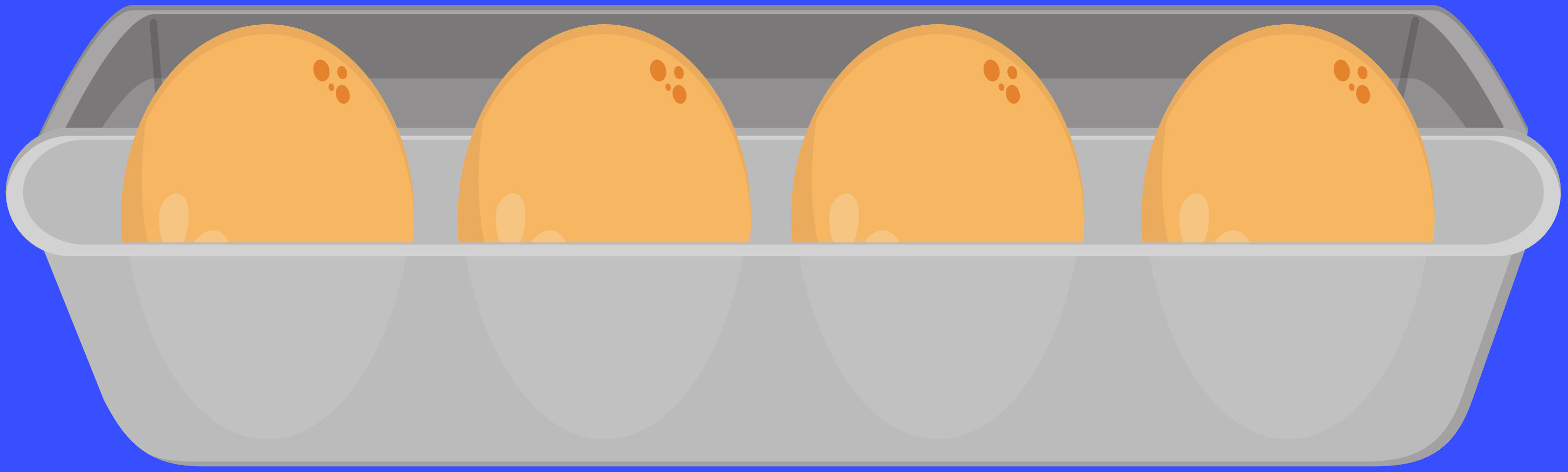
ARRAYS & OBJECTS

Collections of Data



GOALS

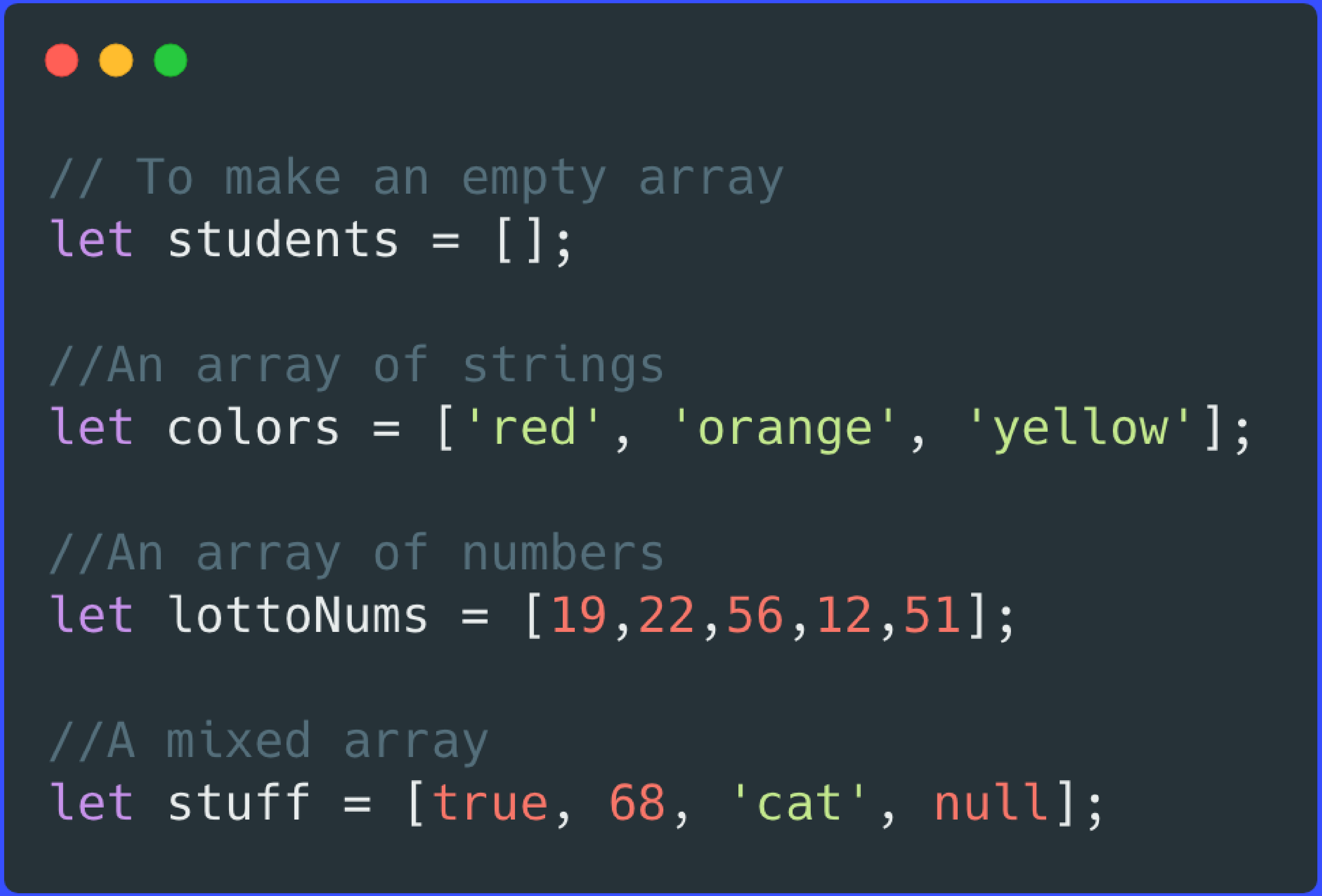
- Work with Arrays
- Write Object Literals
- Understand Reference Types
- Use common Array/Object methods

ARRAYS

Ordered collections of values.

- List of comments on IG post
- Collection of levels in a game
- Songs in a playlist

CREATING ARRAYS



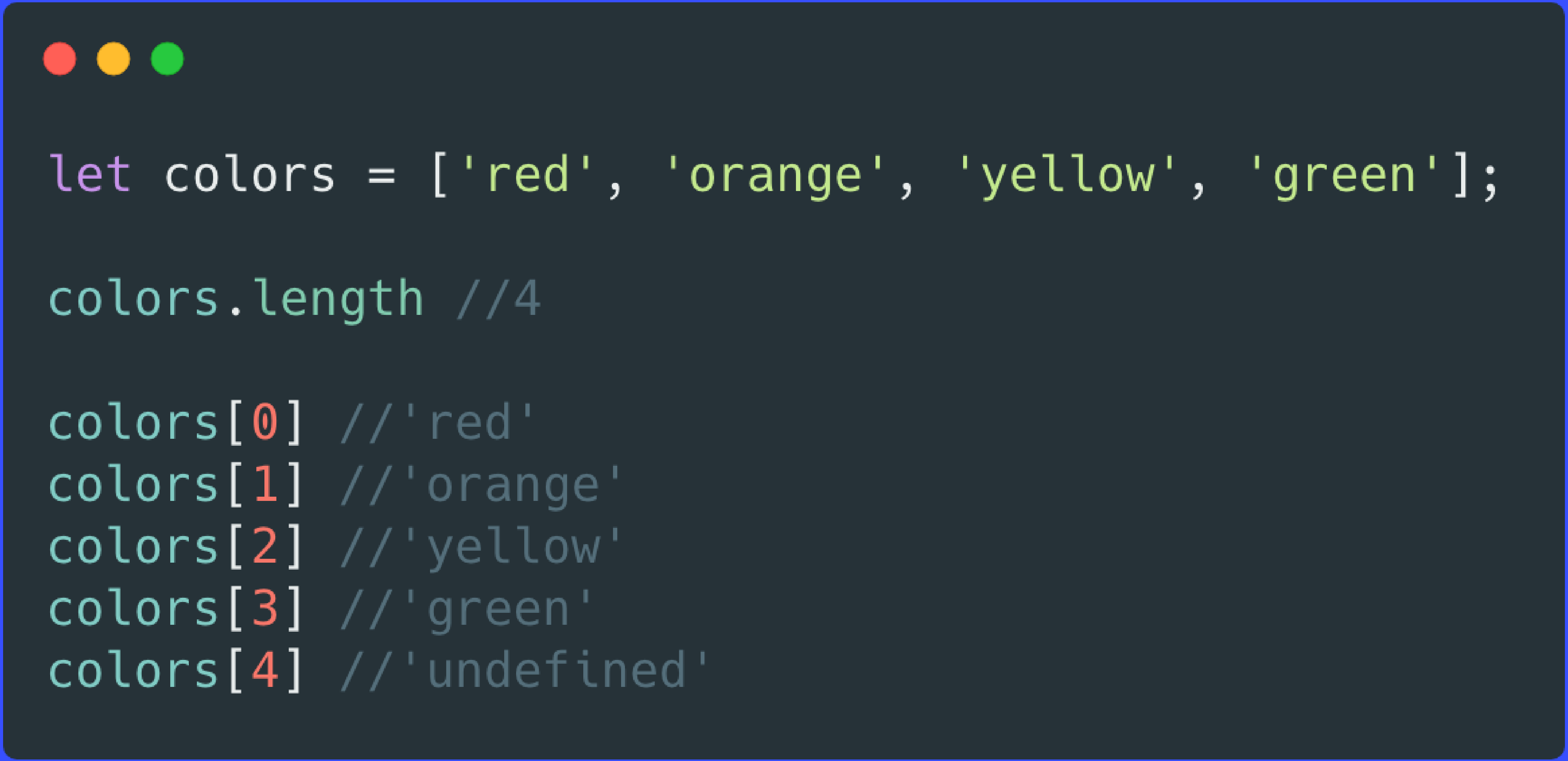
```
// To make an empty array
let students = [];

//An array of strings
let colors = ['red', 'orange', 'yellow'];

//An array of numbers
let lottoNums = [19,22,56,12,51];

//A mixed array
let stuff = [true, 68, 'cat', null];
```

