# The Coin Game - "Heads or Tails"

# 1. A first version

The first version of the game will write in the screen "head" or "tail" at random.

Thus, the player should think of an option (heads or tails) and then run the program, checking whether or not guessed correctly.

In Python, the code that implements this game is as follows:

import random

coin = random.choice(["head", "tail"])

print (coin)

Random numbers are generated by a specific Python library called [random](https://docs.python.org/3/library/random.html), so you first need to import this library in the first statement.

Then, the program creates a first variable, called "coin", to which it assigns the value of "head" or "tail" at random.

The [random.choice](https://docs.python.org/3/library/random.html#functions-for-sequences) statement chooses one of the values from a sequence of values - a list. Python lists are enclosed in square brackets []. In this case, our list has only 2 values: "head" or "tail". Each of these words is enclosed in quotation marks (""). This is the way to define text, which in programming language is called a string. We will address this later...

Finally, in order to visualize the result of the coin thrown by the computer, the [print](https://docs.python.org/3/library/functions.html#print) function is used.

Now try the game!

# 2. A more interactive version

The first version of the game works but it is not very interactive ... Let's now add some more code so that this new version can present as a result if the player wins or loses.

In Python, the code that implements this new version of the game is as follows:

import random

# 1. selects head or tail

coin = random.choice(["head", "tail"])

# input from the player

play = input("head or tail? ")

print ("I played", coin)

# check if the player wins or loses

if coin == play:

print ("You win!")

else:

print ("You lose...")

First of all, observe that there are three lines that start with the hash character (#). These are comments written in English. They are used to make the program more readable and are completely ignored by the Python interpreter.

The algorithm of this new version is as follows:

1. Select face or tail at random.
2. Read the player's move and show the computer's move.
3. Check if the player won or not.

It is always important to post comments that describe the algorithm that has been outlined!

The first step of the algorithm is coded in the same way as in the previous version.

The second step of the algorithm makes the program interactive by asking the player to play. This step is called **Input**. The play variable stores the player's play, which is read through the [input](https://docs.python.org/3/library/functions.html#input) function. It is possible to place a text that serves as context for reading the player’s move. After reading the player's move, the computer also displays its own move.

Finally, the third step of the algorithm checks who wins! For that we need a conditional control structure - an [if](https://docs.python.org/3/reference/compound_stmts.html#the-if-statement). This structure evaluates a conditional expression (an expression that returns true or false). Note that to compare two expressions (in this case the play and coin variables) the == operator is used (two = signs, in order to distinguish from the assignment operator which is =). It is based on this conditional expression that only one of the blocks of code (defined by the indentation) is executed, visualizing who wins (the player or the computer).

|  |  |
| --- | --- |
| 🗬 | *Try writing your moves with uppercase characters. What happens?*  *One clue:* [*ASCII code*](https://en.wikipedia.org/wiki/ASCII) |

# 3. Final challenge

Try now to adapt your game to roll a dice (6 faces with numbers from 1 to 6).

|  |  |
| --- | --- |
| **Share with person** | Share your game! |