.Reduce( )

Just like map( ), reduce( ) also runs a callback for each element of an array. What's different here is that reduce passes the result of this callback (the accumulator) from one array element to the other.

Say you have an array with these pilots and thier ~~represen~~ respectives years of experience:

```
let pilots = [
  {
     id: 10,
     name: "Poe Dameron",
     years: 14
  }
  {
     id: 24,
     name: "Sam Lost",
     years: 2
  }
```

We need to know the total years of experience of all of them.

```
let totalYears = pilots.reduce((acc, pilot) => acc + pilot.years,
0);
```

or

```
{
   return accumulator + pilot.years
}, 0);
```

Notice that you've ~~used an existing~~ set the starting value as 0. You could have also used an existing variable if necessary.

Now lets say you want to find which pilot is the most experienced one

```
let mostExpPilot = pilots.reduce((oldest, pilot) => (oldest.years||0)
   > pilot.year ? oldest : pilot;}, {})
```

If a pilot has more years of experience than oldest, then pilot becomes the new oldest.