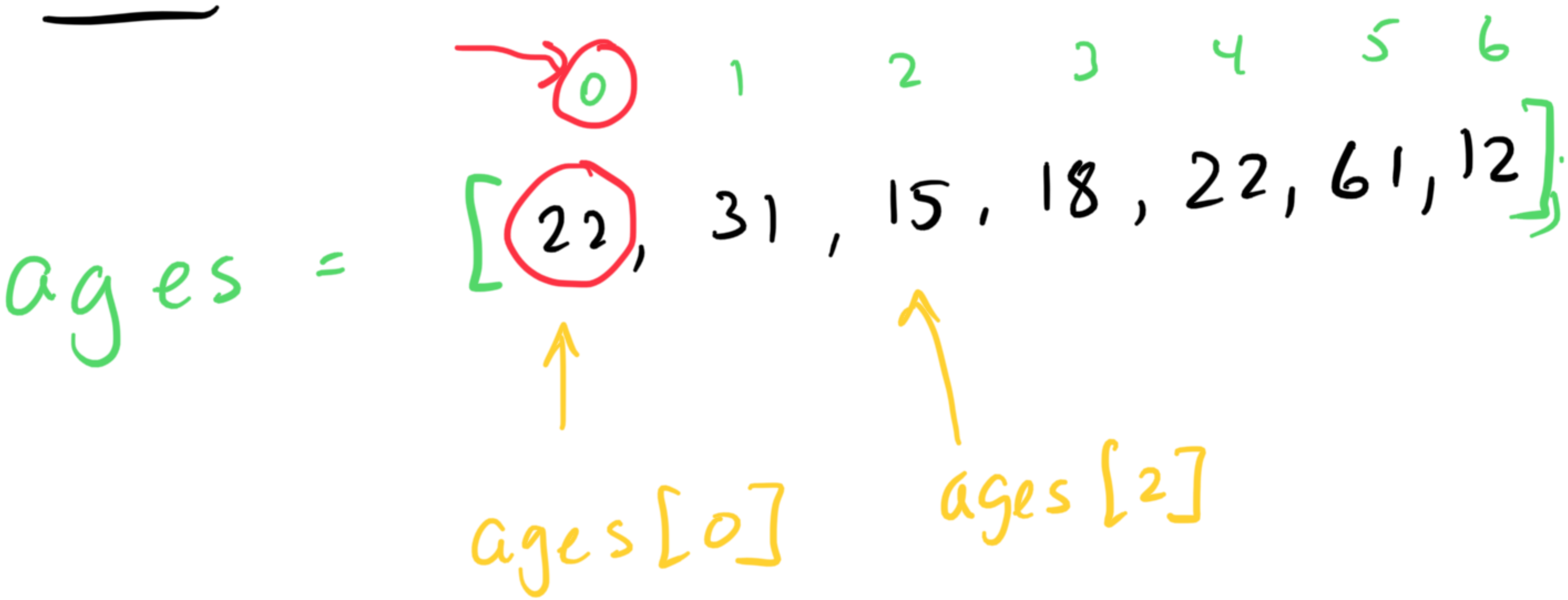
JavaScript Arrays

You have, in CPSC 140/150 seen lists, and possibly numpy arrays. Java script arrays are very similar.

They are an ordered complex data type. Because they are ordered each element can be referred to by its index (or position) in the array.



Javascript arrays → all elements are of the same type, i.e. ints, strings, floats, or other arrays.

matrix = [[1,2;3],[4,5,6],
[7,8,9]];

$$\Downarrow$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Arrays, as a special JavaScript complex data type, come with a lot of built-in capabilities!

ages.push(64); → add 64 as a new element, at the end.

ages.pop(); → remove last element from the end.

ages.unshift(27); → add 27 at the beginning.

Iterators

We can, of course, use for loops to progress through each element of an array, e.g.

```
for (let i=0; i< ages.length; i++) {
            console.log(ages[i]);

}
```

This works, but it is S... L ... O ... W!

For large arrays, it can be painfully slow.

There are these things called iterators that allow for very fast processing of arrays in JavaScript. Think list comprehension in Python, if you know what that is

(ages :)

```
(e.g.   for age in ages
               print(age) )
```

Two methods :

①

```
for (item of ages) {
        console.log (item);

}
```

one thing.

②

```
ages.forEach( function(item) {
                    console.log(item):
              } )
```

complex.