

I've seen the
FUTURE
It's in my
BROWSER



Programa Aula 4

- IndexedDB;
- Drag and Drop;
- AppCache;
- Media API;

IndexedDB

IndexedDB - Contexto

- É uma *base de dados* que fica guardada no navegador;
- Não é uma base de dados relacional;
- Permite guardar objetos JavaScript;
- Assíncrona;
- Transacional;

IndexedDB - Suporte

- Internet Explorer 10+;
- Chrome;
- Firefox;
- Opera;
- Android (4.4);

-
- iOS e Safari suportam *WebSQL*

IndexedDB - Funcionalidades

- Adicionar entradas;
- Remover entradas;
- Percorrer todas as entradas;
- Alterações à estrutura;

IndexedDB - Verificar disponibilidade

```
if (window.indexedDB) {  
  
    /* indexedDB disponível */  
  
} else {  
  
    /* indexedDB não disponível */  
  
}
```

IndexedDB - Abrir base de dados

```
var version = 1;
var request = indexedDB.open("shoppinglist", version);
request.onupgradeneeded = function (event) {
    var db = event.target.result;
    if ( db.objectStoreNames.contains("shoppingitem") ) {
        db.deleteObjectStore("shoppingitem");
    }
    var objStore = db.createObjectStore("shoppingitem", { keyPath: "datahora" });
}
request.onsuccess = function (event) {
    ns.indexedDB.db = event.target.result;
}
```


IndexedDB - Adicionar item

```
var db = ns.indexedDB.db;
var transaction = db.transaction(["shoppingitem"], "readwrite");
var objStore = transaction.objectStore("shoppingitem");
var now = new Date().getTime();
var request = objStore.put({
    "text": itemText,
    "datahora": now
});
request.onsuccess = function(event) {
    ns.indexedDB.getAllShoppingItems();
}
```

IndexedDB - Remover item

```
var db = ns.indexedDB.db;

var transaction = db.transaction(["shoppingitem"], "readwrite");

var objStore = transaction.objectStore("shoppingitem");

var request = objStore.delete(id);

request.onsuccess = function (event) {

    // fazer o que quisermos

}
```

IndexedDB - Percorrer items

```
var db = ns.indexedDB.db;
var transaction = db.transaction(["shoppingitem"], "readwrite");
var objStore = transaction.objectStore("shoppingitem");
var keysRange = IDBKeyRange.lowerBound(0);
var cursor = objStore.openCursor(keysRange);
cursor.onsuccess = function (event) {
    var result = event.target.result;
    if (!!result == false) {
        return;
    }
    // fazemos o que quisermos com o result
    result.continue();
}
```

IndexedDB - Demo

http://bit.ly/VC_HTML5_IndexedDB

IndexedDB - A reter

- API complexa (comparada com local storage);
- Maior capacidade de armazenamento (50MB+ no Firefox);
- Mais rápida para grandes quantidades de dados (desde que corretamente indexados);
- Transaccional;

Drag and Drop

Drag and Drop - Contexto

- Permite arrastar e soltar elementos numa página, recorrendo a pouco JavaScript;
- Permite arrastar ficheiros para dentro do navegador, para processamento;
- Algumas funcionalidades já eram possíveis anteriormente com JavaScript;

Drag and Drop - Suporte

- Internet Explorer 10+;
- Chrome;
- Firefox;
- Safari;
- Opera;

Drag and Drop - Verificar disponibilidade

```
if('draggable' in document.createElement('span')) {  
    /* drag and drop disponível */  
} else {  
    /* drag and drop indisponível */  
}
```

PS: não é sempre 100% exacto. É necessário testar mais eventos e algumas plataformas em particular (iOS).

Demo: http://bit.ly/VC_HTML5_Demo32

Drag and Drop - Arrastar

- Para permitir que um elemento seja arrastável no ecrã basta adicionar o atributo *draggable* com o valor *true*:

`<div draggable="true"></div>`

Base: http://bit.ly/VC_HTML5_Base33

Exercício 33

Drag and Drop - Eventos

- dragstart
 - Disparado quando o elemento começa a ser arrastado
- drag
 - Quando estamos a arrastar;
- dragenter
 - Quando entramos com um elemento em arrastamento por cima de outro (dispara no que está por trás)
- dragleave
 - Quando saímos com um elemento em arrastamento de cima de outro (dispara no que está por trás)
- dragover
 - Quando estamos com um elemento em arrastamento por cima de outro (dispara no que está por trás)
- drop
 - Quando soltamos o elemento (dispara no que está por trás);
- dragend
 - Quando soltamos o elemento (dispara no que está a ser arrastado);

Drag and Drop - Exercício dragstart e dragend

- Alterar a cor de texto da caixa para azul quando se inicia o drag;
- Alterar a cor de texto da caixa para branco quando se solta o elemento;

