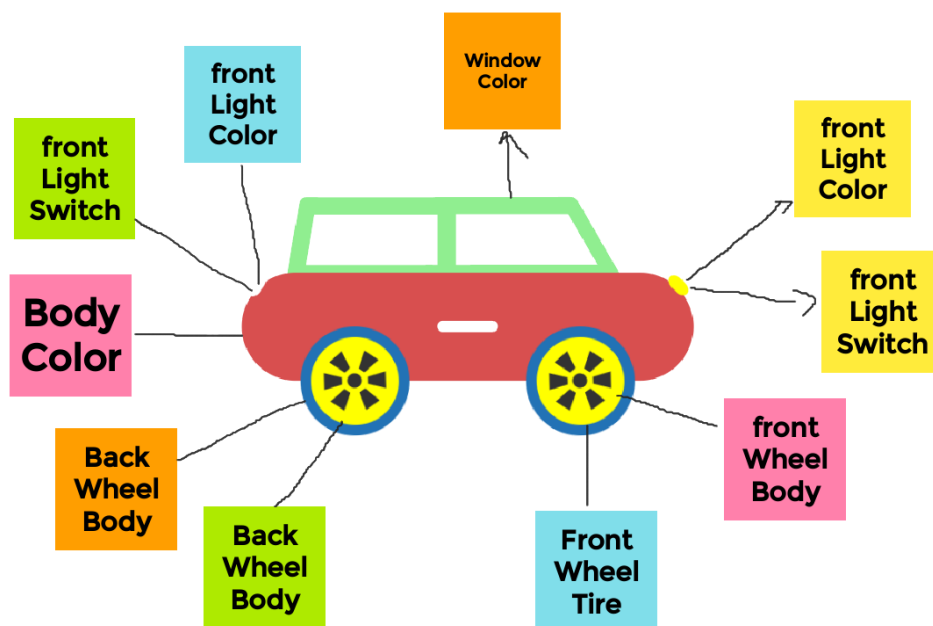
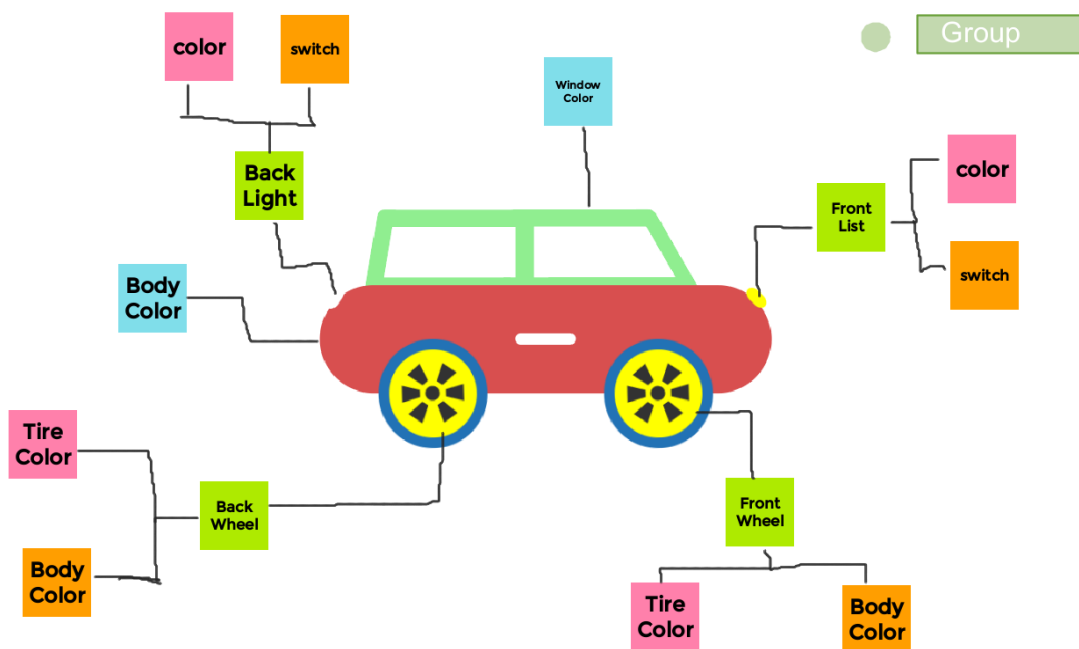


