

# Recursion II

---



# Example: logsAnArray

```
1 function logsAnArray(array) {
2   for (let i = 0; i < array.length; i++) {
3     let element = array[i];
4     console.log(element);
5   }
6 }
7
8 logsAnArray([1, [2, 3]]); // what if the input is nested?
9 /* consider element on line 3. that could be a number, or it could be
10    another array! */
11
12 /* if element is an array, we want to log every number inside of it. if
13    only we had a function that takes an array and logs out all of its
14    values! oh wait... */
```

# Example: logsAnArray

logsAnArray([2, 3])

logsAnArray(1, [2, 3])

```
1  function logsAnArray(array) {  
2    for (let i = 0; i < array.length; i++) {  
3      let element = array[i];  
4  
5      if (Array.isArray(element)) {  
6        logsAnArray(element);  
7      }  
8      else {  
9        console.log(element);  
10     }  
11   }  
12 }  
13  
14 logsAnArray([1, [2, 3]]);
```



# Example: concatEls

Take an array of elements(letters), and concatenate them together

*//['a', 'b', 'c'] ⇒ abc*

```
1 function concatEls(array) {
2   let finalString = "";
3   for (let i = 0; i < array.length; i++) {
4     let element = array[i];
5     finalString += element;
6   }
7   return finalString;
8 }
9
10 let result = concatEls(['a', 'b', 'c']);
11 console.log(result);
12
13
14
```

# Example: concatEls (nested)

*//['a', ['b', 'c']] ⇒ abc*

```
1  function concatEls(array) {  
2    let finalString = "";  
3    for (let i = 0; i < array.length; i++) {  
4      let element = array[i];  
5      if (Array.isArray(element)) {  
6        finalString += concatEls(element);  
7      } else {  
8        finalString += element;  
9      }  
10   }  
11   return finalString;  
12 }  
13 let result = concatEls(['a', ['b', 'c']]);  
14 console.log(result);
```

# Example: concatEls

concatEls(['a', ['b', 'c']])

"

```
1  function concatEls(array) {  
2    let finalString = "";  
3    for (let i = 0; i < array.length; i++) {  
4      let element = array[i];  
5      if (Array.isArray(element)) {  
6        finalString += concatEls(element);  
7      } else {  
8        finalString += element;  
9      }  
10   }  
11   return finalString;  
12 }  
13 let result = concatEls(['a', ['b', 'c']]);  
14 console.log(result);
```

# Example: concatEls

concatEls(['a', ['b', 'c']])

" += 'a'

```
1 function concatEls(array) {  
2   let finalString = "";  
3   for (let i = 0; i < array.length; i++) {  
4     let element = array[i];  
5     if (Array.isArray(element)) {  
6       finalString += concatEls(element);  
7     } else {  
8       finalString += element;  
9     }  
10  }  
11  return finalString;  
12 }  
13 let result = concatEls(['a', ['b', 'c']]);  
14 console.log(result);
```

# Example: concatEls

concatEls(['a', ['b', 'c']])

'a'

```
1 function concatEls(array) {  
2   let finalString = "";  
3   for (let i = 0; i < array.length; i++) {  
4     let element = array[i];  
5     if (Array.isArray(element)) {  
6       finalString += concatEls(element);  
7     } else {  
8       finalString += element;  
9     }  
10  }  
11  return finalString;  
12 }  
13 let result = concatEls(['a', ['b', 'c']]);  
14 console.log(result);
```



# Example: concatEls

concatEls(['a', ['b', 'c']])

'a' += concatEls(['b', 'c'])

```
1 function concatEls(array) {  
2   let finalString = "";  
3   for (let i = 0; i < array.length; i++) {  
4     let element = array[i];  
5     if (Array.isArray(element)) {  
6       finalString += concatEls(element);  
7     } else {  
8       finalString += element;  
9     }  
10  }  
11  return finalString;  
12 }  
13 let result = concatEls(['a', ['b', 'c']]);  
14 console.log(result);
```

# Example: concatEls

concatEls(['b', 'c'])

"

concatEls(['a', ['b', 'c']])

'a' += concatEls(['b', 'c'])

```
1 function concatEls(array) {
2   let finalString = "";
3   for (let i = 0; i < array.length; i++) {
4     let element = array[i];
5     if (Array.isArray(element)) {
6       finalString += concatEls(element);
7     } else {
8       finalString += element;
9     }
10  }
11  return finalString;
12 }
13 let result = concatEls(['a', ['b', 'c']]);
14 console.log(result);
```

