**Циклы**

Констукции и примеры:

**While**

**#1**

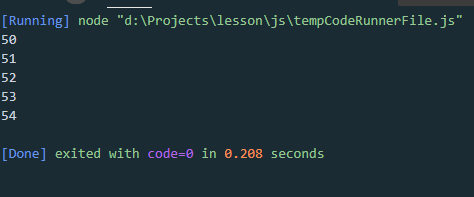
let num = 50;

while(num < 55) {

    console.log(num);

    num++;

}



**DO**

**#2**

let num = 50;

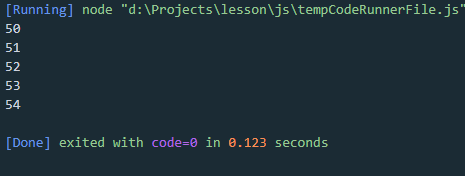
do {

    console.log(num);

    num++;

}

while (num < 55);



**FOR**

**#3**

for (let i = 1; i < 8; i++) {

    console.log(i);

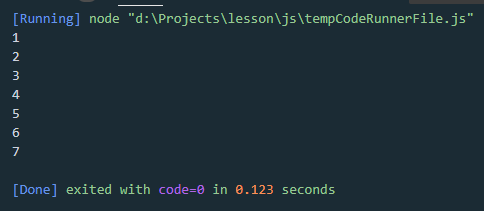
}

i = переменная интератор;

i < 8 = условие

i ++ = инкримент

console.log() = вывод



**Цикл FOR + break**

**#4**

for (let i = 1; i < 10; i++) {

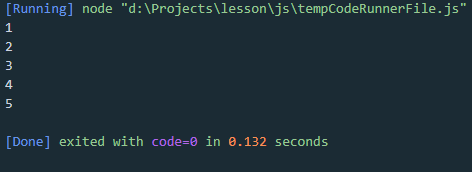
    if (i === 6) {

        break;

    }

    console.log(i);

}



**Цикл FOR + continue**

**#5**

Позволяет пропустить шаг в нашем случае “6”, но цикл не прерывает

for (let i = 1; i < 10; i++) {

    if (i === 6) {

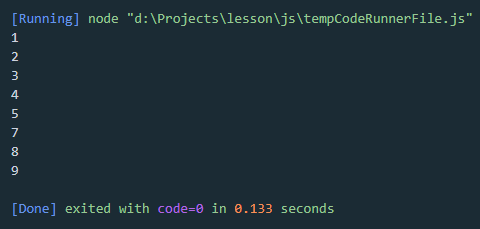
        //break;

        continue;

    }

    console.log(i);

}



**Вложеность циклов**

Пример

**#1**

for (i = 0; i < 3; i++) {

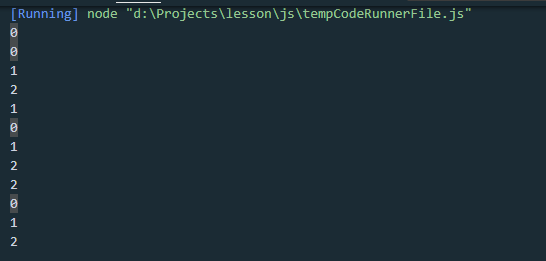
    console.log(i);

    for (j = 0; j < 3; j++) {

        console.log(j);

    }

}



**#2 Задача треугольник со звездочек**

let result = '';

const length = 7;

for (let i = 1; i < 7; i++) {

    for (let j = 0; j < i; j++) {

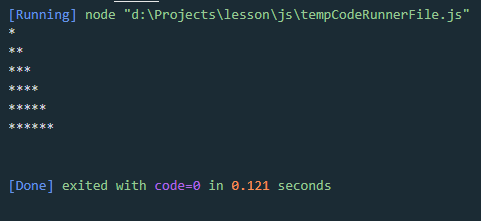
        result += "\*";

    }

    result += '\n'

}

console.log(result);



**Break and continue и метки**

**Break**

first: for (i = 0; i < 3; i++) {

    console.log(`First level: ${i}`);

    for (j = 0; j < 3; j++) {

        console.log(`Second level: ${j}`);

        for (k = 0; k < 5; k++) {

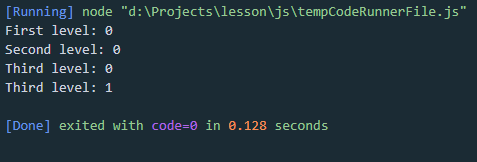
            if (k === 2) break first;

            console.log(`Third level: ${k}`);

            }

    }

}



**continue**

first: for (i = 0; i < 3; i++) {

    console.log(`First level: ${i}`);

    for (j = 0; j < 3; j++) {

        console.log(`Second level: ${j}`);

        for (k = 0; k < 5; k++) {

            if (k === 2) continue first;

            console.log(`Third level: ${k}`);

            }

    }

}