ARRAYS ARE INDEXED

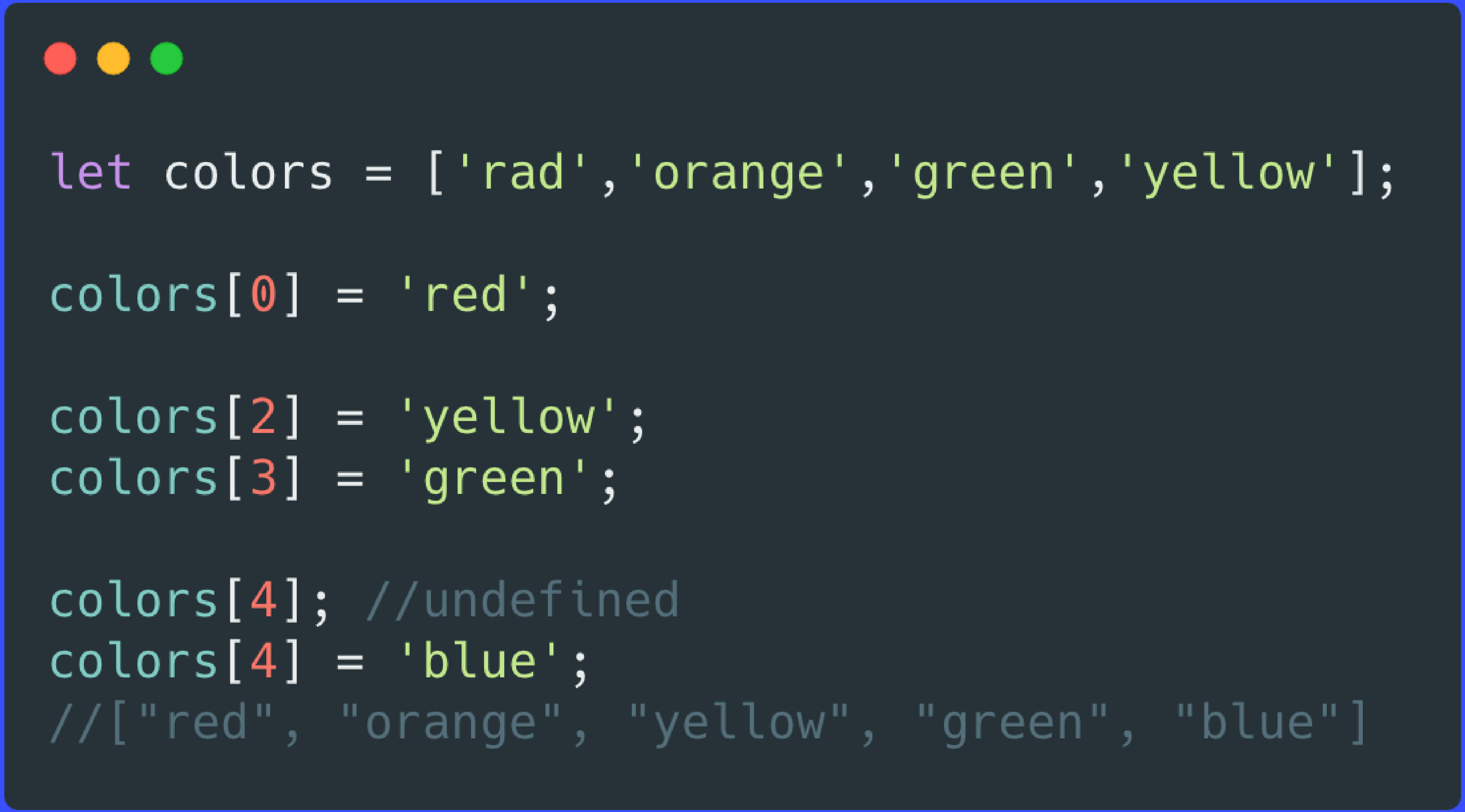


```
let colors = ['red', 'orange', 'yellow', 'green'];

colors.length //4

colors[0] //'red'
colors[1] //'orange'
colors[2] //'yellow'
colors[3] //'green'
colors[4] //'undefined'
```

MODIFYING ARRAYS



```
let colors = ['rad','orange','green','yellow'];

colors[0] = 'red';

colors[2] = 'yellow';
colors[3] = 'green';

colors[4]; //undefined
colors[4] = 'blue';
//["red", "orange", "yellow", "green", "blue"]
```

ARRAY METHODS

- Push - add to end
- Pop - remove from end
- Shift - remove from start
- Unshift - add to start

* You might mix up shift and unshift for a while. I still do...

MORE METHODS!

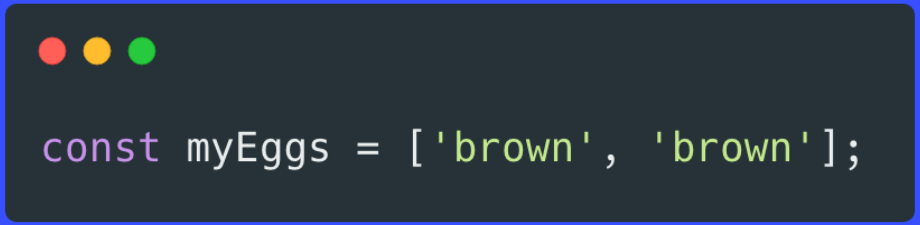
- concat - merge arrays
- includes - look for a value
- indexOf - just like str.indexOf
- join - creates a string from arr
- reverse - reverses an array!
- slice - copy portion of an arr
- splice - remove/replace elements
- sort - sorts an array

CONST & ARRAYS

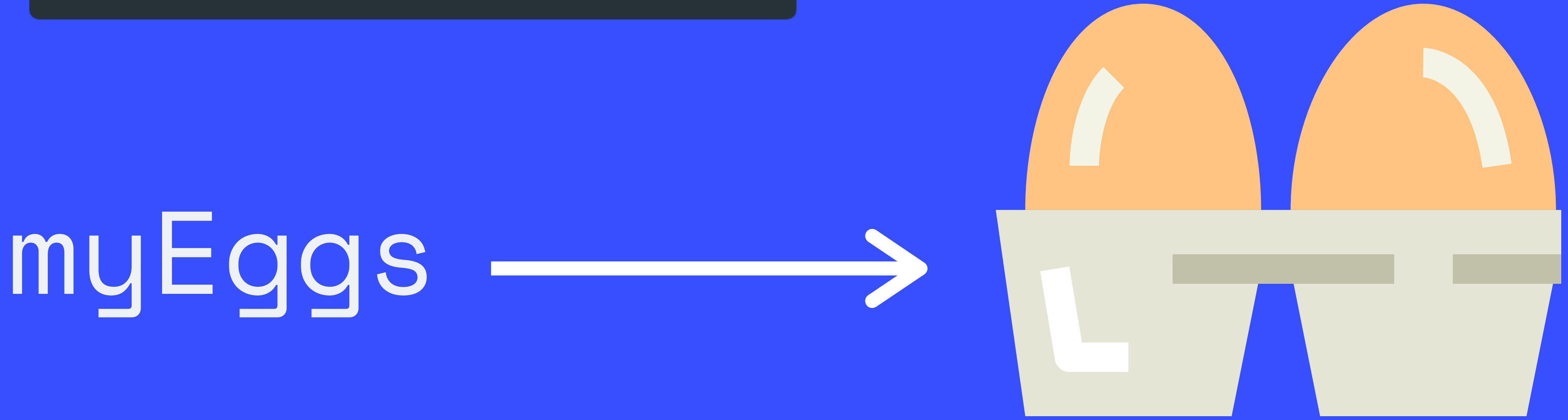
You'll often see people use *const* with arrays.

