Swapping Variables

In Arrays / Lists

Problem: how to exchange two values in an Array / List

Python:

```
my_list = [1,2,3,5,8]
```

C++:

```
int my_list[] = {1,2,3,5,8};
```

Exchange the values of:
my_list[1] (integer 2)
and
my_list[2] (integer 3)

Problem: how to exchange two values in an Array / List

Python:

```
my_list = [1,2,3,5,8]
```

C++:

```
int my_list[] = {1,2,3,5,8};
```

CAUTION!
If you copy with assignment, the original value gets lost!
my_list[1] = my_list[2]
…what happens to 2?

Problem: how to exchange two values in an Array / List

Python:

```
my_list = [1,2,3,5,8]
```

C++:

```
int my_list[] = {1,2,3,5,8};
```

The order you do it in isn't important, the same thing will happen.
my_list[2] = my_list[1]
…what happens to 3?

RULE 1: Think about the problem.

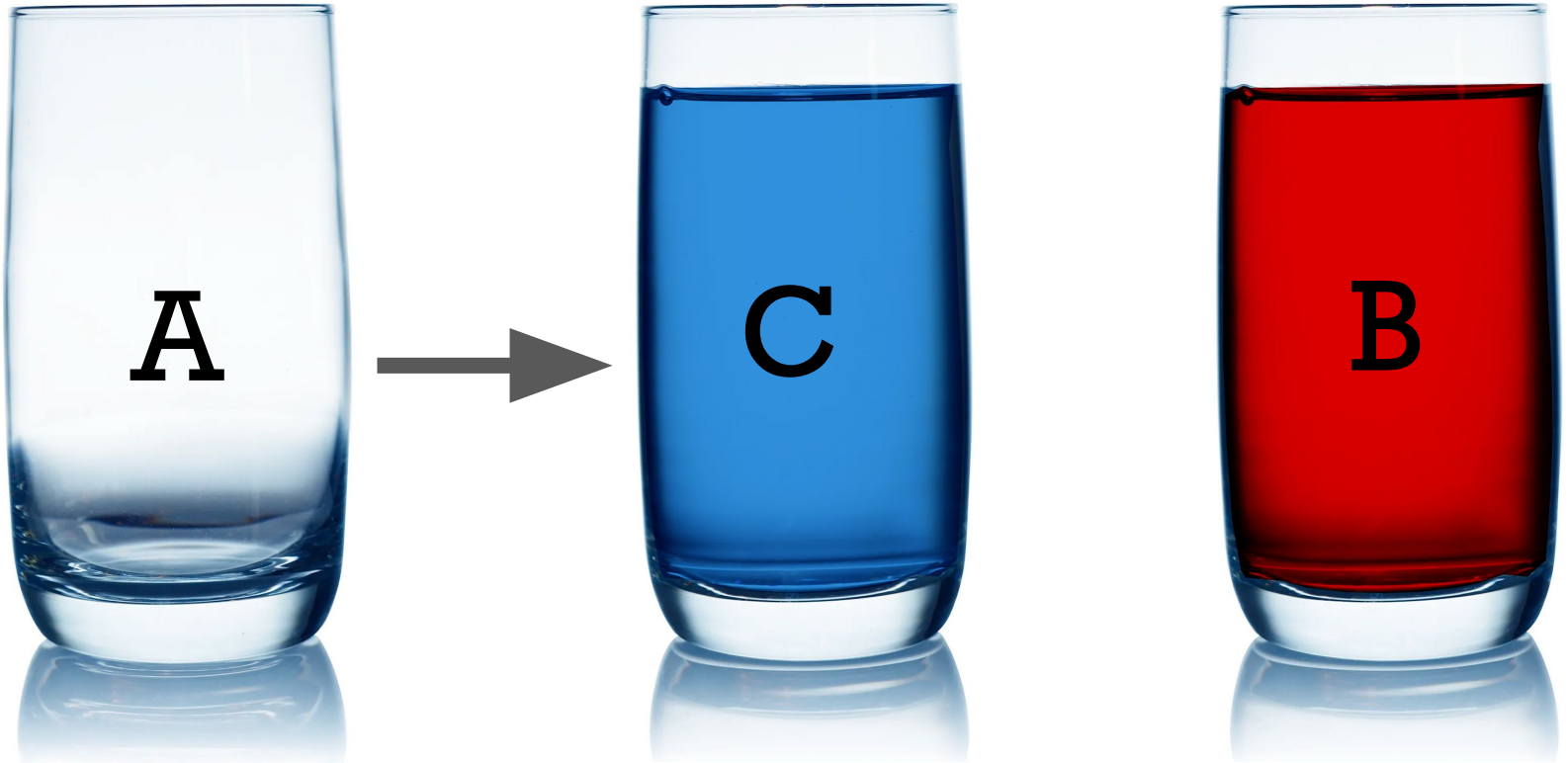
RULE 1: Think about the problem.

