\*) Mookup:

+) Mookup là mô hình thu nhỏ, là hình ảnh mô phỏng mẫu thiết kế của designer dưới dạng file vector hoặc PSD được thiết kế sẵn. Mockup thường được đưa vào sử dụng để trình diễn, giảng dạy, đánh giá thiết kế, quảng bá hay các mục đích khác.

+) Một mookup là một nguyên mẫu cho phép thử nghiệm thiết kế và cung cấp ít nhất một phần chức năng của hệ thống. Mẫu Mockup xuất hiện ở khắp mọi thiết kế các sản phẩm đa dạng, từ các ấn phẩm văn phòng như namecard, folder, bộ nhận diện thương hiệu, website, banner, standee,… cho đến nội thất lẫn kiến trúc nhà cửa,… tạo cảm giác như thật một cách sống động cho người xem.

+) Trong thiết kế web , mookup được hiểu đơn giản là 1 file hình ảnh được thiết kế sẵn giúp cho nhà thiết kế website có thể mô tả lại hình mẫu website của mình khi được vận hành trên máy tính , điện thoại hoặc các thiết bị khác

+) Công việc này sẽ giúp cho chúng ta hoặc khách hang có cái nhìn bao quát , khách quan nhất về website trước khi đưa lên môi trường mạng

\*) Prototype:

+) Prototype là một dạng cơ chế quan trọng để thực hiện mô hình OOP của Javascript. Nó cho phép tạo nên sự kế thừa trong Javascript từ các object khác. Mỗi Prototype trong Javascript đều đi kèm với một object. Nó giúp những object này kế thừa những phương thức và thuộc tính đi kèm. Bản thân Prototype cũng được định nghĩa là một object trong Javascript, nhưng thuộc phân loại đối tượng Prototype, cần phân biệt đối tượng này với thuộc tính Prototype có trong Function.

+)Prototype Javascript sở hữu những cơ chế tuyệt vời phục vụ cho các bước lập trình. Nhờ cơ chế kế thừa và khả năng truy cập các đối tượng, prototype in javascript trở nên vô cùng đa nhiệm và sẵn sàng trợ giúp lập trình viên trong nhiều công đoạn. Để hiểu rõ hơn prototype là gì, bạn đọc cần ghi nhớ kỹ phần sau đây.

**Khả năng kế thừa – Prototype là gì?**

Điều làm nên sự đặc biệt của Prototype chắc chắn nằm ở tính kế thừa mà nó mang lại. Đây là một thuộc tính quan trọng làm nên sự cần thiết của Prototype. Như đã nói, Prototype mang đến sự kế thừa trong javascript. Điều này là duy nhất bởi javascript trong những phiên bản từ ES5 đổ về không được trang bị khái niệm class. Do đó javascript không thể được kế thừa và mở rộng các ứng dụng theo ngôn ngữ OOP.

Để duy trì khả năng này, lập trình viên sẽ cần sử dụng đến Prototype. Cơ chế prototype-based sẽ được tạo nên như một nền tảng cho thuộc tính này. Để tiến hành hoạt động kế thừa, lập trình viên cần khởi tạo nên một hàm. Sau đó hãy thêm vào các thuộc tính cũng như phương thức thuộc tính Prototype cho hàm vừa khởi tạo này. Khi đó instance được tạo ra từ hàm khởi tạo sẽ chứa các phương thức và thuộc tính được định nghĩa sẵn theo mong muốn của bạn. Prototype sẽ là cách duy nhất để thực hiện kế thừa và mở rộng ngôn ngữ OOP trên hệ javascript.

### Khả năng truy cập vào thuộc tính đối tượng

Một nhiệm vụ quan trọng nữa của Prototype là giúp lập trình viên xác định các thuộc tính và phương thức trên một đối tượng. Định nghĩa Prototype object hay Property của Object là định nghĩa dùng để chỉ nơi chứa đựng những thuộc tính, phương thức được kế thừa. Bạn chỉ cần truy cập vào một Property của Object, javascript sẽ tự động khởi tạo quá trình tìm kiếm thuộc tính hoặc phương thức mà bạn đã gọi tên. Đặc tính riêng biệt này của Prototype được biết đến như một loại object cha. Đó là lý do tại sao khi bạn gọi tên một thuộc tính của một đối tượng, javascript sẽ tiến hành tìm kiếm trên khu vực riêng biệt của đối tượng này trước. Nếu có kết quả, javascript sẽ tự động xuất ra. Còn nếu không có, nó lại tiếp tục tìm kiếm trên Prototype của object này. Và cứ thế là Prototype của đối tượng Prototype… Một chu trình lặp lại như vậy được gọi là chuỗi prototype của Javascript. Cả chu trình này và các thuộc tính prototype có trong function đều góp phần mang đến cơ chế kế thừa prototype-based trong Javascript.

+) Nên sử dụng protptype vì :

Giống như hầu hết các mẫu tạo lập, Prototype trong Javascript cho phép ẩn đi quá trình tạo đối tượng từ client. Do vậy, thay cho việc tạo ra một số đối tượng không được thiết lập sau đó trả về, nó sẽ sao chép những giá trị có sẵn từ một đối tượng kiểu mẫu rồi mới tiến hành đến việc tạo lập đối tượng mới được dựa trên những dạng phương thức mà ta vừa học được đó.