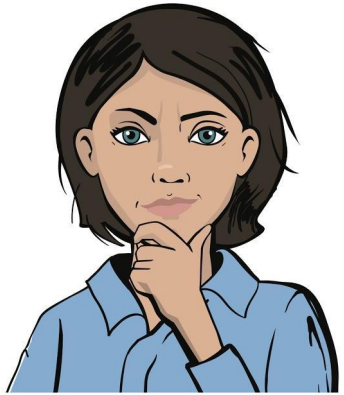
Brainstorm

Let’s go over a task

Task. Write a stopwatch program to calculate the crawl swimming race time in seconds. At startup, the program prompts you to enter “1” to start counting. Entering “0” stops the stopwatch and displays the number of seconds.

It seems that we have not worked with this module yet...

Which standard library module would be useful?



```
1 – start, 0 – stop:
>>> 1
0 – stop:
>>>
```

```
1 – start, 0 – stop:
>>> 1
0 – stop:
>>> 0
Total time: 2.07 sec
```

Brainstorm

The time module contains tools for working with time-related quantities

Brainstorm

Useful functions:

Function	*Purpose*
`time()`	Return the number of seconds since the beginning of the epoch *(for UNIX systems, this is January 1, 1970)*
`sleep(seconds)`	Suspend the program for the number of seconds
`round(number, digits)`	Round the number to the number of digits after the decimal point *Basic function (not from time)*

Let’s study how the functions work

A training program using the functions of the time module

Command	*Value and result*

`from time import *`
`start = time()`
`print(start)`
	1601619244.56282
	Let's fix the start time of the program

`sleep(5)`	Pause the program for 5 seconds

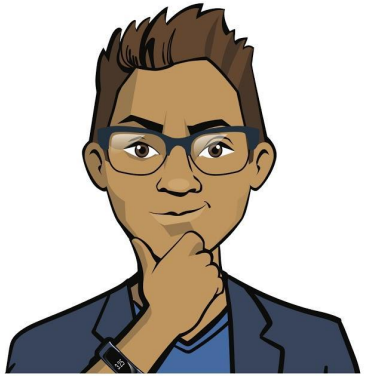
`end = time()`
`total = round(end - start, 2)`
`print(total)`
	6.31
	Calculate the running time of the program and round it up to two digits after the decimal point

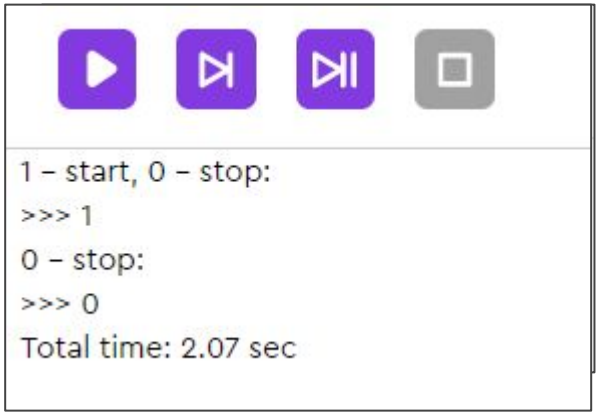
Brainstorm

Let’s go over a task

Task. Write a stopwatch program to calculate the crawl swimming race time in seconds. At startup, the program prompts you to enter “1” to start counting. Entering “0” stops the stopwatch and displays the number of seconds.

How do we solve this task?





Brainstorm

Let’s go over a task

Task. Write a stopwatch program to calculate the crawl swimming race time in seconds. At startup, the program prompts you to enter “1” to start counting. Entering “0” stops the stopwatch and displays the number of seconds.