VALUES CAN CHANGE

As long as the reference remains the same



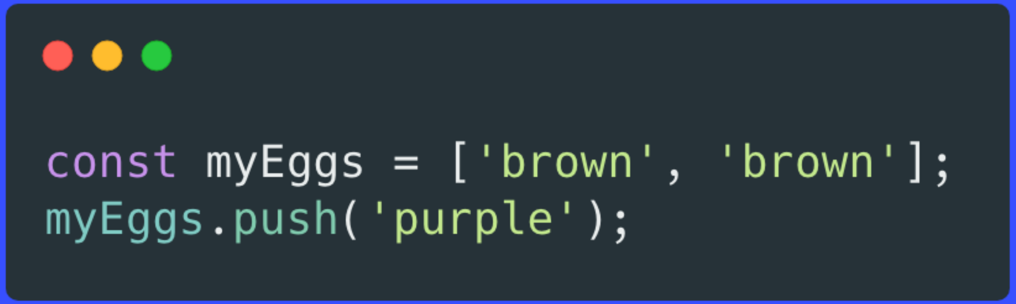
```
const myEggs = ['brown', 'brown'];
```



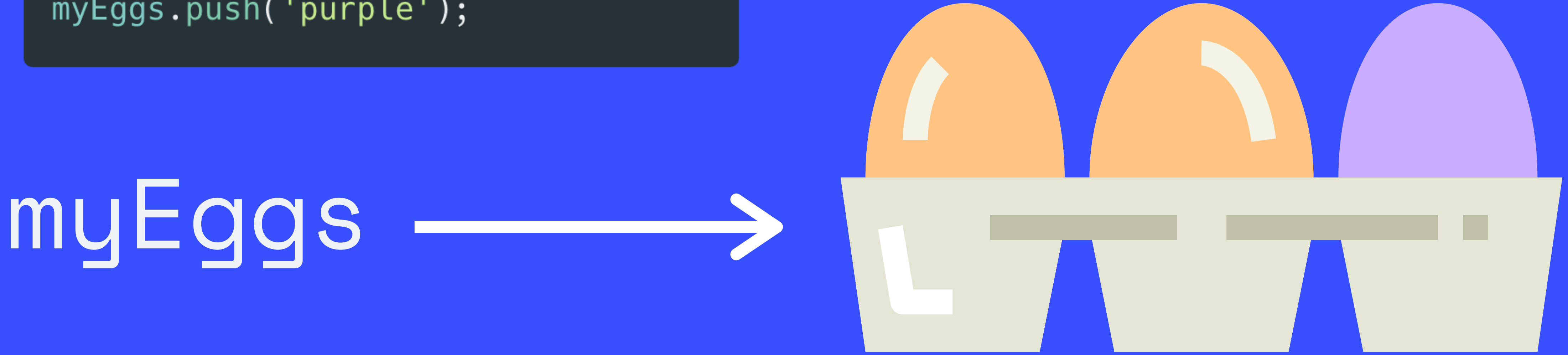
myEggs

VALUES CAN CHANGE

As long as the reference remains the same



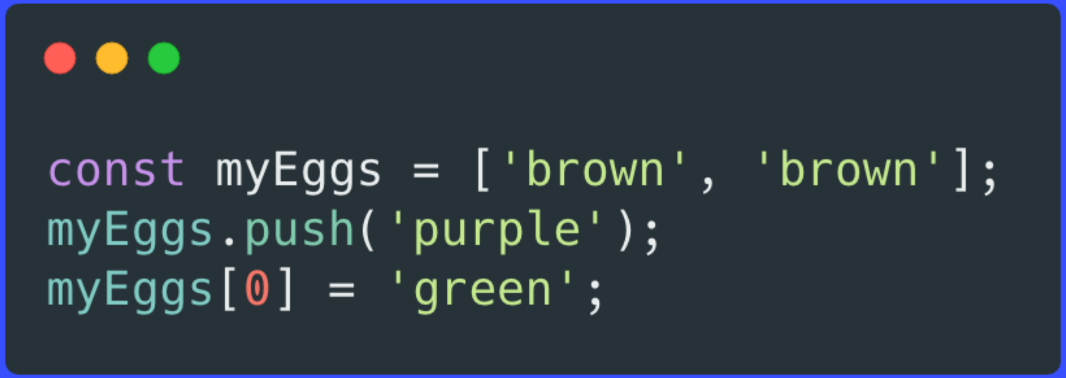
```
const myEggs = ['brown', 'brown'];
myEggs.push('purple');
```



myEggs

VALUES CAN CHANGE

As long as the reference remains the same

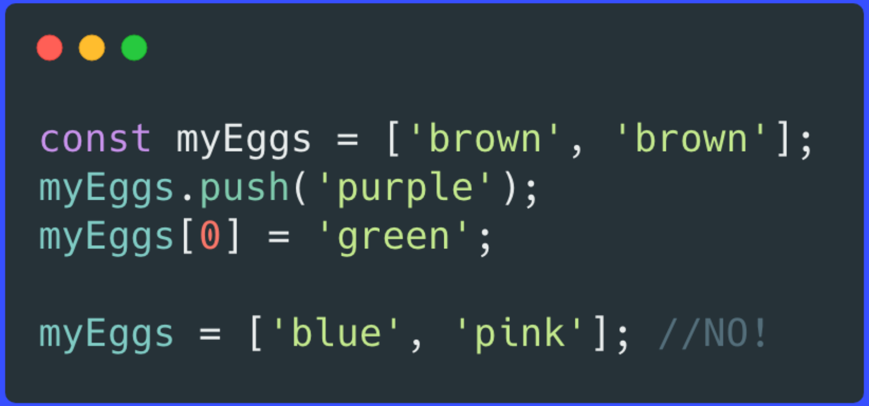


```
const myEggs = ['brown', 'brown'];
myEggs.push('purple');
myEggs[0] = 'green';
```

myEggs →

VALUES CAN CHANGE

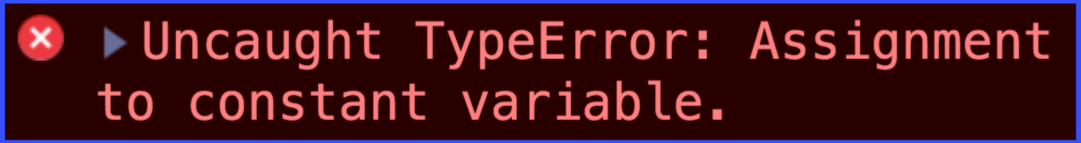
As long as the reference remains the same



```
const myEggs = ['brown', 'brown'];
myEggs.push('purple');
myEggs[0] = 'green';

myEggs = ['blue', 'pink']; //NO!
```