Pour a little bit from one
to the other repeatedly?

❌ NO! ❌

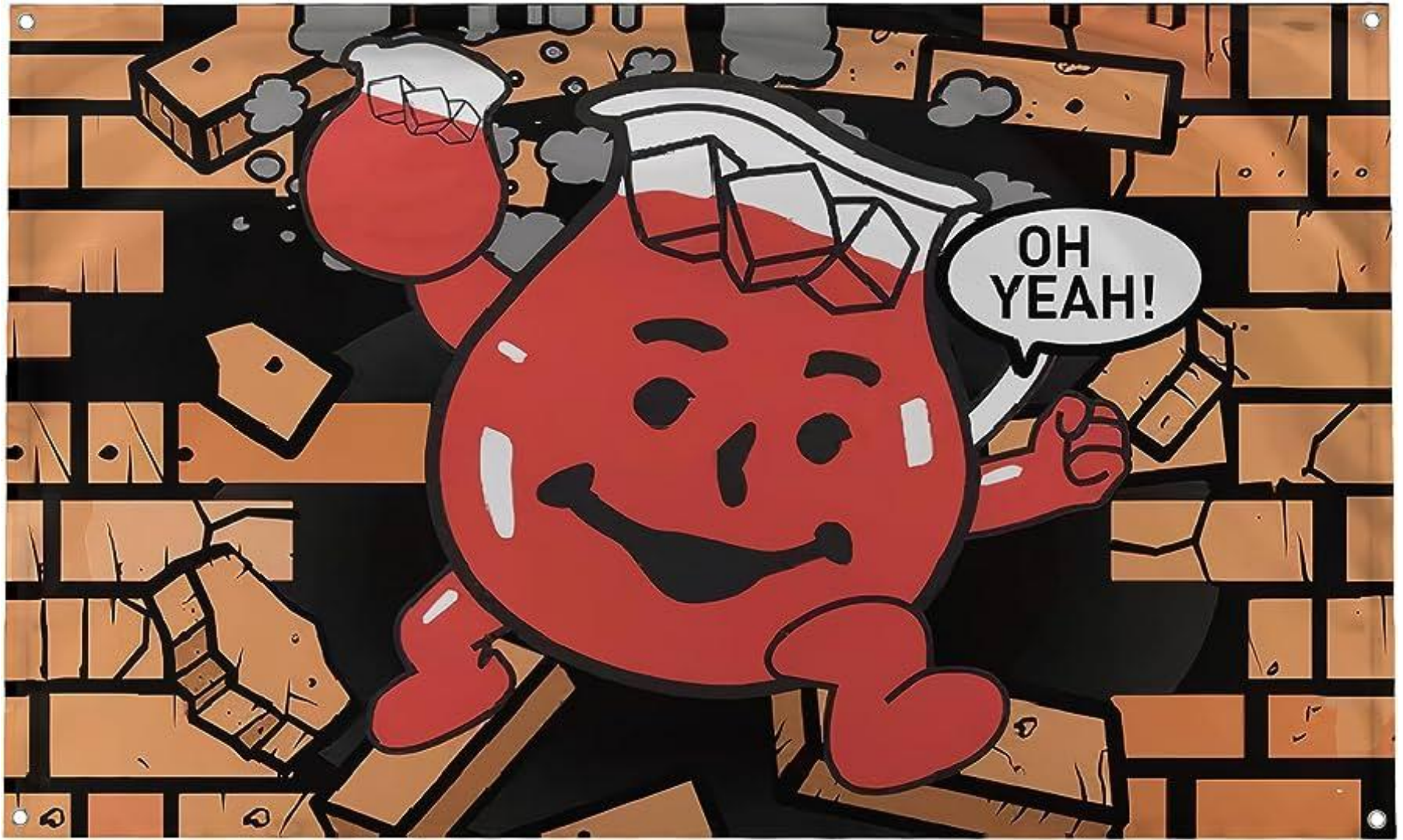
Solution: Introduce a “swap” variable.

Using the “swap” variable:



Using the “swap” variable:

Values preserved!


OH
YEAH!

How to exchange two values in an Array / List

Python:

```
my_list = [1,2,3,5,8]

swap = my_list[1]

my_list[1] = my_list[2]

my_list[2] = swap
```

C++:

```
int my_list[] = {1,2,3,5,8};

int swap;

swap = my_list[1];

my_list[1] = my_list[2];

my_list[2] = swap;
```

This is a GENERIC strategy!

- You can do the same thing with *any* variable type.
- You can swap between *any* array location, as long as you keep track of both the source and the destination indices and do every swap operation in the same order.
- More importantly, it allows us the ability to naively sort *any* array, as long as there are syntax rules to compare the two values.
- We have Boolean operators (>, <, >=, <=) for this.