Exercise Javascript Functions

**Evaluated skills :**

*→ Functions*

**Game more or less :**

The game

The principle is simple: the computer chooses a random number, smaller or equal to 100.

The goal is to guess this number. Whenever we propose a number, we indicate if the number to guess is smaller or larger than this one.

The instructions

The game runs in this way:

* choice of a random number
* we ask for numbers, and we indicate if the number to be found is greater or smaller than those proposed
* when the user has won, the number of tries he has done is displayed.

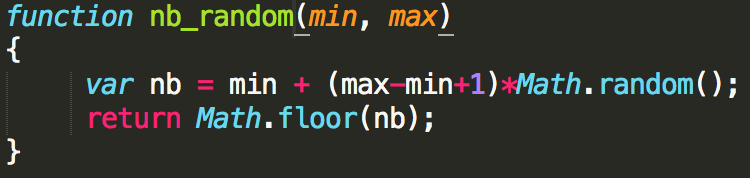
For now, we put the code directly in the code of the page, without creating a function.

Choice of a random number

In JS, there is no "simple" function giving a random integer, it must be created.

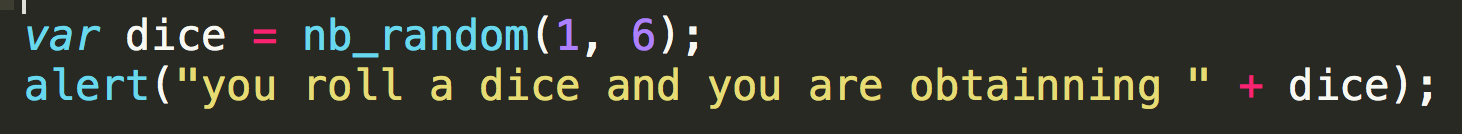
You will learn this in a few chapters, you are missing some notions for now.

I will give you the function :



This function returns an integer randomly chosen between min and max (inclusive).

An example of use to simulate a roll of dice:

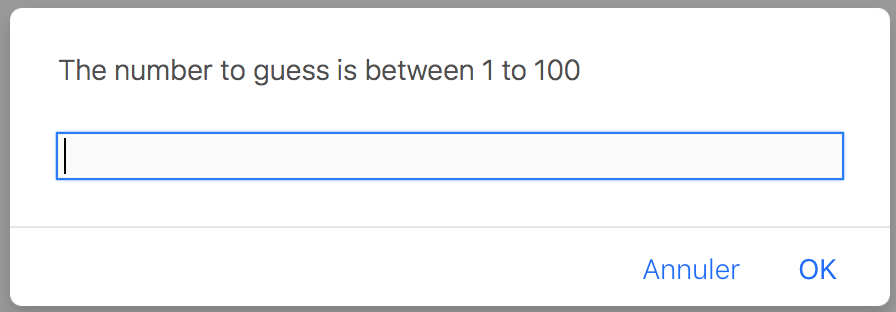


Ask for numbers and display "More" or "Less"

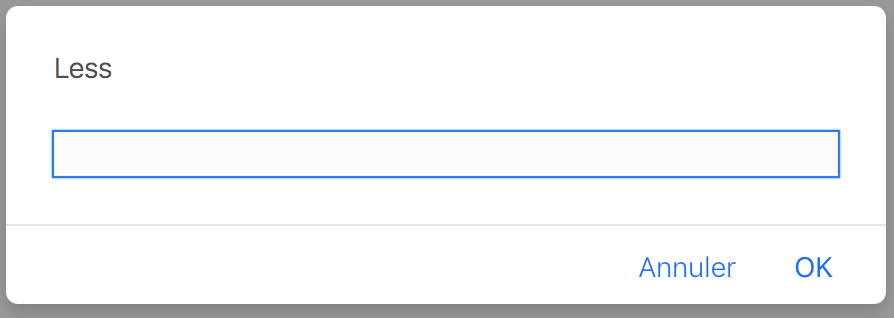
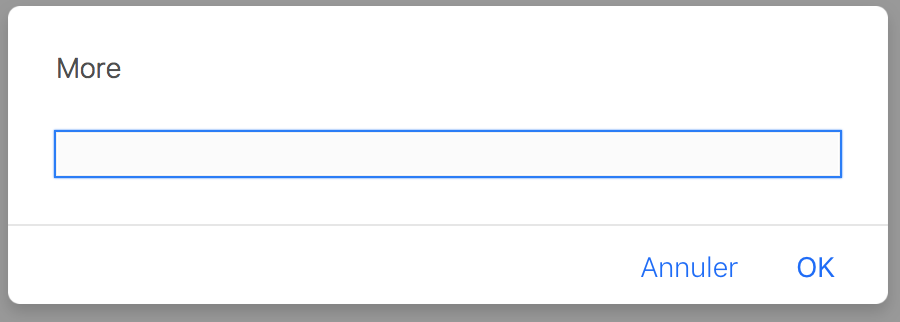
The second step is to ask for numbers, and to display each time if the number to guess is larger (more) or smaller (less), until the user has found the right number.

By the way, we count how much testing he needs before winning.

We will only use prompt, to prevent the user from clicking 50 times on "OK" at each part.