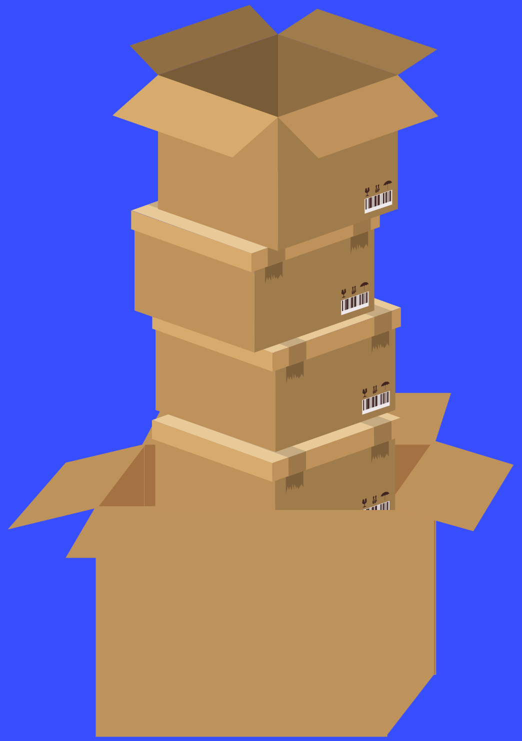
myEggs

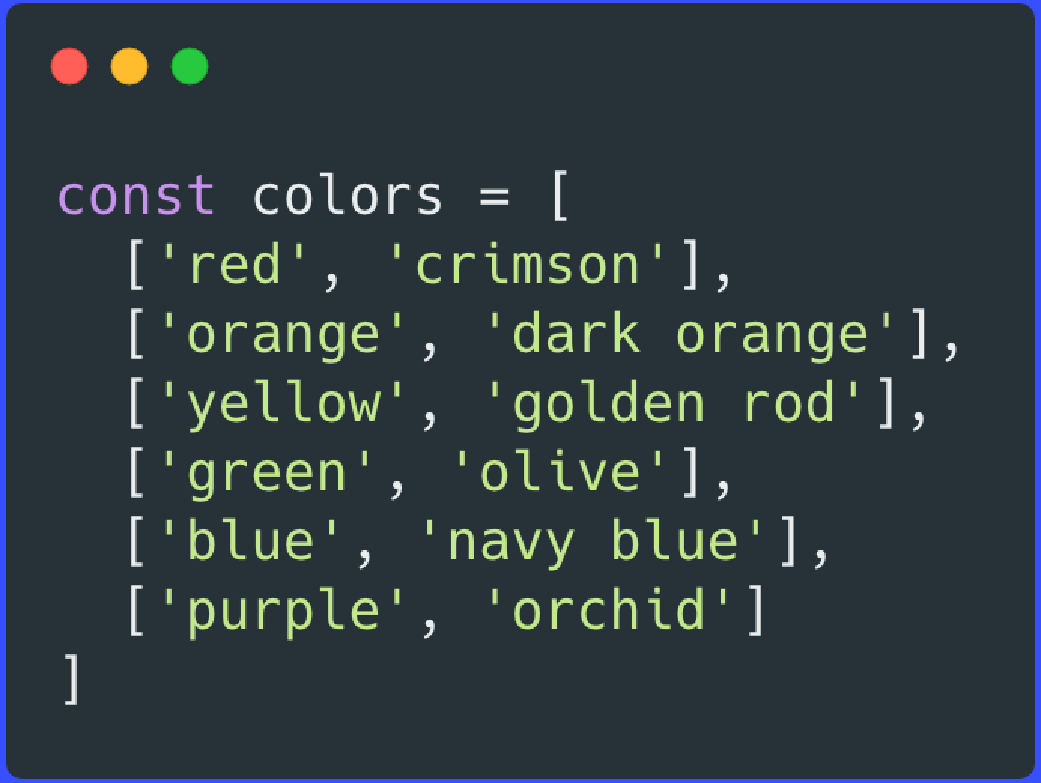


```
Uncaught TypeError: Assignment
to constant variable.
```

NESTED ARRAYS

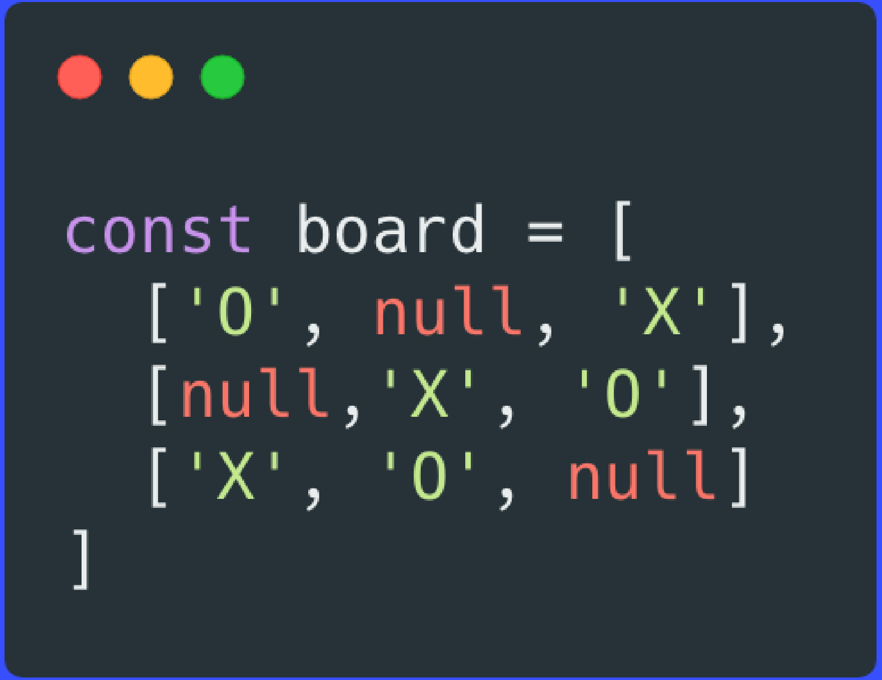
We can store arrays inside other arrays!



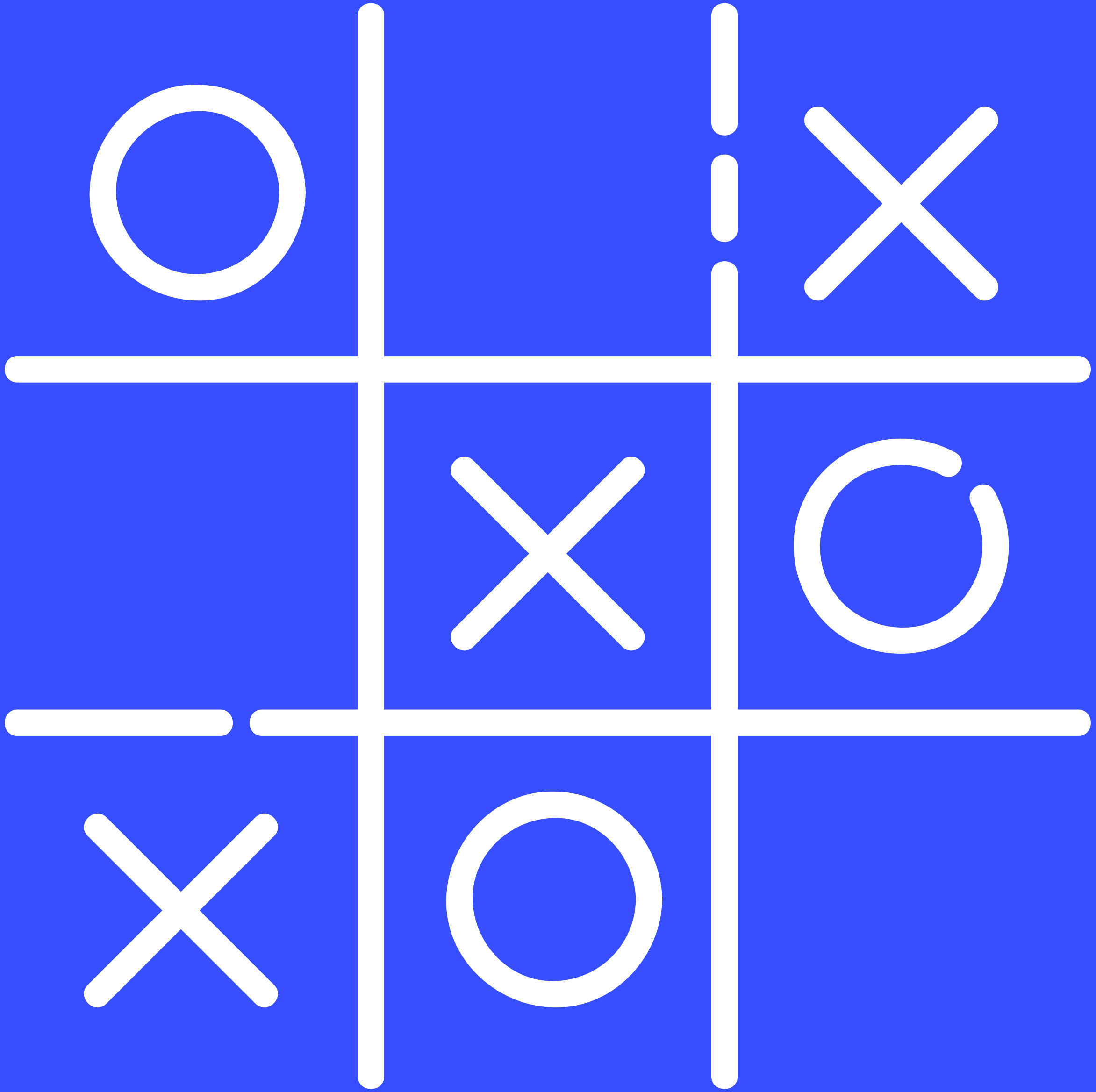


```
const colors = [
  ['red', 'crimson'],
  ['orange', 'dark orange'],
  ['yellow', 'golden rod'],
  ['green', 'olive'],
  ['blue', 'navy blue'],
  ['purple', 'orchid']
]
```

