Topic/Title:
Collections: Working with JavaScript Arrays
Adding Elements and Intermediate Array Techniques
Control Statements: While Loops
Control Statements: For Loops

Keywords/Questions:

array.length;

array.includes(item);

array.push(item);

array.pop;

while(codition){//Do sth}

for(i=0; i<2; ++i{//Do sth}

for(var i=0; i<2; ++i){//Do sth}

Notes:

```
var myEgg = eggs[1];

var eggs = [ 🥚, 🥚, 🥚, 🥚, 🥚 ]
```
-an array

eggs.length;

   -tells you the length of an array

   -this returns the number 5 because there are 5 items inside the array

```
eggs.includes(🥚)
```
eggs.inludes(item1)

   -We get a boolean returned.

```
eggs.push(    );
```
   - .push(item1); :- adds an element to the end of the list.

```
eggs.pop;
```
   - .pop; :-removes the last array element

jitbit.com/alexblog/249-now-thats-what-i-call-a-hacker/ :- a Russian hacker who automated his life.

```
while(condition){              var i = 1;

    //commands               while(i<2) {
                                 console.log(i);
                                 i++;
}                            }
```

Summary:

array.length; :-used to find length of an array

array.includes(item) :-return a boolean after checking if the item is inside the array.

array.push(item); :-adds an item to the end of an array.

array.pop; :- removes the last item of the array

while(condition){//Do sth} :- a while loop.

for(i=0; i<2; ++i){//Do sth} === for(var i=0; i<2;++i){//Do sth} - Both are equivalent for loops.

draw.io :- used to create flow charts.

For Loops

　－The above for loop is the same with for(**var i=0;** i<2; i++) {//Do sth}.

www.draw.io

　－good website for drawing up flow charts.

　－For example, let us draw a flow chart for a fibonacciGenerator

function.

```
function fibonacciGenerator (n) {
    if(n==1){
        return [0];
    }else if(n==2){
        return [0, 1];
    }
    var output=[0, 1];
    for(i=2; i<n; ++i){
        output.push(output[i-1]+output[i-2]);
    }
    return output;
}
```