



The Core4: Loops

What are loops?

- A loop is a coding structure that asks the computer to complete a set of instructions until a condition is reached.
- If someone asks you “Please bring all of the pizzas from the car into the house until there no more”, that is actually a loop!
 - Instructions for each pizza: bring it from the car into the house.
 - Terminating Condition: If there are no more pizzas, stop.
- If you want to buy an interesting book, that can be written as a loop
 - Instructions for each book: For each book in a shelf, shift through the first page. If the book is boring, put it back. Otherwise, pick up another book and repeat the instructions.
 - Terminating Condition: If the book is interesting and it is in your budget, buy the book.



Why are loops important?



- Loops allow you to reuse code to make the same object many times or to apply the exact same process to many objects.
- You may use a loop if:
 - You want to loop through ten buttons in a webpage, and make each button blue.
 - If you want to loop through all the numbers entered in a calculator and add them together.
 - If you want to loop through a player's seven cards and print "Game Over!" if they have the joker card in their hand.
- What other ways might a loop be useful for?

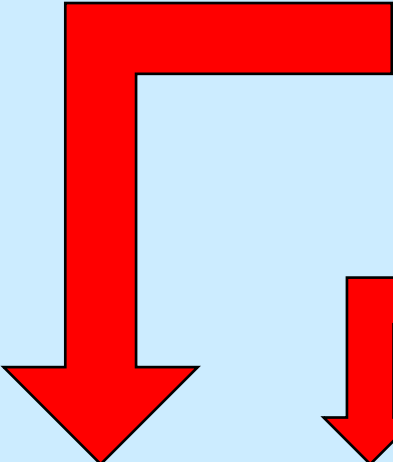
Loop Commands/Vocab

- **Iteration** – a single repetition of a loop. (**Example:** Bringing in one pizza into the house.)
- **Continue** – a command that tells the computer to skip an iteration and go to the next one. (**Example:** If a pizza has the wrong toppings, do not bring it into the house and skip to the next pizza's instructions)
- **Break** – a command that allows the programmer to exit out of the entire loop (**Example:** If you are tired of bringing in pizzas, stop bringing in pizzas)


Types of Loops

- The two most common types of loops are “**for**” loops and “**while**” loops
 - “**For**” loops – continue iterations based on a number of times suggested.
 - Example: A number starts at zero. For each iteration in which the number is between 0 and 10, print out “Coding is fun!” and increase the number by 1.
 - “**While**” loops – continue performing iterations until a certain condition is reached.
 - Example: While a number is less than a hundred, print out the number and increase it by one.

Anatomy of a “For” loop

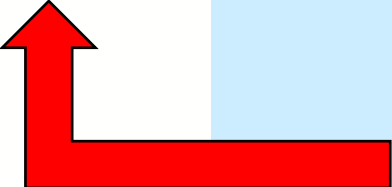


This part means that we are going to start our number, named “i”, at zero.



This part means that we will continue the “for” loop until i is less than 10 is false. In other words, the loop will stop when i = 10 because 10 IS NOT less than 10.

```
2  for (i = 0; i < 10; i++){  
3  console.log(i);  
4  }
```

