



# Lesson 11: Nesting and Problem Solving

Let's solve some problems!

Download the Lesson 11 file, unzip, and drag it to your desktop.

---

## Nesting... like Russian dolls

- All of the structures we've learned can fit inside each other
  - Not functions, they are like the biggest doll
- This is called “nesting” a structure inside another





## Why is nesting useful?

- Many times, we'll only want to execute a loop under certain conditions
  - Solve this by placing a loop inside of an if statement
- Conversely, we may only want to execute the body of a loop under certain conditions
  - Solve this by placing an if statement inside of a loop
- Powerful, but can make your code more confusing
  - Always double check that nesting is necessary
- Unnecessary nesting - needing to apply two conditions at once
  - Nested if statements? No! Use the && operator

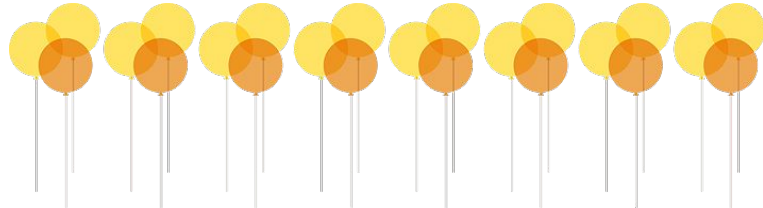
```
for(let i = 0; i < 10; i++) {  
  if (/* condition */) {  
  
  }  
}
```

```
if (/* condition */) {  
  for(let i = 0; i < 10; i++) {  
  
  }  
}
```



# Example

- Print every third number (i.e. 0, 3, 6, 9...) from 0-100
- Two different avenues
  - Nesting if and for loops
- What's another solution?
  - Hint: no if statement!





## Solution 1

- Nesting!
  - For loop on the outside
  - Nested if statement checks whether to print the number

```
for(let i = 0; i < 100; i++) {  
  if (i%3==0) {  
    console.log(i);  
  }  
}
```



## Solution 2

- Increment the loop by 3 each time!

```
for(let i = 0; i < 100; i = i + 3) {  
  if (i%3==0) {  
    console.log(i);  
  }  
}
```



# Task #1

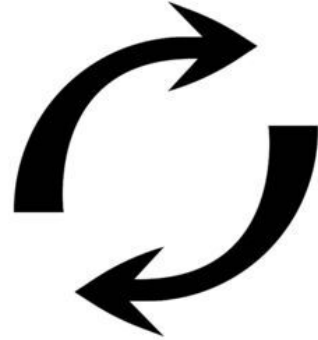
Let's tackle a similar problem!

- Open up Lesson11/main.js in Atom
  - Follow the instructions
- Take 10 minutes - get as far as you can
- Send questions in the chat
- Remember to invoke the task1 function

---

## Now, let's reverse it!

- Let's say we only want to execute different loops depending on some condition
- Concretely, if given an array of numbers and a boolean...
  - Print the sum of the first half of the array if the boolean is true
  - Print the sum of the second half if false





## Solution

- Two different loops with different bounds!

```
function numCounter(nums, first_half) {
  let sum = 0;
  if (first_half) {
    for(let i = 0; i < nums.length/2; i++) {
      sum = sum + nums[i];
    }
  } else {
    for(let i = nums.length/2; i < nums.length; i++) {
      sum = sum + nums[i];
    }
  }
  console.log(sum);
}
```

## Last combo...

- Nested for loops
  - Lots of applications in math/science
  - How to navigate matrices, any 2d structure
- Need to have different names for index variables!
  - $i$  and  $j$  are the most common
- Let's see an example

$$\begin{matrix} & \begin{matrix} 1 & 2 & \dots & n \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ \vdots \\ m \end{matrix} & \left[ \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{array} \right] \end{matrix}$$



# Example

- Suppose we have an array of arrays (a matrix)
- We want to add up all the values
- What would we do?
  - One big loop to hit each element of the outer array
  - Inner loop for each smaller array of numbers



## Solution

```
function matrix(input) {  
  let sum = 0;  
  for(let i = 0; i < input.length; i++) {  
    let subarray = input[i];  
    for(let j = 0; j < subarray.length; j++) {  
      sum = sum + subarray[j];  
    }  
  }  
  console.log(sum);  
}
```



## Task #2 (Our last task!)

This is very similar to what we just did, just a few changes

- Take 10 minutes - get as far as you can, then we will discuss
- Send questions in the chat
- Remember to invoke the task2 function



# Review

3 types of nesting:

- For inside of if
- If inside of for
- For inside of for

These are just a few possible/common scenarios. The possibilities are endless!



# Wrap-Up

Thank you all so much!

Keep in touch, happy to help guide you to further material, or give info/referrals about programs in Wilmington

Coding is hard, but so worth it!

Good luck.