对象

Ref Javascript

Objects & properties

对象对象将相关的值按名称分组在一起。每个名称-值对称为属性属性。该名称被称为键键：

```
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)
```

- 使用点号（pupil.age）或方括号（pupil["age"]）读取属性。当键名存储在变量中时，必须使用方括号：pupil[field]。
- 访问不存在的键会返回 undefined。

通过赋值来修改现有属性或添加新属性：

```
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

记录列表本质上是一个对象数组。可结合主题 7 中的数组方法进行操作：

```
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
```

对某属性求和，可使用 reduce 或循环遍历：

```
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 返回键的数组，而 Object.entries 返回 [key, value] 对，因此可以遍历它们：

```
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
```

对象是理想的计数器：将每一个被统计的对象作为键使用。

```
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}
```

常见错误

- `pupil.field` 用于查找名为 `field` 的键。当键名存储在变量中时，应写成 `pupil[field]`。
- 属性之间用逗号分隔，而非分号：`{ name: "Mei", age: 17 }`。
- `for...of` 无法直接遍历对象。请遍历 `Object.keys(...)` 或 `Object.entries(...)`。