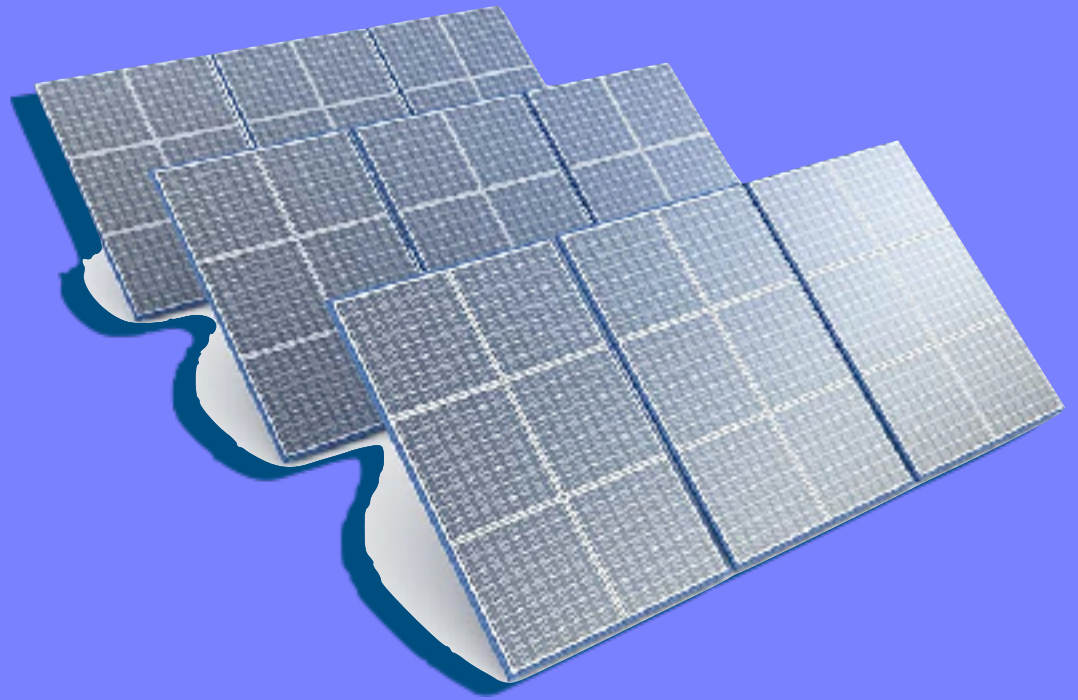
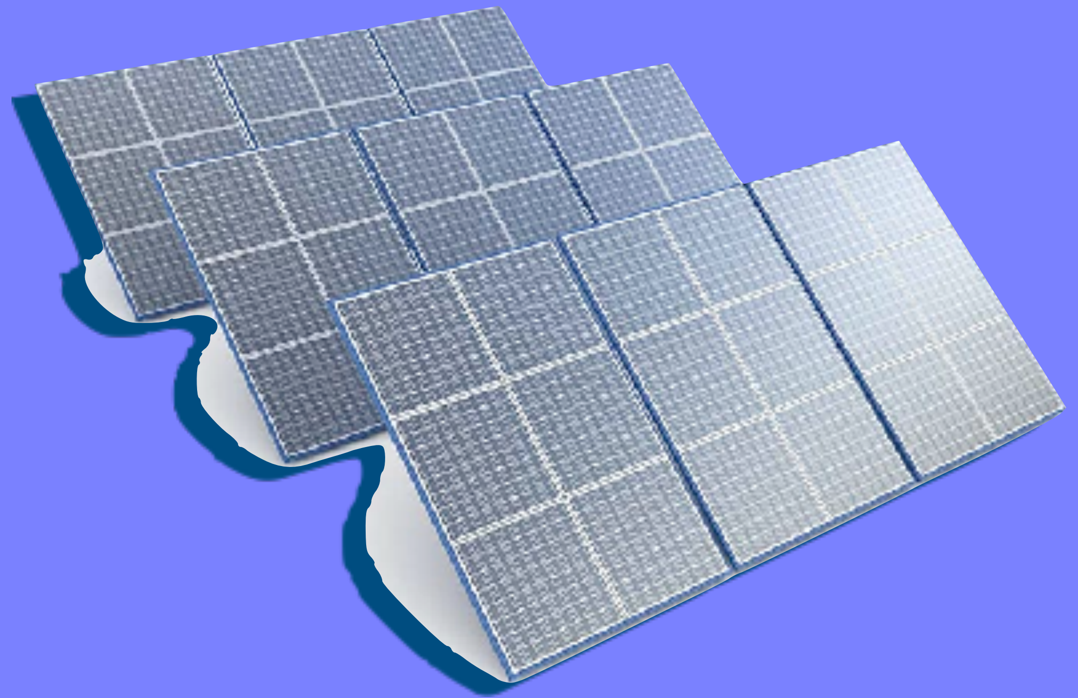
Arrays + Loops



If you heard an **array** of solar panels are being installed, what does that **suggest**?



