

The **for** Loop

while vs for Loops

```
{  
  let i = 0;  
  while (i < n) {  
    // repeat block  
    i++;  
  }  
}
```

```
for (let i = 0; i < n; i++) {  
  // repeat block  
}
```

- These loops are exactly equivalent to one another, with two important notes:
 1. The **i++** step in the **for** loop is evaluated at the very end of the repeat block... just like where it is positioned in the **while** loop.
 2. In the **for** loop, the variable **i** is declared and scoped *inside* of the **for** loop statement and is not visible after the statement completes.

The **for** Loop Statement

- General form:

```
for ( 1 <variable initialization> ; 2 <boolean test> ; 4 <variable modification> ) {  
    <repeat block> 3  
}
```

1. Counter variable is initialized

2. Boolean test is evaluated

True? – 3. Repeat block is entered and runs.

4. *Then*, counter variable modified.

Finally, loop back to step #2.

False? – 5. Skip repeat block and loop is complete.

Example Setup

In VSCode:

1. Start the Development Server
 - View Terminal
 - `npm run pull`
 - `npm start`
2. Open the File Explorer Pane
 - Right click on the src folder
 - Select "New folder"
 - Name it: **x-for-loop**
 - Right click on the x-for-loop folder
 - Select "New file"
 - Name it: **for-loop-app.ts**
3. In `for-loop-app.ts`, write out the code to the right. It has no errors, so review carefully if yours has any.

```
import { print, promptNumber } from "intros";  
  
export let main = async () => {  
    let n = await promptNumber("How many?");  
  
    for (let i = 0; i < n; i++) {  
        print(i);  
    }  
};  
  
main();
```

Why use a **for** loop?

- Special syntax for the common while loop pattern using a counter variable
 - *But to the computer, each is exactly the same!*
- For us as humans, the **for** loop syntax has two benefits:
 1. You are *much* less likely to accidentally write an infinite loop
 2. The counter variable is only defined within the for-loops repeat block
 - Kind of like a function's parameter is only accessible inside of the function body.
 - This means you can have a sequence of for loops that each use, say *i*, as the counter variable.
- Generally, once the syntax is familiar, **for**-loops are less human-error prone