Objects

Ref Javascript

Objects & properties

An object groups related values under names. Each name-and-value pair is a property. The name is called the key:

```
const pupil = {
  name: "Mei",
  age: 17,
  form: "11A",
};
console.log(pupil.name);        // Mei
console.log(pupil["form"]);     // 11A  (bracket form, with the key as a
↪  string)
```

- Read a property with a dot (`pupil.age`) or with brackets (`pupil["age"]`). Use brackets when the key is in a variable: `pupil[field]`.
- A key that does not exist gives `undefined`.

Change a property, or add a new one, by assigning to it:

```
const pupil = { name: "Mei", age: 17 };
pupil.age = 18;                 // change
pupil.house = "Red";            // add
console.log(pupil);             // {"name":"Mei","age":18,"house":"Red"}
```

Arrays of objects

A list of records is an array of objects. Combine it with the array methods from topic 7:

```
const pupils = [
  { name: "Mei", mark: 88 },
  { name: "Tom", mark: 64 },
  { name: "Sara", mark: 95 },
];

const passed = pupils.filter((p) => p.mark >= 70);
console.log(passed.map((p) => p.name));            // ["Mei","Sara"]

const best = [...pupils].sort((a, b) => b.mark - a.mark)[0];
console.log(`${best.name} has the top mark`);       // Sara has the top
↪  mark
```

To total a property, use `reduce` or a loop:

```
const pupils = [{ mark: 88 }, { mark: 64 }, { mark: 95 }];
let sum = 0;
for (const p of pupils) {
  sum += p.mark;
}
console.log((sum / pupils.length).toFixed(1));      // 82.3
```

Looping over an object

`Object.keys` gives an array of the keys, and `Object.entries` gives `[key, value]` pairs, so you can loop over them:

```
const stock = { pens: 12, rulers: 0, books: 5 };

console.log(Object.keys(stock));              // ["pens","rulers","books"]

for (const [item, count] of Object.entries(stock)) {
  if (count === 0) {
    console.log(`${item}: sold out`);
  }
}
// rulers: sold out
```

An object is a good counter : use each thing you count as a key.

```
const words = ["red", "blue", "red", "red"];
const counts = {};
for (const w of words) {
  counts[w] = (counts[w] || 0) + 1;   // 0 the first time the key is seen
}
console.log(counts);                  // {"red":3,"blue":1}
```

Common mistakes

- `pupil.field` looks for a key called `field`. When the key is in a variable, write `pupil[field]`.
- A comma goes between properties, not a semicolon: `{ name: "Mei", age: 17 }`.
- `for...of` cannot loop over an object directly. Loop over `Object.keys(...)` or `Object.entries(...)`.