**HTML5 Canvas**

Web Storage

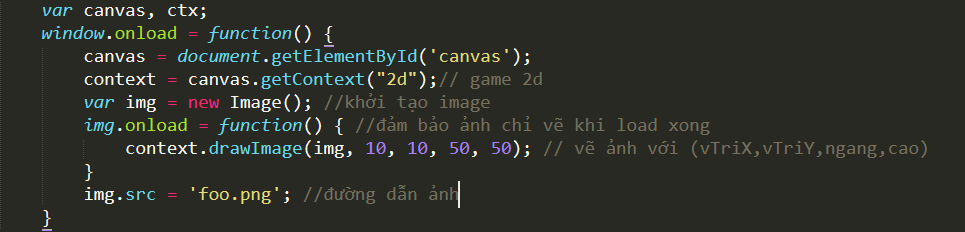
* *sessionStorage*: giới hạn trong một cửa sổ hoăc thẻ của trình duyệt. Một trang web được mở trong hai thẻ của cùng một trình duyệt cũng không thể truy xuất dữ liệu lẫn nhau. Như vậy, khi bạn đóng trang web thì dữ liệu lưu trong sessionStorage hiện tại cũng bị xóa.
* *localStorage*: có thể truy xuất lẫn nhau giữa các cửa sổ trình duyệt. Dữ liệu sẽ được lưu trữ không giới hạn thời gian.

Ví dụ:

sessionStorage.myVariable="Hello. ";

localStorage.myVariable="Nice to meet you!";

Vẽ ảnh:



*context . putImageData(imagedata, dx, dy [, dirtyX, dirtyY, dirtyWidth, dirtyHeight ]) ->* Vẽ dữ liệu từ đối tượng ImageData lên canvas tại vị trí dx, dy. Nếu như hình chữ nhật (từ các tham số dirtyX, dirtyY, dirtWidth, dirtyHeight) được xác định, thì phần dữ liệu của ImageData trong vùng chữ nhật này mới được vẽ lên canvas

*imagedata.width  
imagedata.height*Trả về kích thước thật của đối tượng ImageData, tính theo pixel

var imageData = getImageData(0, 0, img.width, img.height);

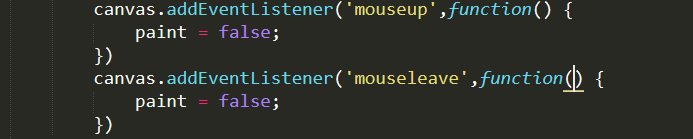
lấy ảnh tại vị trí 0,0 bằng đúng chiều dài, rộng ảnh đó (gọi là copy)

*context.putImageData(imageData, 0, 0)*

*vẽ ảnh vừa copy được lên.*

1. Xác định tọa độ chuột





|  |  |
| --- | --- |
| ctx.fillText("Big smile!",10,90); | Xuất text trong canvas |
| Ctx.fillStyle = “#fff” | Màu sắc cho chữ |
| Ctx.clearRect(0, 0, 300, 1500) | Xóa background |
| Ctx.fillRect(0,0,300,150); | Tô background (viết thêm fillStyle và ctx.fill()->tô) |
| ctx.beginPath(); | Bắt đầu vẽ, or reset lại path. |
| Ctx.arc(VTngang, VTdoc, bk, 0, 2\*Math.PI) | Vẽ hình tròn |
| ctx.globalAlpha = 0; | transparent |
| localStorage.name | Lưu như session |