# Example: concatEls

`concatEls(['b', 'c'])``" += 'b'``concatEls(['a', ['b', 'c']])``'a' += concatEls(['b', c'])`

```
1 function concatEls(array) {  
2   let finalString = "";  
3   for (let i = 0; i < array.length; i++) {  
4     let element = array[i];  
5     if (Array.isArray(element)) {  
6       finalString += concatEls(element);  
7     } else {  
8       finalString += element;  
9     }  
10  }  
11  return finalString;  
12 }  
13 let result = concatEls(['a', ['b', 'c']]);  
14 console.log(result);
```

# Example: concatEls

concatEls(['b', 'c'])

'b'

concatEls(['a', ['b', 'c']])

'a' += concatEls(['b', 'c'])

```
1 function concatEls(array) {
2   let finalString = "";
3   for (let i = 0; i < array.length; i++) {
4     let element = array[i];
5     if (Array.isArray(element)) {
6       finalString += concatEls(element);
7     } else {
8       finalString += element;
9     }
10  }
11  return finalString;
12 }
13 let result = concatEls(['a', ['b', 'c']]);
14 console.log(result);
```

# Example: concatEls

concatEls(['b', 'c'])

'b' += 'c'

concatEls(['a', ['b', 'c']])

'a' += concatEls(['b', 'c'])

```
1 function concatEls(array) {
2   let finalString = "";
3   for (let i = 0; i < array.length; i++) {
4     let element = array[i];
5     if (Array.isArray(element)) {
6       finalString += concatEls(element);
7     } else {
8       finalString += element;
9     }
10  }
11  return finalString;
12 }
13 let result = concatEls(['a', ['b', 'c']]);
14 console.log(result);
```

# Example: concatEls

concatEls(['b', 'c'])

'bc'

concatEls(['a', ['b', 'c']])

'a' += concatEls(['b', 'c'])

```
1 function concatEls(array) {  
2   let finalString = "";  
3   for (let i = 0; i < array.length; i++) {  
4     let element = array[i];  
5     if (Array.isArray(element)) {  
6       finalString += concatEls(element);  
7     } else {  
8       finalString += element;  
9     }  
10  }  
11  return finalString;  
12 }  
13 let result = concatEls(['a', ['b', 'c']]);  
14 console.log(result);
```

# Example: concatEls

`concatEls(['b', 'c'])``=> 'bc'``concatEls(['a', ['b', 'c']])``'a' += concatEls(['b', 'c'])`

```
1 function concatEls(array) {
2   let finalString = "";
3   for (let i = 0; i < array.length; i++) {
4     let element = array[i];
5     if (Array.isArray(element)) {
6       finalString += concatEls(element);
7     } else {
8       finalString += element;
9     }
10  }
11  return finalString;
12 }
13 let result = concatEls(['a', ['b', 'c']]);
14 console.log(result);
```

# Example: concatEls

concatEls(['a', ['b', 'c']])

'a' += 'bc'

```
1 function concatEls(array) {  
2   let finalString = "";  
3   for (let i = 0; i < array.length; i++) {  
4     let element = array[i];  
5     if (Array.isArray(element)) {  
6       finalString += concatEls(element);  
7     } else {  
8       finalString += element;  
9     }  
10  }  
11  return finalString;  
12 }  
13 let result = concatEls(['a', ['b', 'c']]);  
14 console.log(result);
```



# Example: concatEls

concatEls(['a', ['b', 'c']])

'abc'

```
1 function concatEls(array) {
2   let finalString = "";
3   for (let i = 0; i < array.length; i++) {
4     let element = array[i];
5     if (Array.isArray(element)) {
6       finalString += concatEls(element);
7     } else {
8       finalString += element;
9     }
10  }
11  return finalString;
12 }
13 let result = concatEls(['a', ['b', 'c']]);
14 console.log(result);
```

# Example: concatEls

concatEls(['a', ['b', 'c']])

=> 'abc'

```
1 function concatEls(array) {  
2   let finalString = "";  
3   for (let i = 0; i < array.length; i++) {  
4     let element = array[i];  
5     if (Array.isArray(element)) {  
6       finalString += concatEls(element);  
7     } else {  
8       finalString += element;  
9     }  
10  }  
11  return finalString;  
12 }  
13 let result = concatEls(['a', ['b', 'c']]);  
14 console.log(result);
```