First message

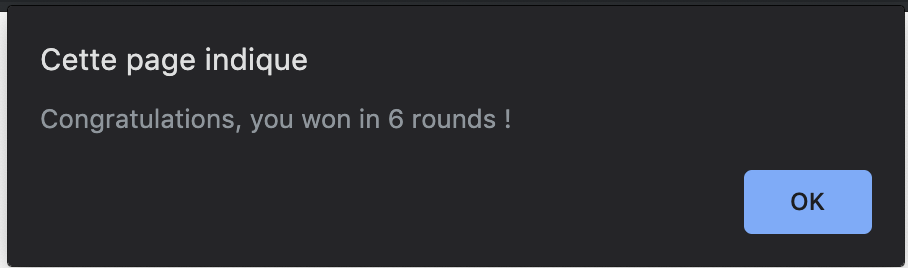


Next message, after entering a number

For now, we just ignore the "Cancel" button.

End of the game

We post a message announcing to the user that he won, telling him how many rounds it took.



That's all for the instructions, it remains only to code all that :).

Before you jump on your keyboard, think about how you will create your script (which loop to use, etc.).

**Help :**

Before giving you a correction, let's explain how to make this script.

**Where to start ?**

Structure of the script

First question to ask yourself:

→ what will our script look like?

It remains to know which structure to use.

Given the steps, it will be something like this:

* an initialization (mainly the declaration of the variables)
* a loop (to request values and display messages)
* some instructions to display the result.

The loop

We will have to choose which type of loop to use.

→ We can already eliminate the loop for: the condition of the loop is not about a counter, but about the numbers that are asked to the user.

→ We still have while and do ... while.

*As a reminder, the main difference between these two loops comes from the fact that the action is executed before the condition in the case of do ... while.*

*With while, the condition is evaluated first, and then only the action is executed.*

In our case, we will ask for a number at each loop round.

The condition will relate precisely to this number (is it equal to the number sought?).

It is better to use do ... while, because you have to ask for the number before checking the condition.

The variables

Let's try to identify the variables that we will have to use.

What we can already say is that we will need:

* a variable that will contain the number to guess: let's call it number
* a variable to count the number of moves, which we call cpt
* one to record the number entered by the user (it will test on), we will call input.
* In addition, when a user is asked for a number, the message to be displayed will depend on the number he has entered before.
* We will create a variable named msg that will contain the message that will be displayed the next time we ask for a number.

**Improvement 1: play several times in a row**

Purpose of this improvement

Short presentation :

Our script, which allows to play a part of "More or less", is now functional.

The improvement that I propose to perform will allow you to play several games, displaying, once the game is over, your best score (your lowest number of attempts).

Do we agree on the vocabulary ...

We will say that the script we created makes us play a round of the game "More or less" (which I will abbreviate in "LoM").

When we play several rounds (this is the object of this improvement), we will call it a game of LoM.

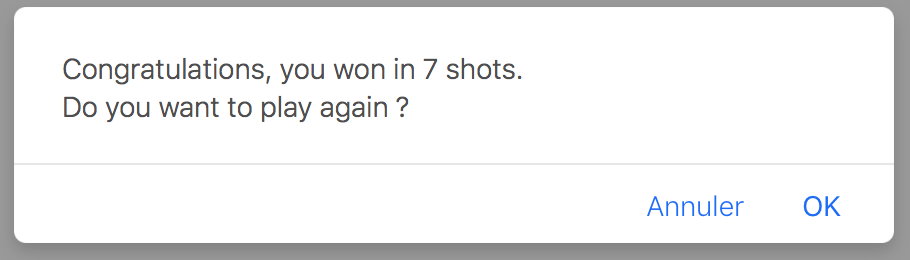
What you need to do

The first modification will consist in creating a function (we will call it LoM\_round), which will play a round of LoM, and which will return the score (instead of displaying it). So, with a few modifications, the code that you have realized.

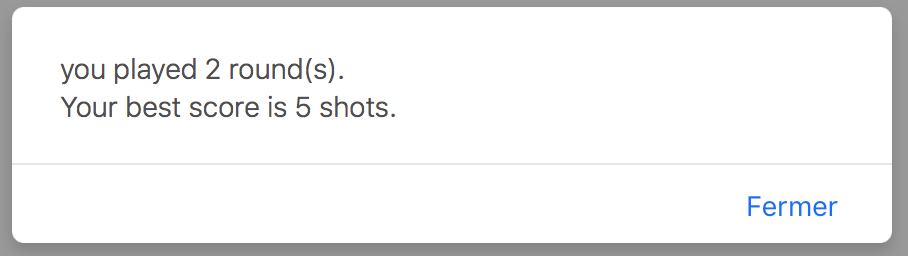
For more possibilities later, we will give two parameters, min and max, to this function, and the number that we will have to guess will be between these two values (included).

Then we will play several rounds, asking at the end of each if the user wants to play again.

Once the game is over, we will display the number of games played and the best score.



End of a round



Game over

**Improvement 2: some finishes**

**Purpose of improvement**

This is to correct some details:

we will create a function, LoM\_game, which will play a game of PoM\_round (from the code of improvement 1).

It will return the highest score (in addition to display it).

This function (in argument) will specify the values between which to choose the random number.

We will take into account the button "Cancel" when asked to enter a number: it stops the current round, displays the best score of the other rounds, and leaves the game.

The advantage of the first two points is that we can launch the game from a link on the page, and we can propose several levels of difficulty (taking a number between 1 and 250, for example) .

HTML to paste :

<p>

<a href="#" onclick="LoM\_game(1,63)">Zer0 Level</a><br />

<a href="#" onclick="LoM\_game(1,100)">Normal Level</a><br />

<a href="#" onclick="LoM\_game(1,250)">Geek Level</a>

</p>