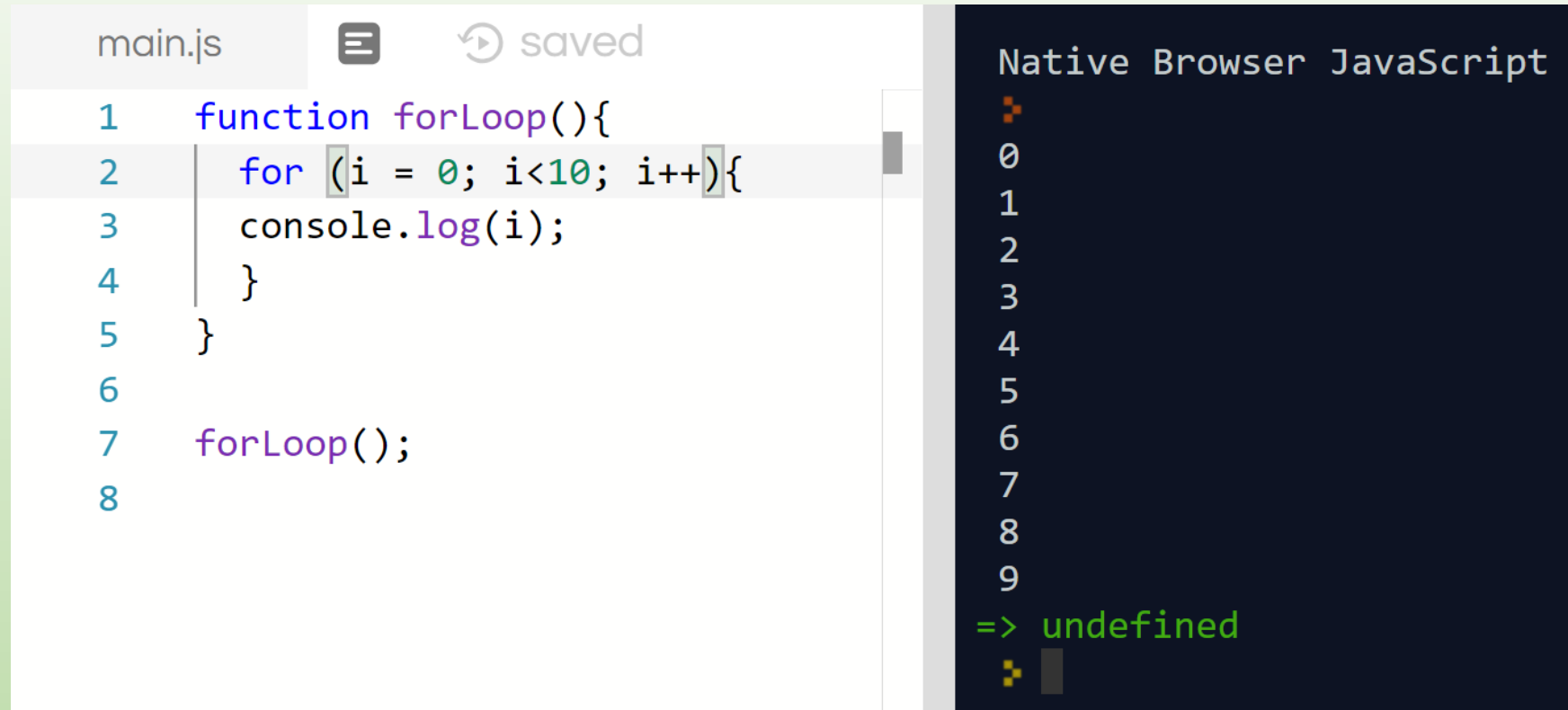


This part means that the number will be automatically be incremented after each iteration



This part is the instructions for each iteration!
In this case, we will simply be printing out i.

“For” Loops in Action



The image shows a code editor window with a file named 'main.js' and a 'saved' status. The code defines a function 'forLoop()' that uses a 'for' loop to iterate from 0 to 9, logging each value to the console. The function is then called. To the right, a browser console window shows the output of the script, displaying the numbers 0 through 9 on separate lines, followed by '=> undefined'.

```
main.js saved
1 function forLoop(){
2   for (i = 0; i<10; i++){
3     console.log(i);
4   }
5 }
6
7 forLoop();
8
```

Native Browser JavaScript

```
0
1
2
3
4
5
6
7
8
9
=> undefined
```

- Why is the first number being printed out a zero instead of a one?
- Why is nine the last number printed out?
- If I wanted to make the “for” loop have the first number printed be one, how would I change the loop instead?

While Loops

```
var i = 0;
```

The number should be declared before the “while” loop starts.

```
while (i < 10)
```

This line asks the “while” loop to continue while the number is less than 10.

```
{  
  console.log(i);  
  i++;  
}
```

These two lines are the instructions within the “while” loop. For each iteration, the computer will follow the instructions until the “while” loop stops.

Before moving on the next slide, which numbers do you predict will be printed out?


```
1  function forLoop(){
2      var i = 0;
3
4      while (i < 10)
5      {
6          console.log(i);
7          i++;
8      }
9  }
10
11  forLoop();
```

```
0
1
2
3
4
5
6
7
8
9
=> undefined
```

- If variable *i* started at 5, what numbers would be printed out?
- **Bonus:** What would happen if we took out the line *i++* from the loop?