NESTED ARRAYS

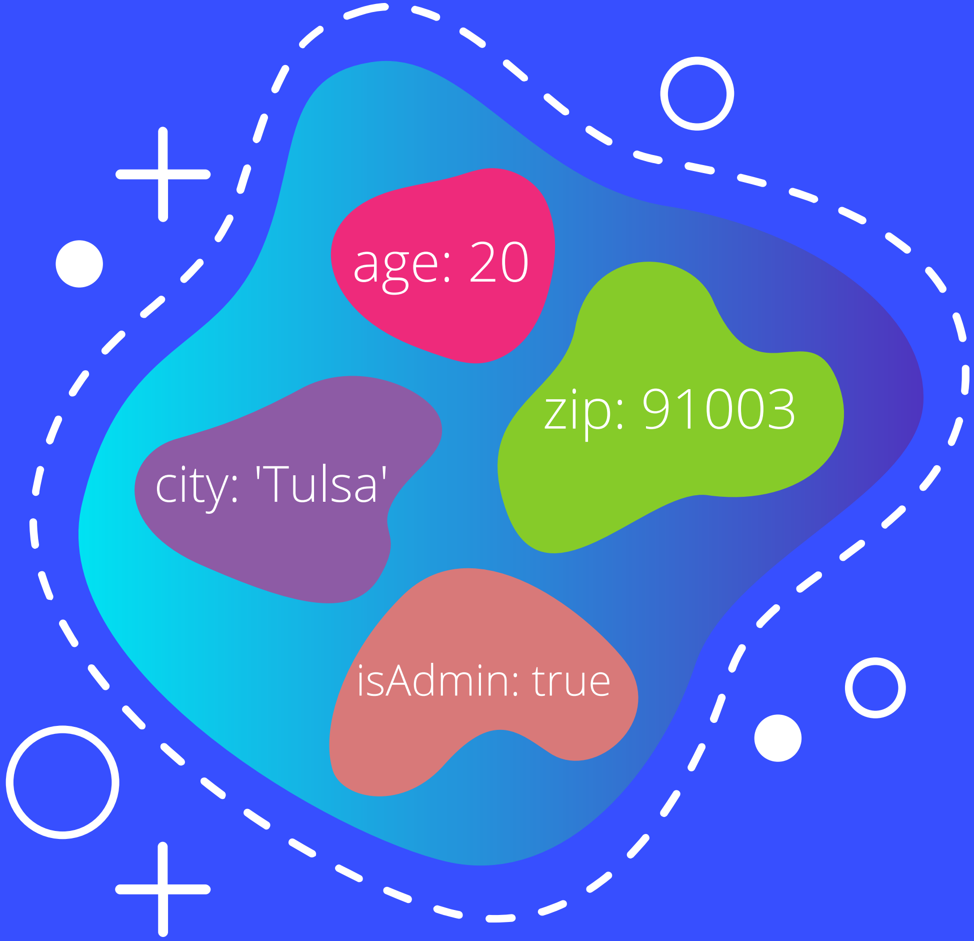


```
const board = [
  ['O', null, 'X'],
  [null,'X', 'O'],
  ['X', 'O', null]
]
```

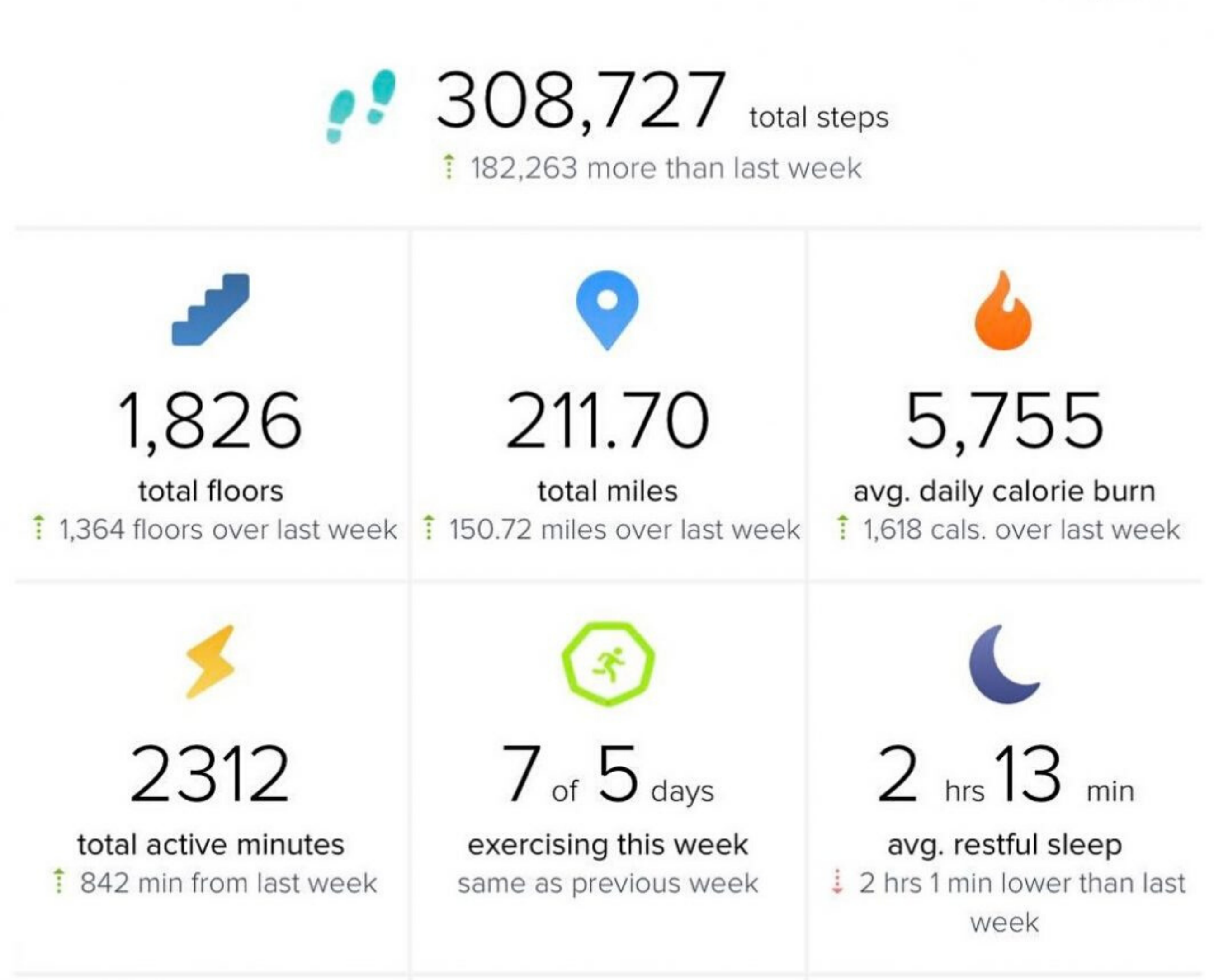


OBJECTS

- Objects are collections of properties.
- Properties are a key-value pair
- Rather than accessing data using an index, we use custom keys.



HOW WOULD YOU STORE THIS?



