HTML značky <video> a <canvas> (10)

HTML značka <video>

- vložení videa do stránky
- alternativa pro flashové přehrávače
- syntaxe téměř shodná s audiem
- ještě výraznější problematika formátů
- s výhodou použití **<source media="…" />**

```
<video controls poster="image.png">
    <source src="video-small.mp4"
        type="video/mp4; codecs='avc1.42E01E, mp4a.40.2'"
        media="screen and (max-width:500px)"
    />
    <source src="video.mp4" type="video/mp4; codecs='avc1.42E01E, mp4a.40.2'" />
    <source src="video.ogv" type="video/ogg; codecs='theora, vorbis'"/>
    <source src="video.webm" type="video/webm; codecs='vp8, vorbis'" />
    <!-- obrázek či odkaz jako fallback -->
</video>
```

Media source Extensions

- tradiční HTML5 audio/video přehrává jen jeden soubor
- MSE dovoluje definovat zdroj multimediálních dat s větší granularitou
- namísto souboru s pevným trváním se přenáší **stream**
- změna datového toku za běhu
- protokol mediasource:, JS API MediaSource

HTML značka <canvas>

- canvas je HTML značka, rastrová kreslící plocha
- bohaté JS API, výborná podpora prohlížečů

- **canvas v HTML:**
  - fallback text
  - pevné rozměry (změna = vymazání)
  - průhledné pozadí

```
<canvas width="800" height="600">Smolíček pacholíček :-(</canvas>
```

- **canvas v JS:**
  - malovací metody patří kontextu
  - existuje též 3D kontext (WebGL)
  - **cheatsheet**: http://www.webmastersucks.com/uploads/HTML5_Canvas_Cheat_Sheet.png

```
1. var canvas = document.querySelector("canvas");
2. var context = canvas.getContext("2d");
3. context.namalujNecoPekneho();
```

Malování, transformace

Obdélníky

- **clearRect(x, y, w, h)** = vymazat
- **fillRect(x, y, w, h)** = vyplnit (aktuálně nastavenou vyplňovací barvou)
- **strokeRect(x, y, w, h)** = orámovat (aktuálně nastavenou kreslící barvou)

Tah štětcem

- **ukázka**: http://ondras.zarovi.cz/slides/2011/html5/3.html#7

```
1. ctx.beginPath();
2. ctx.moveTo(100, 200);
3. ctx.lineTo(200, 300);
4. ctx.arc(300, 300, 50, 0, Math.PI, true);
5. ctx.stroke();
6. ctx.fill();
```

Vzhledové vlastnosti

- `ctx.strokeStyle = "red"`
- `ctx.fillStyle = "#000"`
- CSS barva NEBO barevný přechod NEBO vzor (viz dále)
- `ctx.globalAlpha = 0.5`
- `ctx.globalCompositeOperation = "lighter"`, dokumen
- `ctx.shadow{Blur,Color,OffsetX,OffsetY} = ...`

Vlastnosti čar

- *ctx.lineWidth = 3*
- *ctx.lineCap = „round“*
- *ctx.linmeJoin = „round“*
- **ukázka**: https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Applying_styles_and_colors#Line_styles

Text

```
1. ctx.font = "bold 20px arial";
2. ctx.textAlign = "middle";
3. ctx.textBaseline = "bottom";
4.
5. ctx.fillText("Ahoj", x, y);
6. ctx.strokeText("Ahoj", x, y);
7. var w = ctx.measureText("A").width;
```

Po pixelech

```
var data = ctx.getImageData(x, y, w, h);
/* data.data.length == w*h*4 */

data.data[0] = 100; /* R */
data.data[1] = 200; /* G */
data.data[2] =  50; /* B */
data.data[3] = 255; /* A */

ctx.putImageData(data, 0, 0);
```

- **ukázky**:
  - http://ondras.github.io/coral/
  - http://ondras.github.io/fractal/
  - http://ondras.github.io/primitive.js/

Barevné přechody a vzory

```
var g = ctx.createLinearGradient(0, 0, 100, 100);
g.addColorStop(0, "red");
g.addColorStop(1, "blue");
ctx.fillStyle = g;

var p = ctx.createPattern(image);
var p = ctx.createPattern(canvas);
ctx.fillStyle = p;
```