# Example: concatEls

```
1  function concatEls(array) {  
2    let finalString = "";  
3    for (let i = 0; i < array.length; i++) {  
4      let element = array[i];  
5      if (Array.isArray(element)) {  
6        finalString += concatEls(element);  
7      } else {  
8        finalString += element;  
9      }  
10   }  
11   return finalString;  
12 }  
13 let result = concatEls(['a', ['b', 'c']]);  
14 console.log(result);
```



# Example: concatEls

```
1 function concatEls(array) {
2   let finalString = "";
3   for (let i = 0; i < array.length; i++) {
4     let element = array[i];
5     if (Array.isArray(element)) {
6       finalString += concatEls(element);
7     } else {
8       finalString += element;
9     }
10  }
11  return finalString;
12 }
13 let result = concatEls(['a', ['b', 'c']]);
14 console.log(result);
```



# Nested objects

```
1  /* working with nested objects is very similar to working with nested
2     arrays */
3
4  /* use a for-in loop to iterate through the object */
5
6  /* if the value is another object, recursively call the function on the
7     nested object */
8
9
10
11
12
13
14
```



# Example: sumVals

```
1  /* write a function sumVals that sums the values in an object */
2
3  function sumVals(obj) {
4    let sum = 0;
5    for (let key in obj) {
6      let value = obj[key];
7      sum += value;
8    }
9    return sum;
10 }
11
12 let result = sumVals({a: 10, b: 20});
13 console.log(result);
14
```



# Example: sumVals

```
1 function sumVals(obj) {
2   let sum = 0;
3   for (let key in obj) {
4     let value = obj[key];
5     if (typeof value === 'object') {
6       sum += sumVals(value)
7     } else {
8       sum += value;
9     }
10  }
11  return sum;
12 }
13
14 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
   console.log(result);
```

# Example: sumVals

sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})

0

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```



# Example: sumVals

sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})

0 += 1

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

# Example: sumVals

sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})

1

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

# Example: sumVals

```
sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})    1 += sumVals({c: {d: {e:2, f: 3}}})
```

```
1  function sumVals(obj) {  
2    let sum = 0;  
3    for (let key in obj) {  
4      let value = obj[key];  
5      if (typeof value === 'object') {  
6        sum += sumVals(value)  
7      } else {  
8        sum += value;  
9      }  
10   }  
11   return sum;  
12 }  
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})  
14 console.log(result);
```

# Example: sumVals

sumVals({c: {d: {e:2, f: 3}}})

0

sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})     1 += sumVals({c: {d: {e:2, f: 3}}})

```
3   for (let key in obj) {
4     let value = obj[key];
5     if (typeof value === 'object') {
6       sum += sumVals(value)
7     } else {
8       sum += value;
9     }
10  }
11  return sum;
12 }
13
14 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
   console.log(result);
```

# Example: sumVals

sumVals({c: {d: {e:2, f: 3}}})

0 += sumVals({d: {e:2, f: 3}})

sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})

1 += sumVals({c: {d: {e:2, f: 3}}})

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

sumVals({d: {e:2, f: 3}})

0

sumVals({c: {d: {e:2, f: 3}}})

0 += sumVals({d: {e:2, f: 3}})

sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})

1 += sumVals({c: {d: {e:2, f: 3}}})

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

# Example: sumVals

sumVals({d: {e:2, f: 3}})

0 += sumVals({e:2, f: 3})

sumVals({c: {d: {e:2, f: 3}}})

0 += sumVals({d: {e:2, f: 3}})

sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})

1 += sumVals({c: {d: {e:2, f: 3}}})

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

# Example: sumVals

Callstack	sum
sumVals({e:2, f: 3})	0
sumVals({d: {e:2, f: 3}})	0 += sumVals({e:2, f: 3})
sumVals({c: {d: {e:2, f: 3}}})	0 += sumVals({d: {e:2, f: 3}})
sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})	1 += sumVals({c: {d: {e:2, f: 3}}})

```

1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);

```

## Example: sumVals



Callstack	sum
sumVals({e:2, f: 3})	0 += 2
sumVals({d: {e:2, f: 3}})	0 += sumVals({e:2, f: 3})
sumVals({c: {d: {e:2, f: 3}}})	0 += sumVals({d: {e:2, f: 3}})
sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})	1 += sumVals({c: {d: {e:2, f: 3}}})

```

1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);

