Fullstack Academy

Block 14: Loops and Arrays

["we", "will", "learn", "about", "loops", "and", "arrays"]

Agenda

- Demonstration: Fruit Salad
- Guided Practice: Instruments
- Workshop: Statistics

- On your own:
  - Take competency check
  - Finish Workshop: Statistics

Learning Teams: Connecting

Open the Block 14: Loops and Arrays page in Canvas and be prepared to discuss the Learning Team section with your Unit 2 team.

Learning Objectives

Where are we going?

After today's lesson, you should be able to:

- Define how arrays are used to organize data.
- Use indexes to access and modify individual elements of an array.
- Read and write code that uses loops.
- Use loops to iterate through an array.
- Distinguish between for and while loops.

What’s In It for Me? (WIIFM?)

Simple Code
+
Efficiency

Arrays

An **array** is a data structure stores multiple *values* in a single *variable*.

```
const vegetables = ["carrots", "cabbage", "lettuce"];

const numbers = [1,3,5,7,9];
```

Arrays

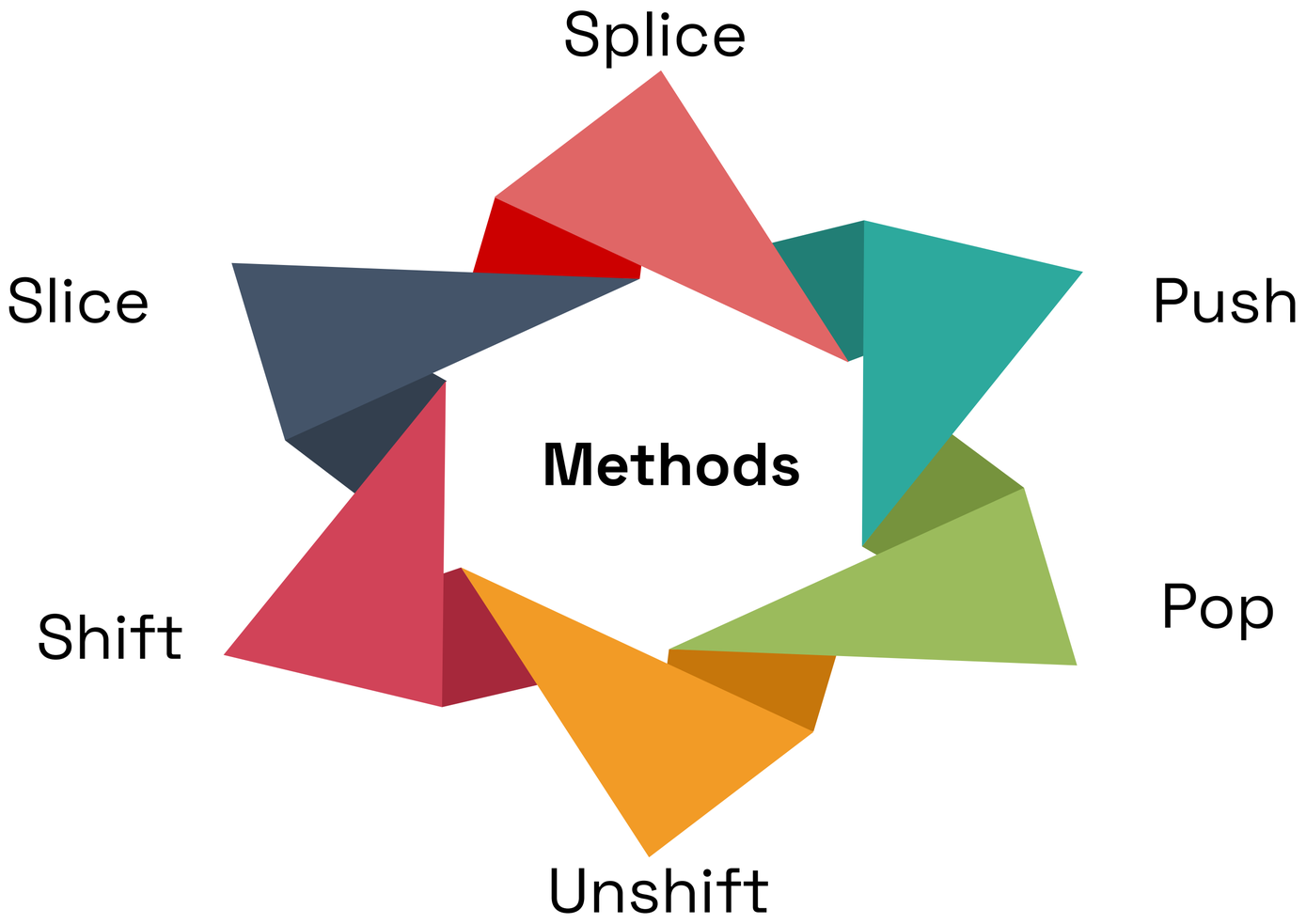
Instead of storing **each** vegetable in a **single** variable:

```
const carrots = "carrots";
const cabbage = "cabbage";
```