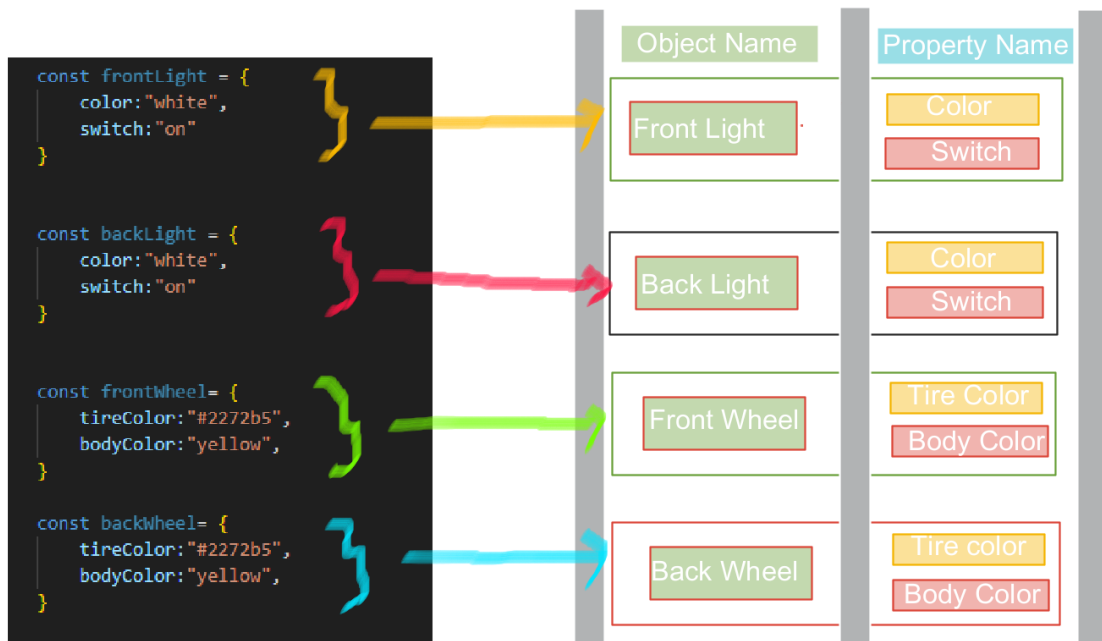
## Javascript Nested Object



আমরা আগের কার অবজেক্ট এর দিকে ভালো ভাবে লক্ষ্য করলে এটার আরো কিছু propertis খোঁজে পাই। যেমন: wheel এর বডি, টায়ার। লাইট এর সুইজ এবং কালার। এগুলোকে গ্রুপ করলে আর সুন্দর ভাবে চিন্তা করা যায় এত হিজিবিজি এতগুলো propertis না নিয়ে।



উপরের ছবিতে দেখা যাচ্ছে **propertis** গুলোকে আলাদা আলাদা গ্রুপ করে অল্প কি ওয়ার্ড দিয়ে সহজে অনেক গুলো **propertis** এ ভাগ করা যায়। এত মনে রাখা সহজ। আলাদা আলাদা পার্টকে কন্ট্রোল করাও সহজ। এখানে গ্রুপ হল এক একটা **object**। তার ভিতরে প্রায় একই নামের **propertis** বাকি গুলো।



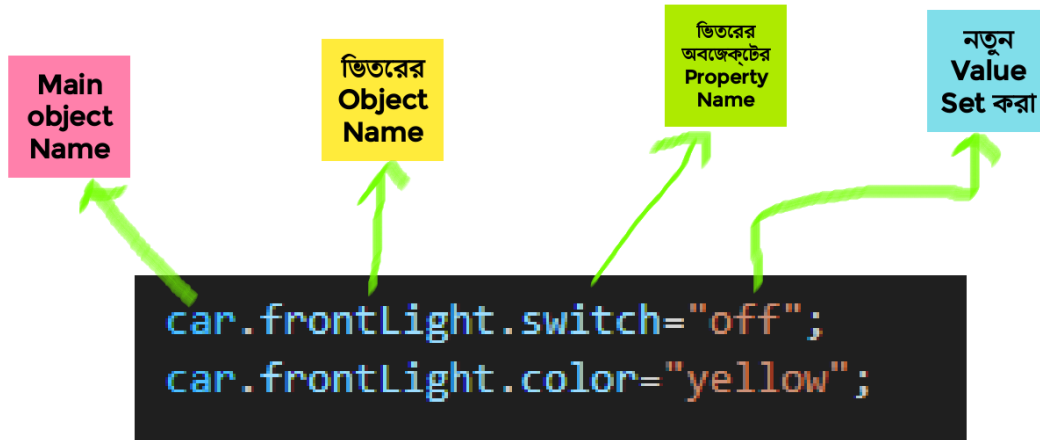
এই গ্রুপ গুলোকে যখন **Car object** এর **propertis** হিসেবে নিব তখন **object** এর ভিতরে অবজেক্ট হয়ে যাবে। এই অবজেক্টের ভিতরের অবজেক্ট কে **Nested Object** বলে।