Hint: Remember that *i++* means “*increment i by one*”

FizzBuzz!

- We are going to try our hand today at one of the most popular interview questions for developers today! See if you can solve it!
- Program instructions:
- Go through each number from 1 to 25. If a number is divisible by 3, print “Fizz”. If the same number is divisible by 5, print “Buzz”. If the same number is divisible by three and five, print “FizzBuzz”. Otherwise, simply print the number.
- **Major Hint:** Remember that a number is divisible by another if it's remainder is zero. In other words $10 \% 5 == 0$ means ten divided by five has a remainder of zero, so ten is divisible by five.

Here is a possible solution to the Fizzbuzz challenge

1st Note: To find a number divisible by 5 AND 3, you need to check both statements.

2nd Note: In the case of a while, loop, the number must always be incremented to continue the loop. Otherwise, a loop that will NEVER terminate will occur

```
1  function fizzBuzz(){
2  var currentNum = 1;
3  while (currentNum<26){
4      if (currentNum%3==0 && currentNum%5==0){
5          console.log("fizzBuzz");
6      }
7      else if (currentNum%5==0){
8          console.log("Buzz");
9      }
10     else if (currentNum%3==0){
11         console.log("Fizz");
12     }
13     else{
14         console.log(currentNum);
15     }
16     currentNum++;
17 }
18 }
19 }
20 fizzBuzz();
21
```

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
fizzBuzz
16
17
Fizz
19
Buzz
Fizz
22
23
Fizz
Buzz
```



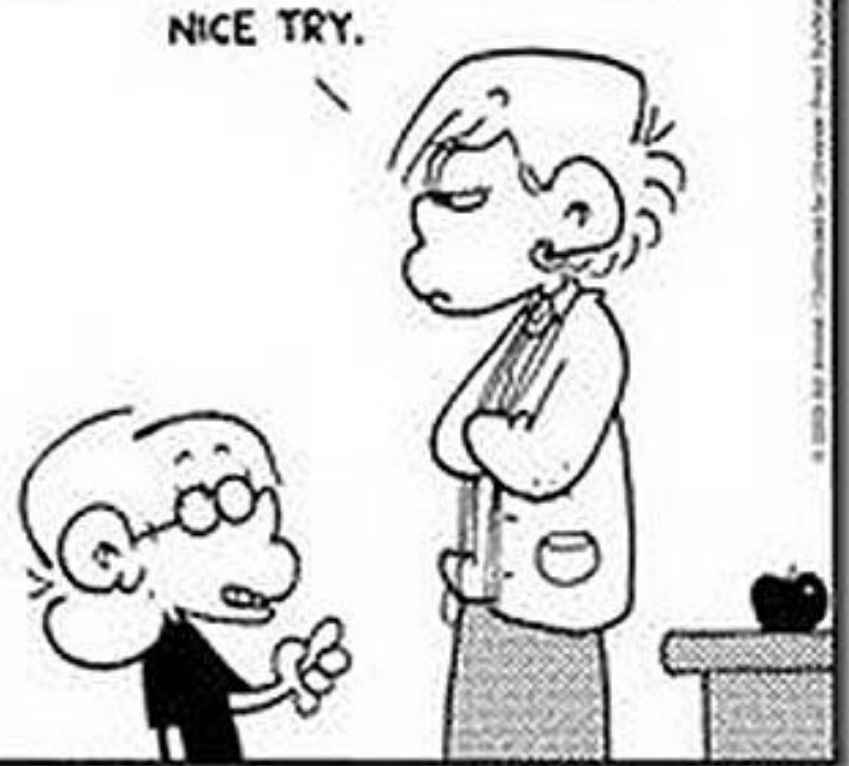

Quick Reflection



- Were there any coding problems that you ran into today? How did you solve them?
- What are some tips you can give your classmates when coding solutions?
- What is something you learned in this lesson that might be useful next time?

```
#include <stdio.h>
int main(void)
{
    int count;

    for (count = 1; count <= 500; count++)
        printf("I will not throw paper airplanes in class.");
    return 0;
}
```



Have a great weekend!