A simple way to think about an **array** is that it is a **list**

(think of it like a grocery list)

```
//To avoid using variables names such as:
var player1;
var player2;

//We instead would create an Array of players.
var players = new Array();        // bad
var players = [];               // good!
```

Sometimes we need to create **more than one variable** for something.

```
/* The ‘new Array’ keyword is bad because it creates two elements
in the array (this is good): */
var players = new Array(40, 100);

/* However here it creates 40 undefined entries (this is bad or
rather it’s misleading to the programmer): */
var players = new Array(40);

// Easiest way (used by most programmers in JS) is simply
var players = [40, 100];
```

Have you noticed it’s a good idea to **name your array as a plural**?

i.e. cars not car, players not player

(Why is this a better naming convention?)

```
var cars = [‘Saab’, ‘Volvo’, ‘BMW’];
//How do I access the first element?

cars[0];

//Can have different information in an array:
var person = [‘John’, ‘Doe’, 46];

//Objects use names to access its "members"
var person = {
  firstName: ’John’,
  lastName: ’Doe’,
  age: 46
};
```

```
var cars = [‘Saab’, ‘Volvo’, ‘BMW’,];
//We have special properties and methods we can call on arrays:

cars.length;
// The length property returns the number of elements in cars: 3

cars.sort();
// The sort() method sort cars in alphabetical order

car[0];
```

```
var cars = [‘Saab’, ‘Volvo’, ‘BMW’];
delete cars = [0];

/* Unfortunately, this makes the first position
of the array undefined */
var cars = [undefined, ‘Volvo’, ‘BMW’];

// So we need a better way (delete is rarely used)
```

How do
we **delete**
something
from an array?

```
//Add to the end of Array
fruits.push(“orange”);

//Remove from end
fruits.pop();

//Remove from start
 fruits.shift();

//Add to front
fruits.unshift(“Strawberry”) ;
```

Compare the **Spread** method to the **old way**

```
var cars = [‘Saab’, ‘Volvo’, ‘BMW’];
cars.splice(0, 1);
/* This will remove one element from the array
at position 0, completely */

//We can also use this to add in elements to the array
cars.splice(0, 1, ‘Cadillac’,‘Tesla’);

/* This will add two new elements Cadillac and Tesla
at the first position (Saab = 0, so after that)
and remove no elements */
```

How do
we **delete**
something
from an array?

```
var cars = [‘Saab’, ‘Volvo’, ‘BMW’];

//To add a new car into the array use:
cars[cars.length] = ‘VW’;

//Or an easier way is:
cars.push(‘VW’);

//If you used:
cars[10] = ‘Mercedes’;
//it would make:

var cars = [‘Saab’, ‘Volvo’, ‘BMW’,”,”,”,”,”,”,”,‘Mercedes’];
var rivers = [‘Rio Grande’, ‘Ganjes’, ‘Yangtzee’];
var moreRivers = [‘Danube’, ‘Amazon’, ‘Nile’];

rivers.push(...moreRivers);
console.log(rivers);
//the ‘...’ joins (spreads) the new array values together.
```

NEW for JS ES6, there is an easy way to join two arrays together; it’s called **Spread**.

```
//Old way, in ES5 using Splice
Array.prototype.push.apply(rivers, moreRivers);

//New way in ES6
rivers.push(...moreRivers);
```

Compare the **Spread** method to the **old way**

```
var cars = [‘Saab’, ‘Volvo’, ‘BMW’];
cars.reverse;
//Reverses the order of the array
```

How do
we **reverse**
the order
of the array?

In order to **access** all of the elements in an **array**, it is necessary to **loop** through them

```
for (statement 1; statement 2; statement 3) {
    //code block to be executed
}

//example
for ( let i = 0; i < cars.length; i++) {
  console.log(‘Make of car’+ cars[i]);
}
```

The most
common way
to do this
is using a
for loop.

```
let scores = [70, 90, 80];

for (let score of scores) {
  score = score + 5;
  console.log(score);
}
/* The score variable is the individual
value that iteration of the loop */
```

New!
ES6 way
to loop

We can also use a **for of**.

Let’s look at some examples

Other ways to loop

Practice Quiz!

```
var players = new Array();
var players = [];
```

Which is good and which is bad? Why?

```
var cars = [‘Saab’, ‘Volvo’, ‘BMW’];
```

How to we access the 3rd element in this array?

```
var cars = [‘Saab’, ‘Volvo’, ‘BMW’];
```

How do we find out the length of this array?

```
var cars = [‘Saab’, ‘Volvo’, ‘BMW’];
```

How would we completely remove the first element in this array?

```
var cars = [‘Saab’, ‘Volvo’, ‘BMW’];
```

How would we add Cadillac and Tesla after Volvo in this array?

```
var rivers = [‘Rio Grande’, ‘Ganjes’, ‘Yangtzee’];
var moreRivers = [‘Danube’, ‘Amazon’, ‘Nile’];
```

How would we join these two arrays together in JavaScript ES6?

What is the 3rd statement in the for loop?

What does the second statement of the for loop do?

```
let scores = [70, 90, 80];
```

Create a new JavaScript ES6 ‘for of’ loop for the above array.