|  |
| --- |
| ASSigmnet 2-Section-1 |
|  |  |
|  | 1. |
|  | <!DOCTYPE html> |
|  | <html> |
|  | <body> |
|  | <script> |
|  | function composedValue(square,double,n) |
|  | { |
|  | return square(double(n)); |
|  | } |
|  | function square(n) |
|  | { |
|  | return n\*n; |
|  | } |
|  | function double(n) |
|  | { |
|  | return n\*2; |
|  | } |
|  | var n = parseInt(prompt("Enter a number: ")); |
|  | document.write(composedValue(square,double,n)); |
|  | </script> |
|  | </body> |
|  | </html> |
|  |  |
|  | 2. |
|  | <!DOCTYPE html> |
|  | <html> |
|  | <body> |
|  | <script> |
|  | var f1 = function func1(n) |
|  | { |
|  | function compose(square,double,n) |
|  | { |
|  | return square(double(n)); |
|  | } |
|  | return compose(square,double,n) |
|  | } |
|  | var f2 = function func2(n) |
|  | { |
|  | function compose(double,square,n) |
|  | { |
|  | return double(square(n)); |
|  | } |
|  | return compose(double,square,n) |
|  | } |
|  | function square(n) |
|  | { |
|  | return n\*n; |
|  | } |
|  | function double(n) |
|  | { |
|  | return n\*2; |
|  | } |
|  | document.write(f1(5)); |
|  | document.write("<br>"+f1(10)); |
|  | document.write("<br>"+f2(5)); |
|  | document.write("<br>"+f2(10)); |
|  | </script> |
|  | </body> |
|  | </html> |
|  |  |
|  | 3. |
|  | <!DOCTYPE html> |
|  | <html> |
|  | <body> |
|  | <script> |
|  | function isEven(num) |
|  | { |
|  | return num%2 == 0; |
|  | } |
|  | var arr = [1,3,5,4,2]; |
|  | function find(arr, isEven) |
|  | { |
|  | for(let i=0;i<arr.length;i++) |
|  | { |
|  | if(isEven(arr[i]) == true) |
|  | { |
|  | return arr[i]; |
|  | } |
|  | } |
|  | } |
|  | document.write(find(arr,isEven)); |
|  | </script> |
|  | </body> |
|  | </html> |
|  |  |
|  | 4. |
|  |  |
|  | <!DOCTYPE html> |
|  | <html> |
|  | <body> |
|  | <script> |
|  | function mapValues(arr, func) |
|  | { |
|  | return arr.map(func) |
|  | } |
|  | function square(n) |
|  | { |
|  | return n\*n; |
|  | } |
|  | document.write(mapValues([1,2,3,4,5],square)); |
|  | document.write("<br>"+mapValues([1,4,9,16,25],Math.sqrt)); |
|  | </script> |
|  | </body> |
|  | </html> |
|  |  |
|  |  |
|  | SECTION 2 |
|  |  |
|  | 1. |
|  |  |
|  | <!DOCTYPE html> |
|  | <html> |
|  | <body> |
|  | <script> |
|  | function isEven(num) { return(num%2 == 0); } |
|  | var arr = [1, 3, 5, 4, 2]; |
|  |  |
|  | function findRecursive(arr, func) { |
|  | var p = arr.shift(); |
|  | if (func(p) == true) |
|  | return p; |
|  | else |
|  | return findRecursive(arr, func); |
|  | } |
|  | document.write(findRecursive([1, 3, 5, 4, 2], isEven)); |
|  | document.write(findRecursive([1, 3, 5], isEven)); |
|  | </script> |
|  | </body> |
|  | </html> |
|  |  |
|  | 2. |
|  |  |
|  | <!DOCTYPE html> |
|  | <html> |
|  | <body> |
|  | <script> |
|  | var recursiveMapSquare = function(arr, func) |
|  | { |
|  | if(arr.length === 1) |
|  | { |
|  | return func(arr); |
|  | } |
|  | else |
|  | { |
|  | return [func(arr[0])].concat(recursiveMapSquare(arr.slice(1),func)) |
|  | } |
|  | }; |
|  |  |
|  | var arr = [1,2,3,4,5]; |
|  | var square = function(s) |
|  | { |
|  | return s\*s; |
|  | } |
|  | var arr2 = [1,4,9,16,25]; |
|  |  |
|  | var result = recursiveMapSquare(arr, square); |
|  | document.write("Original Array: ",arr); |
|  | document.write("<br>Squared Array: ",result); |
|  |  |
|  | var result2 = recursiveMapSquare(arr2, Math.sqrt); |
|  | document.write("<br><br>Original Array: ",arr2); |
|  | document.write("<br>Square Rooted Array: ",result2); |
|  | </script> |
|  | </body> |
|  | </html> |