New global attrs:

* contentEditable , draggable , hidden , spellcheck

page structure:

* <header><nav><footer><section><article><aside><figure>

Multimedia

* Video:

<video width="320" height="240" controls="controls" autopley=”autoplay” loop=”loop” muted=”muted” poster=”URL”>

<source src="movie.mp4" type="video/mp4" />

<source src="movie.ogg" type="video/ogg" />

Your browser does not support video tag.

</video>

* Audio:

<audio controls="controls" loop=”loop”>

<source src="song.ogg" type="audio/ogg" />

<source src="song.mp3" type="audio/mpeg" />

Your browser does not support audio element.

</audio>

* Event : oncanplay , onemptied,onended,onerror,onloadstart,onpause,onplay,onplaying,onvolumechange

Forms:

* New input type : email , date , month , day , url , range , color
* New attrs for input type : max , min , step , pattern="[A-z]{2}" , autofocus
* list :

web page: <input type="url" list="url\_list" name="link" />

<datalist id="url\_list">

<option label="W3Schools" value="http://www.w3schools.com" />

<option label="Google" value="http://www.google.com" />

<option label="Microsoft" value="http://www.microsoft.com" />

</datalist>

* association:

<form id="foo"> ... </form>

<textarea form="foo"></textarea

Storage:

* session storage : sessionStorage.lastname = “karimi”

deleted when closed tab browser

* local storage : localStorage.lastname = “karimi”

methods:

localstorage.removeItem(lastname)

localstorage.clear()

drag&drop:

* events:

draggable element:

ondragstart , ondragend

targetelement :

ondragover -> return false , ondragenter , ondrop -> must return false after commands

* drag object out of web page : only dragstart and dragend event fire . use event.dataTransfer.setData
* drag object into web page : only dragover , dragenter , drop event fire . use event.dataTransfer.getData

canvas:

* access:

<canvas id="canvas\_id" width="200" height="100">

Your browser does not support Canvas

</canvas>

var canvas = document.getElementByID("canvas\_id");

var ctx = canvas.getContext("2d");

* global setting:

ctx.fillStyle = "rgb(R, G, B)";

ctx.strokeStyle = "rgb(R, G, B)";

ctx.lineWidth = 5;

* shapes :

ctx.fillRect(X, Y, W, H);

ctx.strokeRect(X, Y, W, H);

ctx.fillText(text,x,y,maxWidth)

ctx.strokeText(text,x,y,maxWidth)

* paths:

A new path: ctx.beginPath()

Move the pen: ctx.moveTo(X,Y)

Line: ctx.lineTo(X,Y)

Arc: ctx.arc(X,Y,R,sAngle,eAngle,anticlock);

Close the path: ctx.closePath()

* gradiant:

ctx.createLinearGradient(x0,y0,x1,y1)

ctx.createRadialGradient(x0,y0,r0,x1,y1,r1)

* insert image :

var img=new Image();

img.src="URL";

cxt.drawImage(img,0,0);

* save & restore:

save and restore state ( not drawing) :

ctx.save(): push state on stack

ctx.restore(): pop sate from stack

save & restore drawing:

image date :

ctx.getImageData(X,Y,width, height);

ctx.putImageData(imagedata, X, Y);

data url:

var canv = document.getElementById("canvas");

imageurl = canv.toDataURL("image/png");