```

## Example: sumVals

Callstack	sum
sumVals({e:2, f: 3})	2
sumVals({d: {e:2, f: 3}})	0 += sumVals({e:2, f: 3})
sumVals({c: {d: {e:2, f: 3}}})	0 += sumVals({d: {e:2, f: 3}})
sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})	1 += sumVals({c: {d: {e:2, f: 3}}})

```

1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);

```

## Example: sumVals

Callstack	sum
sumVals({e:2, f: 3})	2 += 3
sumVals({d: {e:2, f: 3}})	0 += sumVals({e:2, f: 3})
sumVals({c: {d: {e:2, f: 3}}})	0 += sumVals({d: {e:2, f: 3}})
sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})	1 += sumVals({c: {d: {e:2, f: 3}}})

```

1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);

```

## Example: sumVals

Callstack	sum
sumVals({e:2, f: 3})	5
sumVals({d: {e:2, f: 3}})	0 += sumVals({e:2, f: 3})
sumVals({c: {d: {e:2, f: 3}}})	0 += sumVals({d: {e:2, f: 3}})
sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})	1 += sumVals({c: {d: {e:2, f: 3}}})

```

1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);

```

## Example: sumVals

Callstack	sum
sumVals({e:2, f: 3})	=> 5
sumVals({d: {e:2, f: 3}})	0 += sumVals({e:2, f: 3})
sumVals({c: {d: {e:2, f: 3}}})	0 += sumVals({d: {e:2, f: 3}})
sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})	1 += sumVals({c: {d: {e:2, f: 3}}})

```

1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);

```

# Example: sumVals

sumVals({d: {e:2, f: 3}})

0 += 5

sumVals({c: {d: {e:2, f: 3}}})

0 += sumVals({d: {e:2, f: 3}})

sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})

1 += sumVals({c: {d: {e:2, f: 3}}})

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

# Example: sumVals

sumVals({d: {e:2, f: 3}})

5

sumVals({c: {d: {e:2, f: 3}}})

0 += sumVals({d: {e:2, f: 3}})

sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})

1 += sumVals({c: {d: {e:2, f: 3}}})

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

# Example: sumVals

`sumVals({d: {e:2, f: 3}})``=> 5``sumVals({c: {d: {e:2, f: 3}}})``0 += sumVals({d: {e:2, f: 3}})``sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})``1 += sumVals({c: {d: {e:2, f: 3}}})`

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

# Example: sumVals



# Example: sumVals

sumVals({c: {d: {e:2, f: 3}}})

0 += 5

sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})

1 += sumVals({c: {d: {e:2, f: 3}}})

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

sumVals({c: {d: {e:2, f: 3}}})

5

sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})     1 += sumVals({c: {d: {e:2, f: 3}}})

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

## Example: sumVals

# Example: sumVals

```
sumVals({c: {d: {e:2, f: 3}}})
```

```
=> 5
```

```
sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}}) 1 += sumVals({c: {d: {e:2, f: 3}}})
```

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

# Example: sumVals

```
sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
```

```
1 += 5
```

```
1  function sumVals(obj) {  
2    let sum = 0;  
3    for (let key in obj) {  
4      let value = obj[key];  
5      if (typeof value === 'object') {  
6        sum += sumVals(value)  
7      } else {  
8        sum += value;  
9      }  
10   }  
11   return sum;  
12 }  
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})  
14 console.log(result);
```

# Example: sumVals

sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})

6

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

# Example: sumVals

```
sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
```

```
=> 6
```

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
14 console.log(result);
```

# Example: sumVals

```
1  function sumVals(obj) {  
2    let sum = 0;  
3    for (let key in obj) {  
4      let value = obj[key];  
5      if (typeof value === 'object') {  
6        sum += sumVals(value)  
7      } else {  
8        sum += value;  
9      }  
10   }  
11   return sum;  
12 }  
13 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})  
14 console.log(result);
```



# Example: sumVals

```
1  function sumVals(obj) {
2    let sum = 0;
3    for (let key in obj) {
4      let value = obj[key];
5      if (typeof value === 'object') {
6        sum += sumVals(value)
7      } else {
8        sum += value;
9      }
10   }
11   return sum;
12 }
13
14 let result = sumVals({a: 1, b: {c: {d: {e:2, f: 3}}}})
   console.log(result);
```





# Recap

```
1  /*
2    - Nested data structures and recursion
3    - Nested arrays
4    - Nested objects
5  */
6
7
8
9
10
11
12
13
14
```