অবজেক্টের ভিতর অবজেক্ট থাকলে ভিতরে থাকা অবজেক্টের **property** ধরার নিয়ম কি?

Main object Name এরপর ডট দিয়ে ভিতরের Object Name লিখে ডট দিয়ে এরপর ভিতরের অবজেক্টের property লিখে।

যেমনঃ



গত পিডিএফ এ লিখেছিলাম সব HTML Element Javascript এ এক একটা আলাদা অবজেক্ট। **HTML** থেকে একটা **Element** ধরে বা **get** করে যে **variable** এ নিব সে **variable** একটা সেই **HTML Element** এর একটা অবজেক্ট হয়ে যাবে।

যেমনঃ একটা HTML DIV element আছে।

```
<div class="box"></div>
```

একে জাভাস্ক্রিপ্টে Get করে box নামের variable এ রাখলাম। এখন এই box variable টা এই element এর একটি জাভাস্ক্রিপ্ট অবজেক্ট।

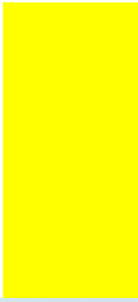
```
const box = document.querySelector(".box");
```

আর এই DIV Element এর Style গুলো হল এই box এর ভিতরে থাকা আরেকটা অবজেক্ট। মানে Nested Object। অনেকটা দেখতে নিচের ছবির মত।

```
box={
  onchange:null,
  oninput:null,
  style:{
    width:"200px",
    height:"440px",
    backgroundColor: "yellow",
  }
}
```

এটি একটি নুমনা মাত্র। এরকম ভিতরের অনেক propertis এবং method আছে।

**Output:**



এখন যদি আমি **background color change** করতে চাই কিভাবে করবো?

Main object Name এরপর ডট(.) দিয়ে ভিতরের Object Name লিখে ডট(.) দিয়ে এরপর ভিতরের অবজেক্টের property লিখে।

এখানে মেইন অবজেক্ট **box** এর ভিতরে আরেকটি অবজেক্ট property হিসেবে আছে। সেটি হল **style**. এবং এর ভিতরের property হল **backgroundColor**..... একে change করে দিলে div এর ব্যাগ্রাউন্ড কালার change হয়ে যাবে।

```
box.style.backgroundColor="red";
```

**Output:**



এভাবে যত **style** আছে সব **change** করা যায়।

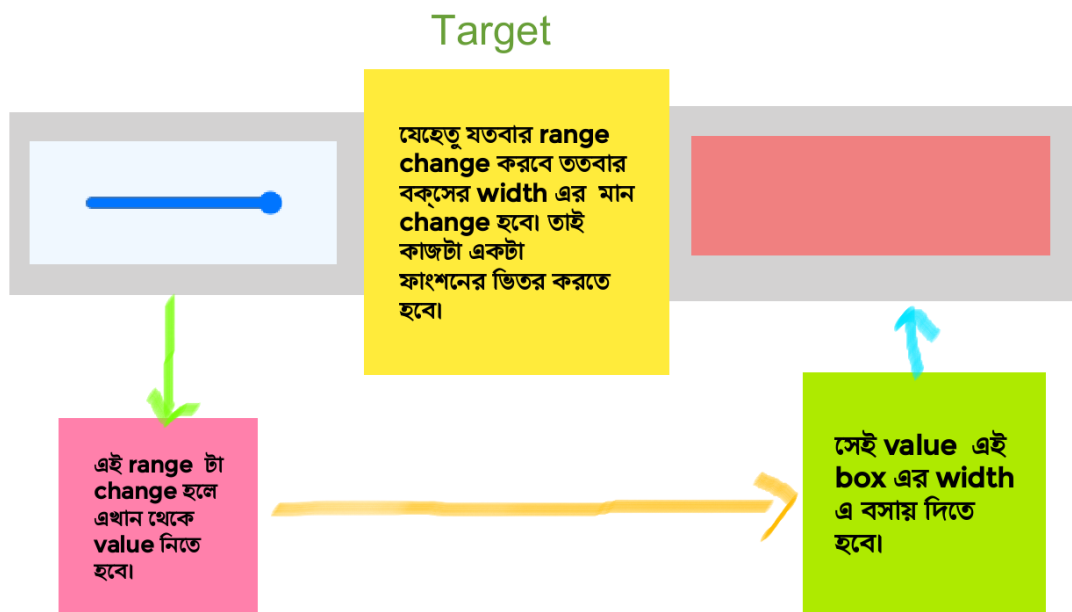
**nestedobject.html** ফাইল **run** করিয়ে বাকিটা নিজে নিজে করে দেখলে ক্লিয়ার হয়ে যাবে।