Transformace

- podobné jako u SVG
- sada afinních transformací, která upravuje souřadný systém
- *ctx.scale(0.5, 0.5)*
- *ctx.rotate(Math.PI/2)*
- *ctx.translate(dx, dy)*
- **ukázka**: http://ondras.zarovi.cz/demos/space-filling/

Animace, časování

- canvas je ideální technologie pro animace a hry
- správná práce s canvasem je rychlá
- podstatné je dobré časování a množství překreslování
- ukázka: http://ondras.github.io/drago/game/

```
var x = 0;
while (true) {
    ctx.clearRect();
    x += 3;
    ctx.drawRect(x, 0, 10, 10);
}
```

```
var x = 0;
var draw = function() {
    ctx.clearRect();
    x += 3;
    ctx.drawRect(x, 0, 10, 10);
}

setInterval(draw, 1000/30);
```

```
var x = 0;
var draw = function() {
    ctx.clearRect();
    x += 3;
    ctx.drawRect(x, 0, 10, 10);
    requestAnimationFrame(draw);
}

draw();
```

```
var T = Date.now();
var speed = 0.1;

var draw = function() {
    var t = Date.now();
    x += (T - t) * speed;
    T = t;
    /* ... */
}
```

Komunikace s dalšími prvky

Export

- **var data = canvas.toDataURL(„image/png“)**
- technika data URI představuje reprezentaci dat přímo v řetězci URI
- možno nastavit např. jako src obrázku
- **ukázka**: http://ondras.github.io/photo/

Import

- **ctx.drawImage(image, dx, dy, [dw, dh])**
- **ctx.drawImage(image, sx, sy, sw, sh, dx, dy, dw, dh)**
- „image“ může být obrázek, canvas či video

Tipy a triky

Ostré hrany

- canvas používá antialiasing, nelze vypnout
- souřadný systém má celočíselné hodnoty mezi pixely
- svislé/vodorovné čáry o liché tloušťce posuňme o půl pixelu
- **ukázka**: https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Canvas_tutorial/Applying_styles_and_colors#A_lineWidth_example

FOUT

- kreslení textu při použití @font-face
- dokud není písmo načteno, použije se výchozí
- soubor s písmem lze specifikovat pomocí data-URI

Animace

- překreslení (mazání) celého canvasu je drahé
- pokud se vyplatí, je lepší mazat jen změněné části
- různé kreslící operace jsou různě drahé, nejlepší je **drawImage**
- **benchmark**: http://ondras.github.io/html5-animation-framework/benchmark.html

WebGL

- akcelerovaná 3D grafika
- nové API HTML5 canvasu
- kompatibilita API s OpenGL ES 2.0

- **koncepty**:
  - **canvas.getContext("webgl")** || **canvas.getContext("experimental-webgl")**
  - veliké API
  - nové datové typy (typovaná pole)
  - kompatibilní shadery (GLSL)
  - nejen 3D scény, ale třeba i částicové systémy

- **obtíže**:
  - nejistá podpora
  - práce s maticemi
  - silně ukecané (stejně jako OpenGL)

- budoucnost grafiky na webu
- použití knihovny pro matice – *glMatrix*
- použití knihovny pro WebGL – *three.js*

WebRTC

- **RTC = Real Time Communications**
- technologie pro komunikaci mezi klienty (bez serveru)
- primárně přenos zvuku a obrazu, sekundárně JS dat
- obtížné navazování spojení (tzv. *signalling* – nutný server pro iniciální nastavení)

- **getUserMedia**
  - **navigator.mediaDevices.getUserMedia()**
  - nejlépe podporovaná komponenta WebRTC
  - přístup ke kameře a mikrofonu
  - prefixované varinty

```
1. var ok = function(stream) {
2.     video.srcObject = stream;
3.     video.play();
4. }
5.
6. var error = function(e) {
7.     alert(e);
8. }
9.
10. var options = {audio:true, video:true};
11. navigator.mediaDevices.getUserMedia(options).then(ok, error);
```

  - **ukázka**: https://jsfiddle.net/smap/6fgu4/15/

- **getDisplayMedia**
  - nejnovější přírustek pro navigator.mediaDevices
  - videostream plochy uživatele
  - uživatel volí, bude-li poskytovat plochu, prohlížeč či jeho záložku
  - použitelné též pro screenshoty