## Different Set of Operation:

```
<!DOCTYPE html>
<html>
<head>
   <title>Set Operation</title>
</head>
<body>
<script>
// Union
function union(setA, setB) {
   let unionSet = new Set(setA);
   for (let elem of setB) {
      unionSet.add(elem);
   }
   return unionSet;
}

// Intersection
function intersection(setA, setB) {
   let intersectionSet = new Set();
   for (let elem of setB) {
      if (setA.has(elem)) {
         intersectionSet.add(elem);
      }
   }
   return intersectionSet;
}

// Difference
function difference(setA, setB) {
   let differenceSet = new Set(setA);
```

```
    for (let elem of setB) {
        differenceSet.delete(elem);
    }
    return differenceSet;
}

// Symmetric Difference
function symmetricDifference(setA, setB) {
    let differenceSet = new Set(setA);
    for (let elem of setB) {
        if (differenceSet.has(elem)) {
            differenceSet.delete(elem);
        } else {
            differenceSet.add(elem);
        }
    }
    return differenceSet;
}

// Test Sets
const setA = new Set(['apple', 'banana', 'cherry']);
const setB = new Set(['banana', 'date', 'fig']);

// Output results in console
console.log("Union:", union(setA, setB)); // Output: Set { 'apple', 'banana', 'cherry', 'date', 'fig' }
console.log("Intersection:", intersection(setA, setB)); // Output: Set { 'banana' }
console.log("Difference:", difference(setA, setB)); // Output: Set { 'apple', 'cherry' }
console.log("Symmetric Difference:", symmetricDifference(setA, setB)); // Output: Set { 'apple', 'cherry', 'date', 'fig' }
</script>
</body>
```

</html>