308,727 total steps
↑ 182,263 more than last week

1,826
total floors
↑ 1,364 floors over last week

211.70
total miles
↑ 150.72 miles over last week

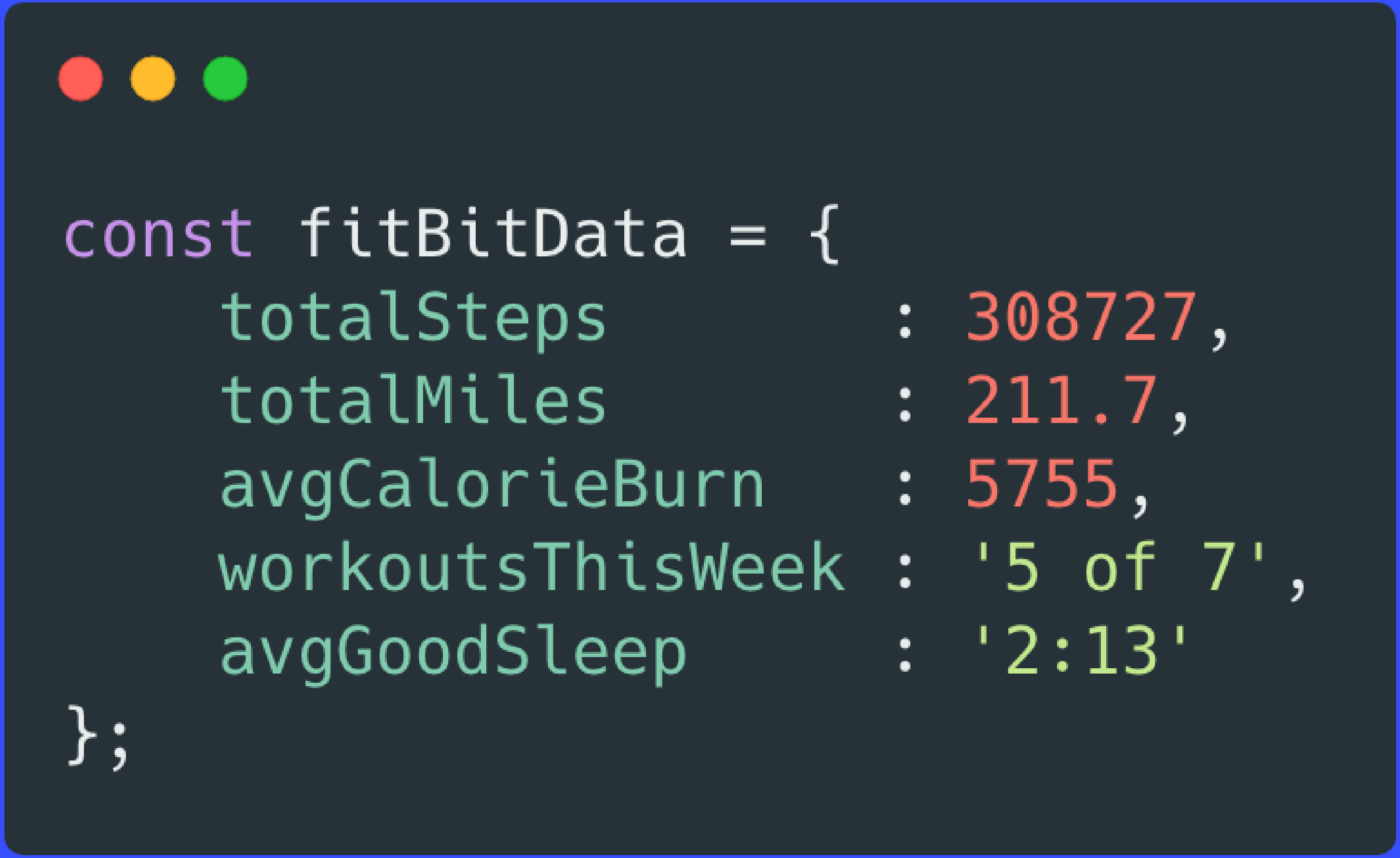
5,755
avg. daily calorie burn
↑ 1,618 cals. over last week

2312
total active minutes
↑ 842 min from last week

7 of 5 days
exercising this week
same as previous week

2 hrs 13 min
avg. restful sleep
↓ 2 hrs 1 min lower than last week

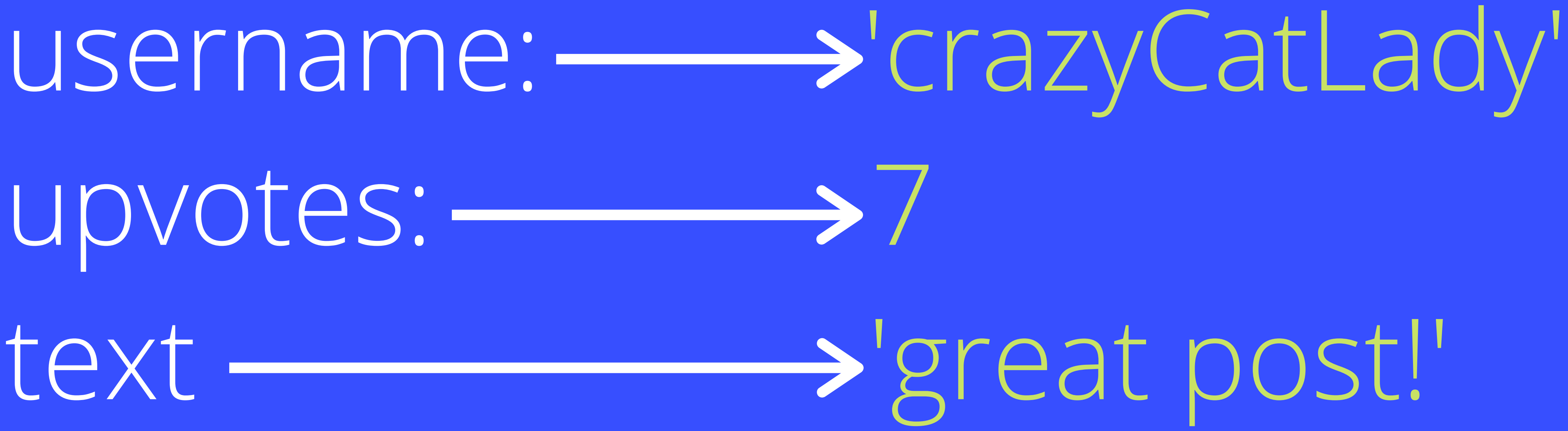
AN OBJECT!



```
const fitBitData = {
    totalSteps       : 308727,
    totalMiles       : 211.7,
    avgCalorieBurn   : 5755,
    workoutsThisWeek : '5 of 7',
    avgGoodSleep     : '2:13'
};
```

PROPERTY = KEY + VALUE

KEY-VALUE PAIRS



username: → 'crazyCatLady'

upvotes: → 7

text → 'great post!'

DICTIONARY

clarion:
brilliantly clear

donnybrook:
uproad & disorder

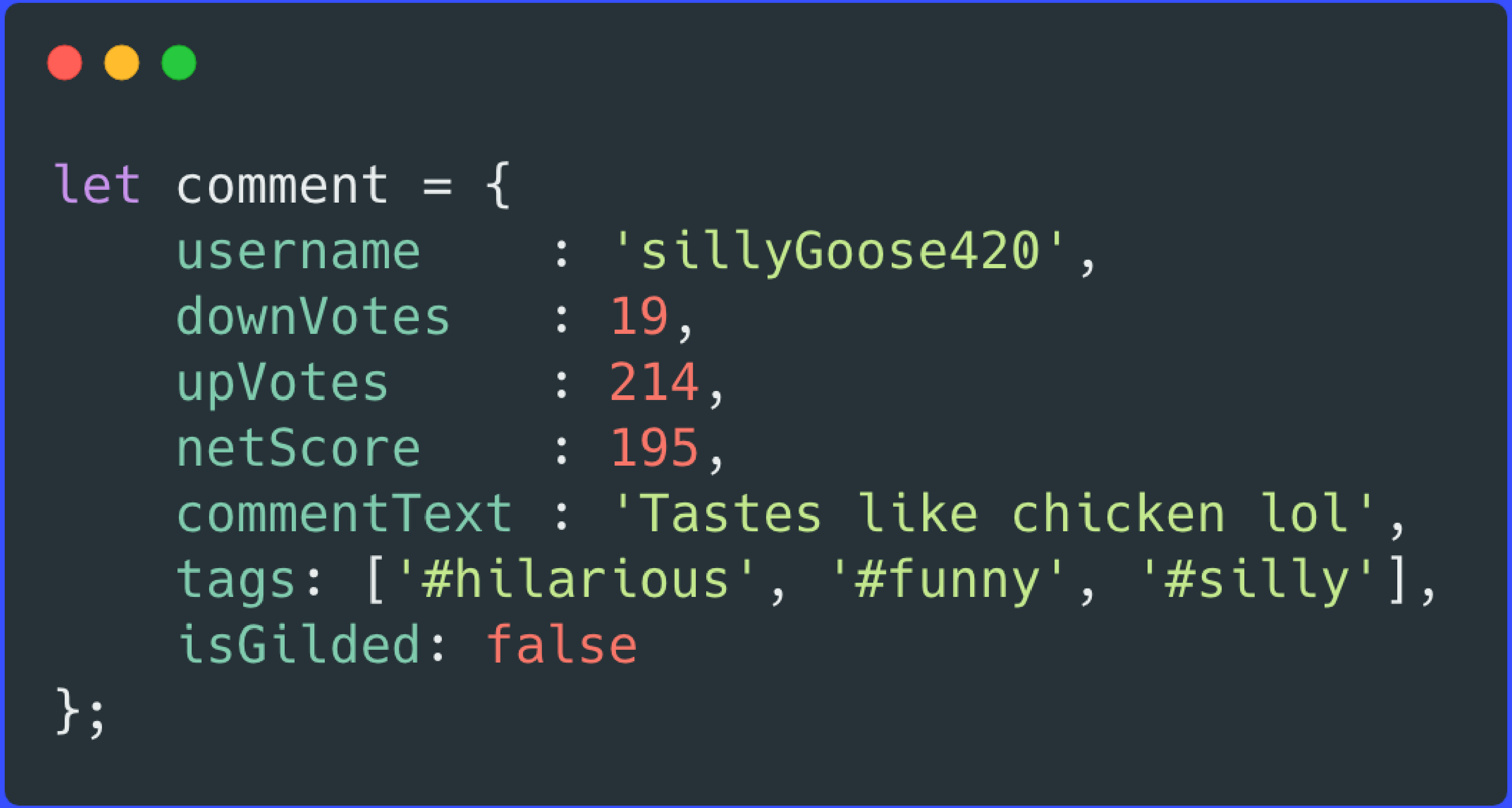
fantod:
emotional outburst

fecund:
fruitful in offspring

pursy:
having a puckered appearance

remonstrate:
argue in protest

ALL TYPES WELCOME!



