Structs:

↳ Our own datatype

What if we want to make a data type which holds a flower?

↳ What things may we want to know about the flower?

- height
- Color
- kingdom
- Clade ... etc

```
struct flower {
    int height; // holds inches
    char color; // holds colorchar
};
```

we can declare a flower by:

```
flower myflower;
```

datatype name

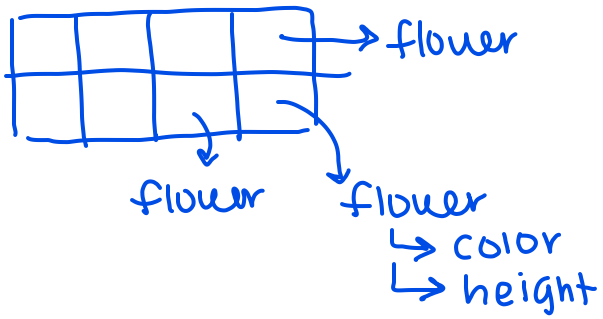
and modify it's values:

```
myflower.height = 2;
```

name attribute

```
myflower.color = 'c'; //cyan
```

What if we want to store these flowers in a garden box?




```
flower** garden = new flower*[3];
```

datatype (double ptr) | name | heap allocation | data type (ptr) | size

what does this line do?

(flower**) garden → | flower* | flower* | flower* |

we've now made an array of flower* how do we turn this into a 2D array?

```
for (int i = 0; i < 3 , i++) {
    garden[i] = new flower[2];
}
```

name | loop through | heap alloc. | datatype | size

what will this loop look like when it's finished?

(flower**) garden → | flower* | flower* | flower* |

flower	flower	flower
flower	flower	flower

each one of these flowers can have it's height & color attributes changed.