# s = Number.isInteger(s) ? s : s.toFixed(2); 6 task

# FL8. Lesson 7. Control flow.

# Homework

1. The result of your work should be packed to **folder** with html and js files (separate html and js files for each task). Inner folder structure should be exactly as follows:

*|\_\_ homework\_07 /  
 |\_\_ js /*

*|\_\_ task1.js*

*|\_\_ task2.js*

*|\_\_ task1.html`*

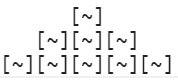
*|\_\_ task2.html*

1. Code has to be readable, tested and well-formatted.
2. The folder must be loaded into the GitHub repository '**front-end-lab-8**' in the **master** branch

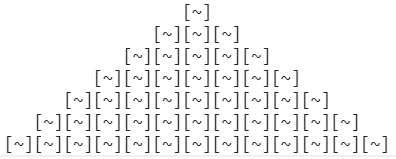
## Task #1. Pyramid

Write a program due to the following requirements:

* Ask user to enter natural number **N** (0<**N**<=20) // use prompt()
* If number **N** is not natural or doesn`t match required range, output error message in console // For example: console.error(‘Incorrect!’);
* If **N** match all requirements you should build and output in console pyramid with **N** floors



*Pyramid with 3 floors (N = 3)*



*Pyramid with 7 floors (N = 7)*

* To build pyramid use spaces, square brackets, tilde and **\n** (new line) characters.

function createPyramid(rows)

{

for(var i=0;i<rows;i++) {

if (i % 2 == 1) continue;

var output="";

for(var j=0;j<rows-i;j++) {

output+=" ";

}

for(var k=0;k<=i;k++) {

output += "[~] ";

}

console.log(output);

}

}

createPyramid(11);

const stars = function(x) {

if(x === 0) {

return '';

}

return '[~]' + stars(x-1);

};

const space = function(y) {

if (y === 0) {

return '';

}

return ' ' + space(y-1);

};

const triangleStars = function(z) {

if (z === 0) {

return 0;

}

console.log(space(z-1) + stars((num-z)\*2 + 1));

return triangleStars(z-1);

};

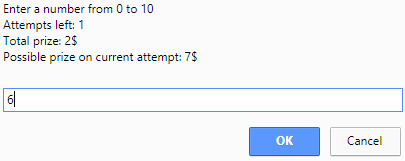
const num = 4;

triangleStars(num);

## Task #2. Guessing game

Your task is to write a game. Requirements:

* Show a message ‘Do you want to play a game?’ *// use confirm()*
* If user clicked ‘Cancel’ – output in console message `You did not become a millionaire`, if clicked ‘Ok’ – start a game: randomly /\* Math.random() \*/ choose a number in range [0; 5], and ask user to enter a number
* User has 3 attempts to guess a number
* If user guessed number on 1-st attempt prize is 10$ (maximum prize for current numbers range), 2-nd attempt – 5$, 3-rd attempt – 2$
* If user did not guess a number, output in console message ‘Thank you for a game. Your prize is: …’, and ask if he wants to play again.
* If user did guess - ask if he wants to continue a game *// use confirm().*
* If user does not want to continue – output in console message ‘Thank you for a game. Your prize is: …’, and ask if he wants to play again.
* If user does want to continue, make number range twice as big as the previous one /\* for example [0; 5] -> [0; 10] \*/, and three times bigger maximum prize /\* for example on 1-st attempt prize will be 30$, 2-nd attempt – 15$, 3-rd attempt – 7$ \*/. Prize must be added to the previous one and number of attempts should be set to 3 (user should has 3 attempts to guess a number for each numbers range)
* Each time you ask user to enter a number */\* using prompt() \*/* you should show him a range of numbers, how much attempts he has left, his total prize and possible prize on current attempt. See example below:



* All these stuff should be repeated until user lose or decide to quit