We can store a lot more data in **one** variable:

```
const vegetables = ["carrots", "cabbage", "lettuce","celery"];
```

Array Manipulation



Example

```
// Example: Push

let array_number =
[10,20,30,40]
array_number.push(50)
console.log(array_number)
//Output: array_number =
[10,20,30,40, 50]
```

```
// Example: Pop

let array_number = [10,20,30]
array_number.pop()
console.log(array_number)

//Output: array_numbers =
[10,20]
```

Example

```
//Example: Unshift

let numbers = [7,8,9]
numbers.unshift(100)
console.log(numbers)
//Output numbers =
[100,7,8,9]
```

```
//Example: Shift

let number = [10,20,30]
number.shift()
console.log(number)
//Output numbers = [20,30]
```

Example

Example: Slice

```
let numbers = [1, 2, 3, 4];
let slicedNumbers = numbers.slice(1, 3);
console.log(slicedNumbers)


//Output numbers = [2,3]
```

Example: Splice

```
let numbers = [1, 2, 3];
let removedNumbers =
numbers.splice(0,1);
console.log(numbers);
console.log(removedNumbers);
numbers.splice(0, 1, 10, 20);
console.log(numbers);
//Output numbers =
(2) [2, 3]
[1]
(3) [10, 20, 3]
```

What is a loop?

Loops offer a quick and easy way to do something repeatedly.

You can use them to:

- Traverse over a large set of data.
- Repeat an action over the items in an array.
- Do something until a certain condition is met.

There are various types of loops, but we will focus on two today

for and **while**

Types of Loops

For loops
loop through a block of code a **number** of times.

While loops
loop through a block of code while a **specific condition** is true.

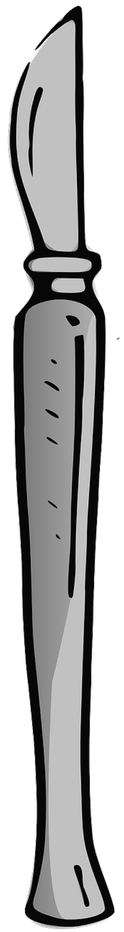
For Loop

A for loop statement is created with **three** expressions.

Notice the use of parentheses, semicolons, and curly braces

```
for(initialization; condition; afterthought){
// statement;
}
```

Dissection of a For Loop



```
for(initialization; condition; afterthought){
// statement;
}
```

- **initialization** = executed once before the execution
- **condition** = defines the condition for executing the code block.
- **afterthought** = executed after the code block finishes.

For Loop Dissected

```
for(let i = 0; i < 10; i++){
   console.log(`Step ${i+1}`);
}
```

Print to the Terminal:

Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Step 7
Step 8
Step 9
Step 10

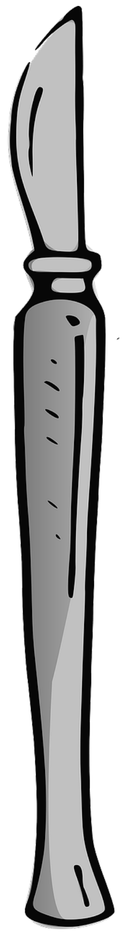
While Loop

While loops = loop through a block of code as long as a **specified condition** is met.

Note the differences in the amount of required expressions

```
while (condition) {
// code block to be executed
}
```

Dissection of a While Loop

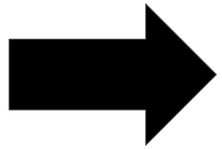


```
let num = 5;
while(num < 10){
  console.log("our current number is " + num);
  num++;
}
console.log("end of loop");
```

While our num variable is less than 10, we will console log the sentence on line 3. Afterward, we would increase our num variable by one until our num variable reaches 10. We then log the final sentence.

While Loop Dissected

```
let num = 5;
while(num < 10){
  console.log("our current number is " + num);
  num++;
}
console.log("end of loop");
```



```
Our current number is 5
Our current number is 6
Our current number is 7
Our current number is 8
Our current number is 9
end of loop
```

Demonstration
Fruit Salad

Guided Practice

- Open and preview the guided practice tab, found in the Block 14: Loops and Arrays page in Canvas.
- Complete guided practice, then come back with questions before beginning the workshop.

Learning Objectives

Where have we been?

After today's lesson, you should be able to:

- Define how arrays are used to organize data.
- Use indexes to access and modify individual elements of an array.
- Read and write code that uses loops.
- Use loops to iterate through an array.
- Distinguish between for and while loops.

Block 14: Workshop

- Open and preview Block 14 Workshop: Statistics.

Learning Teams: Concluding

Open the Block 14: Concluding page in Canvas and be prepared to discuss and share with your learning team.