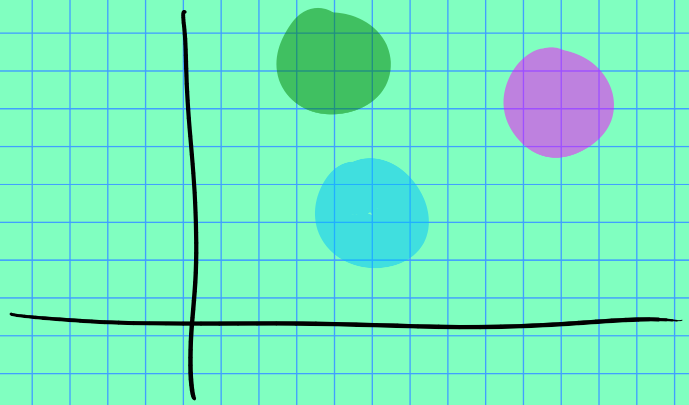
How to make your own datatypes?

"struct" in C/C++.

Idea! glue together a few pieces of data, and give each a nice name.

Example: how to represent circles in the plane, each w/ its own color:

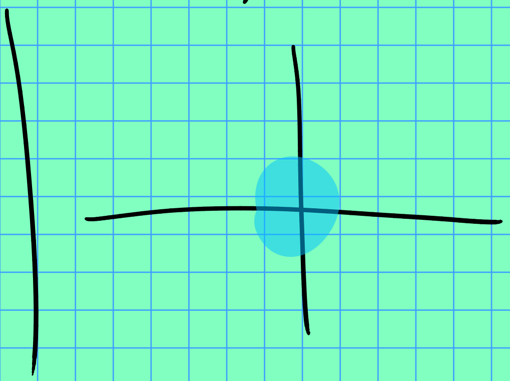


```
struct circle {
    double x;   // x coord. of center
    double y;   // y coord of center
    int color;  // identifies color from
                // enumerated palette.
};
```

How to use?

```
circle c;  // c is of type circle.

c.x = 0;
c.y = 0;
c.color = 2;
```



Can also declare arrays the same way:

circle C[100];

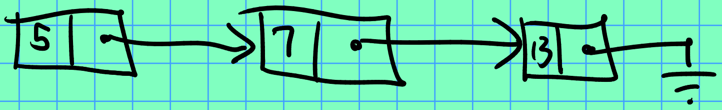
Alternative, w/o using struct?

x	y	color
0	0	2
1	3	4
4	7	1

```
vector<double> x;  // hold all the x coords
vector<double> y;  // hold all y coords
vector<int> color; // hold all colors.
```

Then
C[i].x ≡ x[i]
C[i].y ≡ y[i]
C[i].color ≡ color[i]

Better example: linked lists.

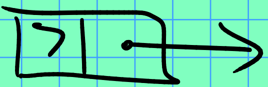


Similar to an array/vector, but:

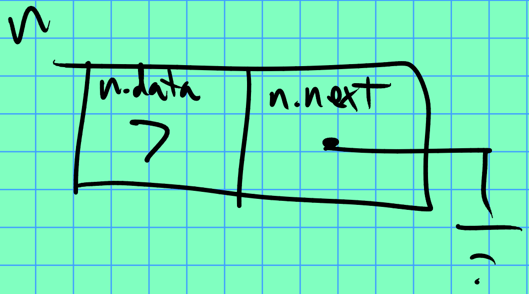
- Bad at random access (no [i])
- Good at insertion to any location (for vectors, only insertion at end (push_back) is efficient.)

How to do it in C/C++?

```
struct node {
    int data;
    node* next;
};
```

// 

program	picture

node n;
n.data = 7;
n.next = NULL;
	

Exercise: read integers from stdin, and store in a list. Note! we will only explicitly keep track of the first node, similar to how a dynamically allocated array holds a pointer to the first element.