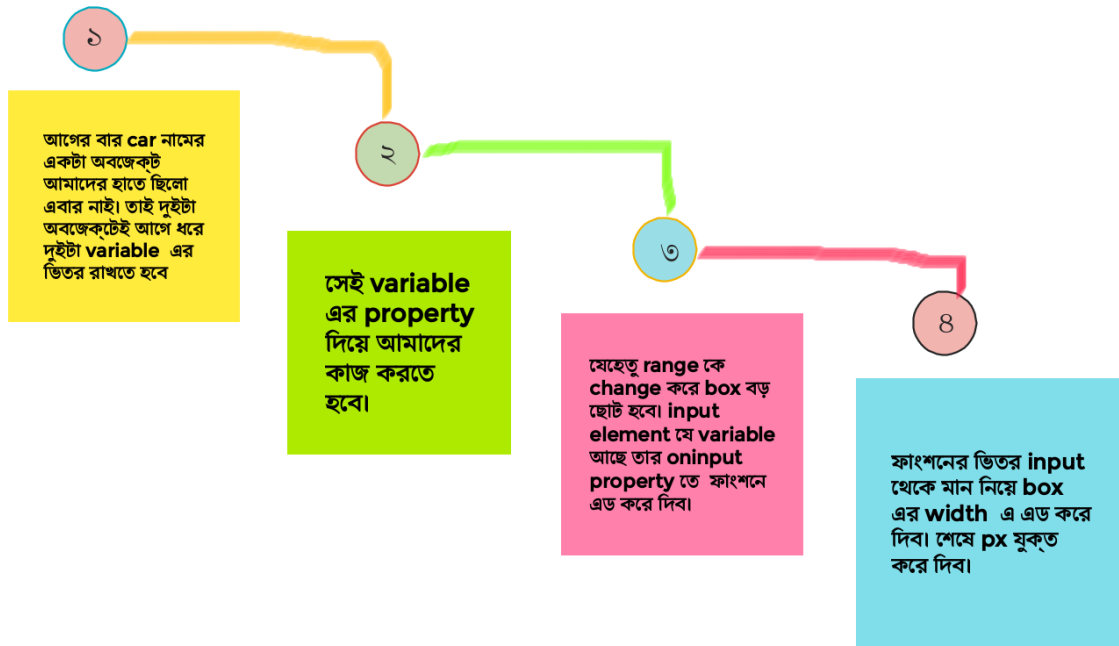
গতবার দেখেছিলাম **onchange property** এর ব্যবহার। আজকে দেখবো। **oninput** এর ব্যবহার।



**Problem 2:** **Range** এর মান কমানো বাড়ানোর সাথে সাথে লাল **Box** এর মান কমবে বা বাড়বে। মাউজ ছেড়ে না দিয়ে।

সব গতবার করা **onchange** এর মত যেহেতু রিয়েল টাইম বা মাউজ ড্রাগ করার সাথে সাথে তাল মিলে লাল বক্সের মান কমবে বা বাড়বে তাই। **onchange** এর জায়গায় **oninput** ব্যবহার করতে হবে।





একই নিয়মে **background-color change** করতে চাইলে কিভাবে করবো?  
একই নিয়মে **height** কিভাবে **change** করবো?