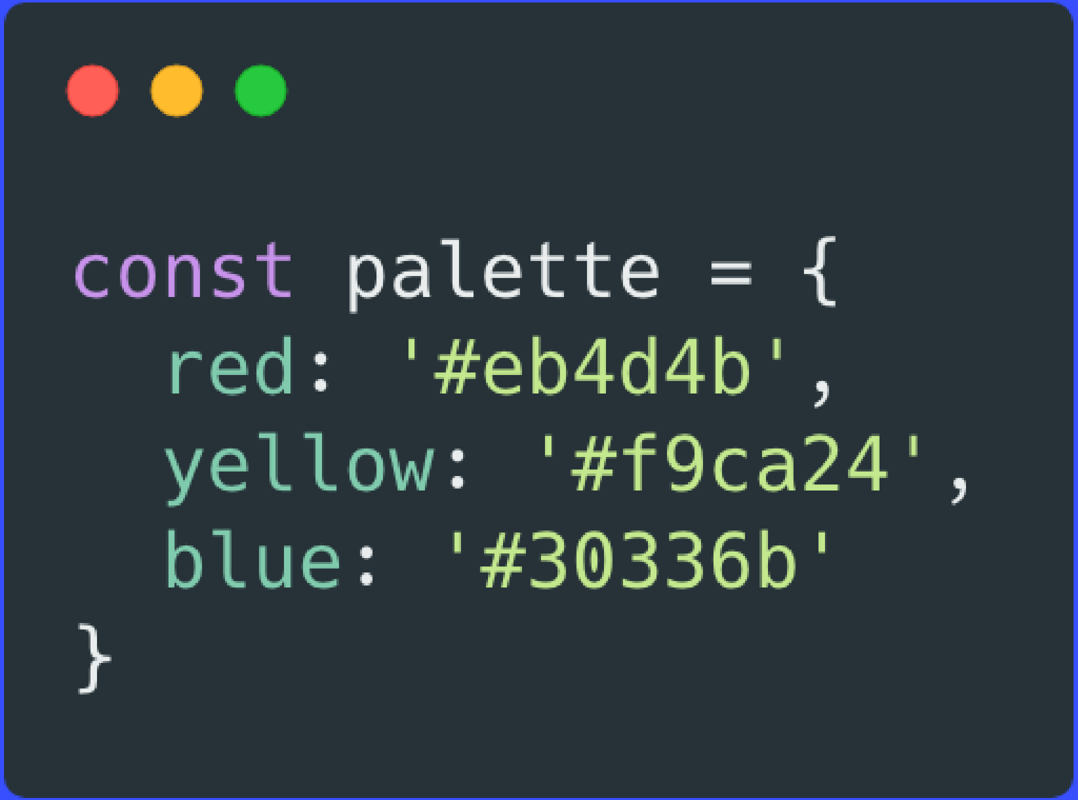
```
let comment = {
    username    : 'sillyGoose420',
    downVotes   : 19,
    upVotes     : 214,
    netScore    : 195,
    commentText : 'Tastes like chicken lol',
    tags: ['#hilarious', '#funny', '#silly'],
    isGilded: false
};
```

VALID KEYS

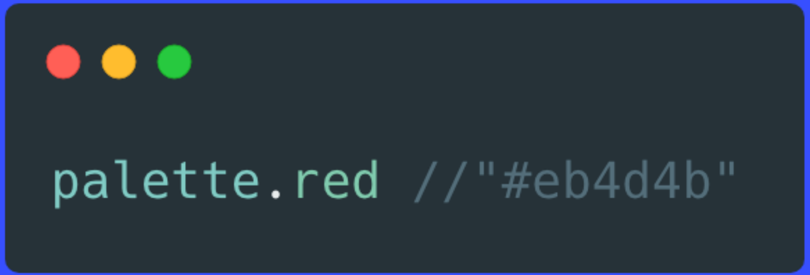
All keys are converted to strings *

* Except for Symbols, which we haven't covered yet

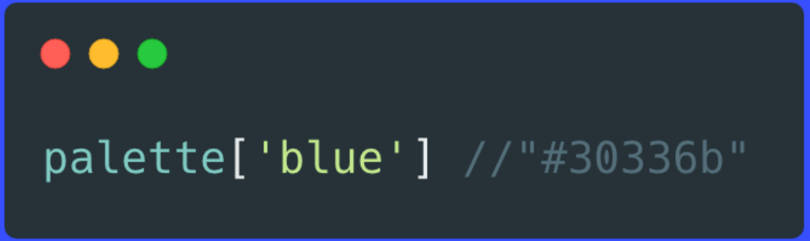
ACCESSING DATA



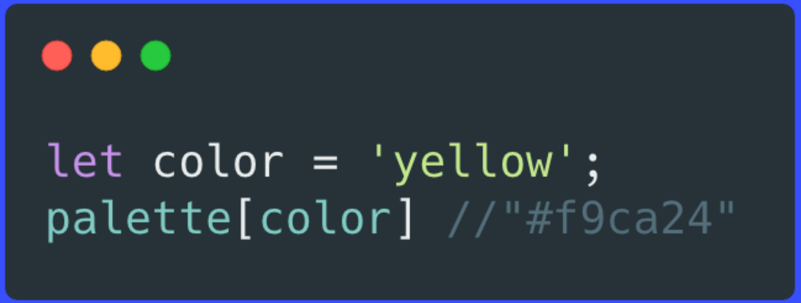
```
const palette = {
  red: '#eb4d4b',
  yellow: '#f9ca24',
  blue: '#30336b'
}
```



```
palette.red //"#eb4d4b"
```

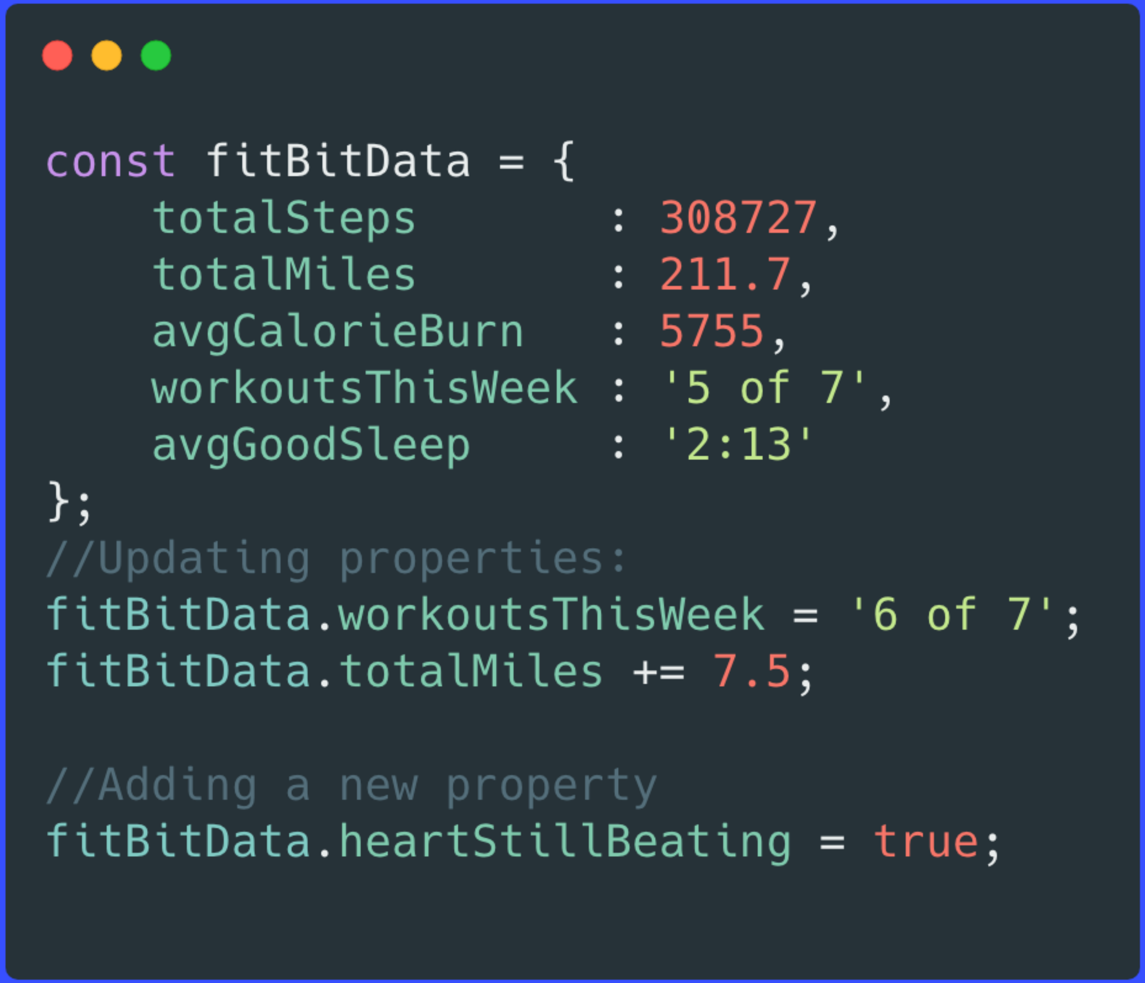


```
palette['blue'] //"#30336b"
```



```
let color = 'yellow';
palette[color] //"#f9ca24"
```