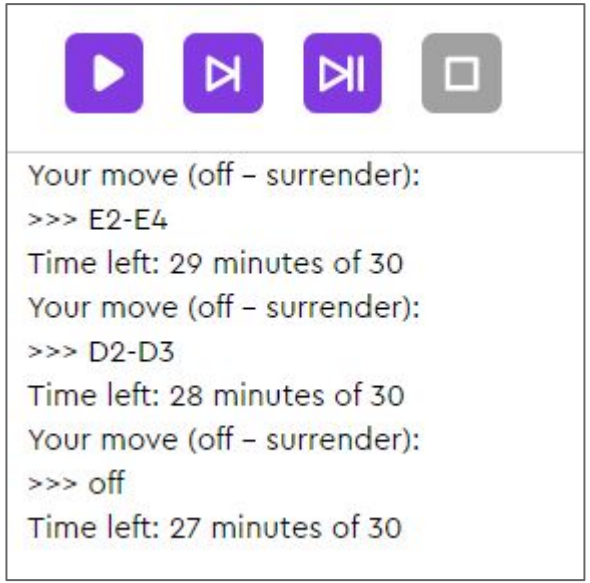
```
from time import time


stopwatch = input('1 - start, 0 - stop:')
while stopwatch != '0':
  if stopwatch == '1':
      start = time()
  else:
      print('Action not found!')
  stopwatch = input('0 - stop:')
end = time()
total = end-start
print('Total time:', round(total, 2), 'sec')
```

Brainstorm

Let’s go over a task

Task. In rapid chess, a player is given 30 minutes to make decisions for all their moves. The program must:

- ❏ Invite the player to make a move (e.g., E2 – E4) and count the elapsed time.
- ❏ After receiving the move, print the remaining time in minutes.
- ❏ If 30 minutes are over or the player enters “off” – exit.



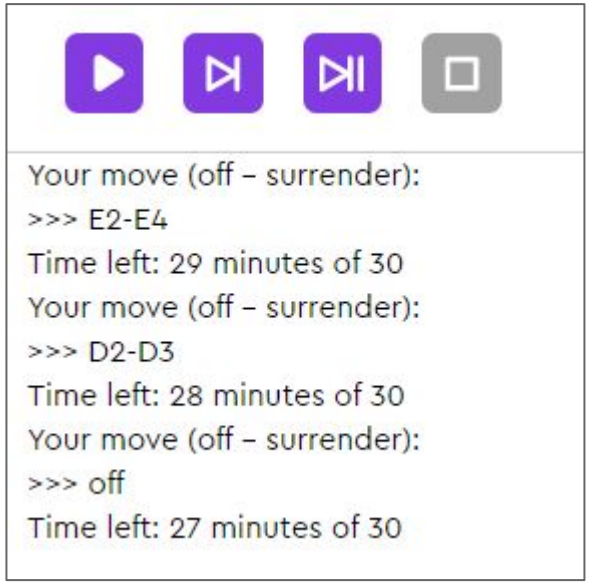
Brainstorm

Let’s go over a task

Task. In rapid chess, a player is given 30 minutes to make decisions for all their moves. The program must:

- ❏ Invite the player to make a move (e.g., E2 – E4) and count the elapsed time.
- ❏ After receiving the move, print the remaining time in minutes.
- ❏ If 30 minutes are over or the player enters “off” – exit.

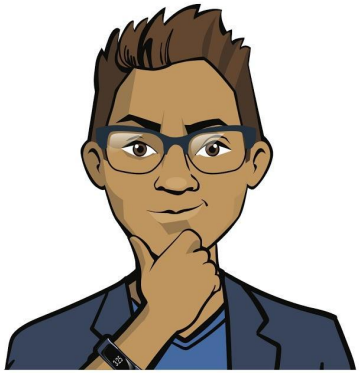
```
from time import time

rest = 30
move = ''
beginning = time()
while rest > 0 and move != 'off':
  move = input('Your move (off - surrender):')
  end = time()
  rest = 30 - (end - beginning)/60
  print('Time left:', int(rest), 'minutes of 30')
```



Brainstorm

Before we continue:

1. **How can I change the program** to display the time not in minutes but in (whole) seconds?
2. According to the new regulations, the time for thinking has increased from 30 minutes to 1 hour. **How will the program change?**



Brainstorm