

```
class _TaskCardState extends State<TaskCard> {
  bool test = false;
  Task item;
}
```

Variable

```
Color checkDueDate(dueDate) {
    // determines if the card should be red or blue depending on its completion status
    if (dueDate > DateTime.now().millisecondsSinceEpoch / 1000) {
        return Colors.blueAccent;
    } else if (item.status) {
        return Colors.blueAccent;
    } else {
        return Colors.red;
    }
}
```

- Adds padding around the card to make it look cleaner

```

_TaskCardState(this.item);
@override
Widget build(BuildContext context) {
  return Padding(
    padding: const EdgeInsets.all(8),
    child: InkWell(

```

Gets the colour of the card depending on its completion status. If it's complete, it should return blue. If it's past its due date and also uncomplete. It will return red

This container  
constrains the  
width of the  
tasks  
heading/context  
so it doesn't  
overflow the  
card

Aligns children of column at start of row

## The tasks heading

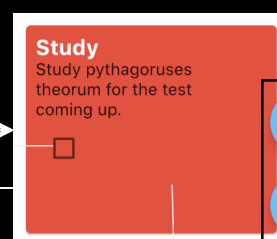
## Style of heading

tasks  
context

Checkbox that shows the completion status of the task.

setState rebuilds the  
card with the changed  
status

changes the status  
in the database



this is what this code would return if:

```
item.heading = "Study"  
item.context = "Study Pythagorases is theorem  
for the test coming up"  
item.status = false  
item.due = < current DateTime
```