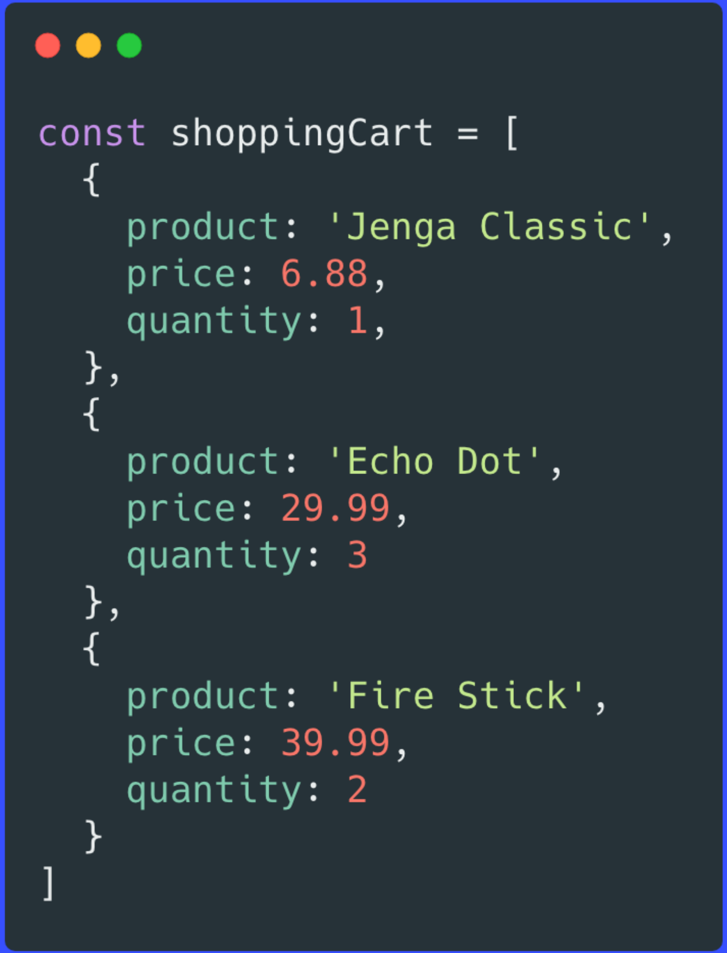
UPDATING & ADDING PROPERTIES



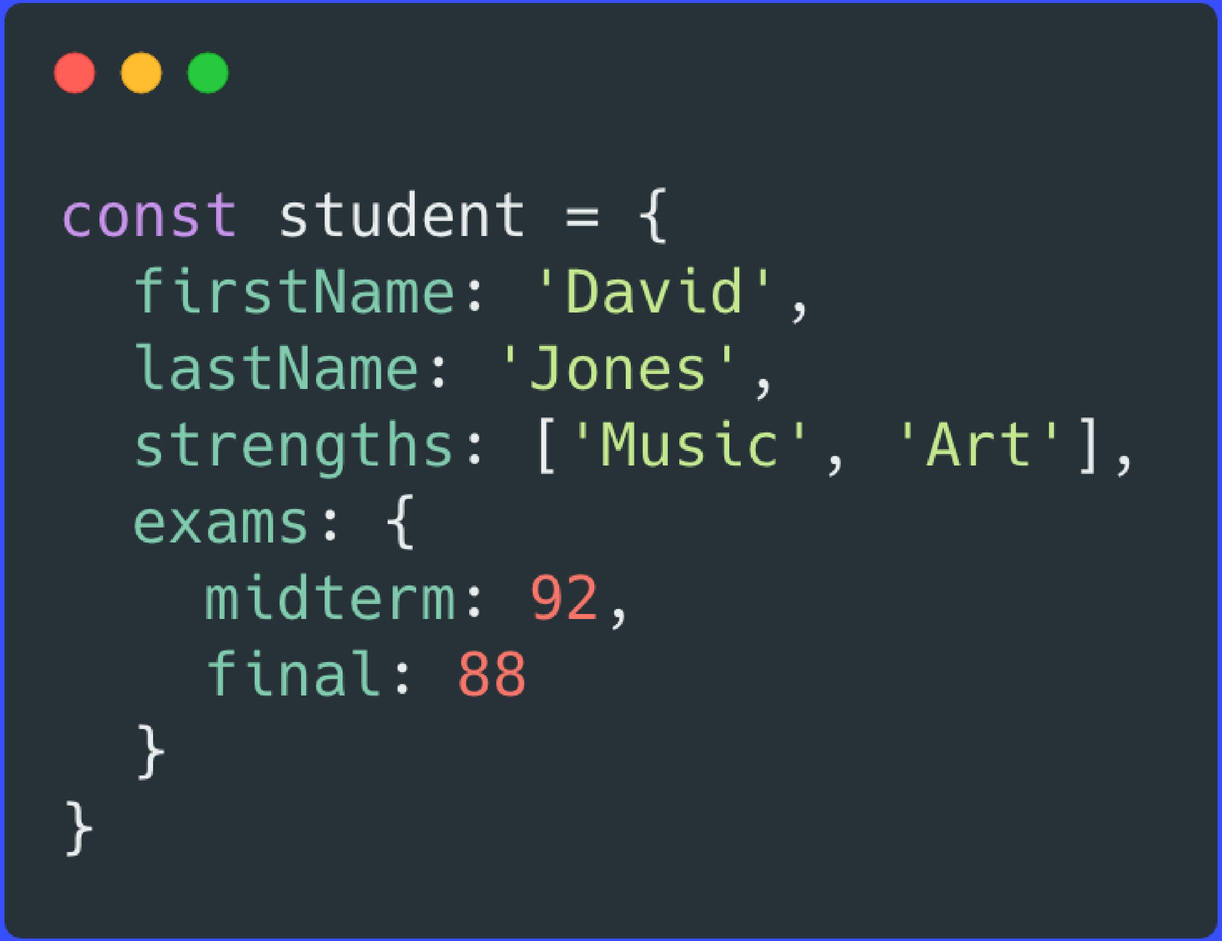
```
const fitBitData = {
    totalSteps       : 308727,
    totalMiles       : 211.7,
    avgCalorieBurn   : 5755,
    workoutsThisWeek : '5 of 7',
    avgGoodSleep     : '2:13'
};
//Updating properties:
fitBitData.workoutsThisWeek = '6 of 7';
fitBitData.totalMiles += 7.5;

//Adding a new property
fitBitData.heartStillBeating = true;
```

ARRAYS + OBJECTS



```
const shoppingCart = [
  {
    product: 'Jenga Classic',
    price: 6.88,
    quantity: 1,
  },
  {
    product: 'Echo Dot',
    price: 29.99,
    quantity: 3
  },
  {
    product: 'Fire Stick',
    price: 39.99,
    quantity: 2
  }
]
```



```
const student = {
  firstName: 'David',
  lastName: 'Jones',
  strengths: ['Music', 'Art'],
  exams: {
    midterm: 92,
    final: 88
  }
}
```

CHECKING FOR ARRAY & OBJECT EQUALITY