Base: http://bit.ly/VC_HTML5_Base34

[Exercício 34](#)

Drag and Drop - Exercício dragenter e dragleave

- Adicionar a class over quando entramos dentro de elemento a arrastar;
- Remover a class over quando saímos de um elemento a arrastar;

Base: http://bit.ly/VC_HTML5_Base35

Exercício 35

Drag and Drop - Demo dragover e drop

- Como trocar a ordem das colunas ao soltar o elemento;
- Utilização do *dataTransfer* (*setData* e *getData*);

Demo: http://bit.ly/VC_HTML5_Demo36

Drag and Drop - Demo file drag

- Como aceder ao ficheiro (ou ficheiros) que foi arrastado;
- Utilização do *dataTransfer.files*;

Demo: http://bit.ly/VC_HTML5_Demo37

Para aceder aos conteúdos do ficheiro é necessário usar o objeto *FileReader()*;

Drag and Drop - A reter

- API simples de usar;
- Permite a abertura de ficheiros no navegador;
- Pode ser usado para permitir o upload de ficheiros (tipo gmail);
- Permite arrastar múltiplos ficheiros;

Application Cache

ApplicationCache - Contexto

- Permite que as aplicações funcionem offline;
- O programador pode definir quais os ficheiros que quer disponibilizar offline;
- Permite mostrar conteúdo se o site estiver em baixo (p.e. manutenção);

ApplicationCache - Suporte

- Internet Explorer 10+;
- Chrome;
- Firefox;
- Safari;
- Opera;

ApplicationCache - Verificar disponibilidade

```
if ('applicationCache' in window) {  
    /* applicationCache disponível */  
} else {  
    /* applicationCache indisponível */  
}
```

Base: http://bit.ly/VC_HTML5_Base38

[Exercício 38](#)

ApplicationCache - Manifesto

- Ficheiro de texto normal;
- Tem que ser servido como “*text/cache-manifest*”;
- Pode ter qualquer extensão (comum: *.appcache*);
- Define quais os ficheiros a *cache*ar;
- Primeira linha é sempre:
CACHE MANIFEST

ApplicationCache - Exemplo manifesto

CACHE MANIFEST

version 1

/logo.png

/styles.css

Demo: <http://vascocosta.net/appcache/>

Carregar o site // Desligar internet // Recarregar

ApplicationCache - Recursos

- Master
 - Carregados por omissão (documento que declara appcache);
- Explicit
 - Recursos declarados na zona CACHE MANIFEST
- Network
 - Recursos declarados na zona NETWORK
 - Recursos que necessitam de acesso à rede
- Fallback
 - Recursos declarados na zona FALLBACK
 - Permite configurar uma segunda fonte de dados (local) caso algum acesso falhe

ApplicationCache - Exemplo manifesto fallback

CACHE MANIFEST

version 2

/logo.png

/styles.css

FALLBACK:

/appcache/page2.html /appcache/offline.html

Demo: <http://vascocosta.net/appcache/>

Carregar o site // Desligar internet // Carregar em “Página 2”

ApplicationCache - Estados

- CHECKING [2]
 - O manifesto está a ser descarregado
- DOWNLOADING [3]
 - Recursos estão a ser descarregados
- IDLE [1]
 - Não está a ser actualizada
- OBSOLETE [5]
 - A AppCache está obsoleta
- UNCACHED [0]
 - O objeto ainda não foi inicializado
- UPDATEREADY [4]
 - Existe uma nova versão da AppCache

O número entre [?] representa o número resultado da operação *window.applicationCache.status*

ApplicationCache - Eventos

- `cached`
 - Dispara após a primeira cache do manifesto
- `checking`
 - Dispara quando está verificar se existe algum update
- `downloading`
 - Dispara quando um update foi encontrado e os recursos estão a ser descarregados
- `error`
 - Dispara quando o servidor dá 404 (not found) ou 410 (gone), o download falhou, ou o manifesto mudou enquanto estava a ser descarregado

ApplicationCache - Mais eventos

- `noupdate`
 - Dispara após o primeiro download do manifesto
- `obsolete`
 - Dispara quando o servidor dá 404 ou 410, e apaga a cache toda
- `progress`
 - Dispara para cada recurso definido no manifesto, antes de descarregar
- `updateready`
 - Dispara quando todos os recursos foram descarregados (apenas para o segundo download e seguintes)

ApplicationCache - Diversos

- `window.applicationCache.update()`
 - Verifica se existe algum manifesto novo
- `window.applicationCache.swapCache()`
 - Usado normalmente na callback *updateready*, para forçar a mudança de cache

ApplicationCache - A reter

- Apenas precisa de um ficheiro bem configurado para funcionar
- Excelente para proporcionar suporte offline
- Útil para proporcionar uma experiência menos negativa no caso de falhas de rede

Media API

Media API - Contexto

- Permite aceder aos recursos de media do navegador sem plugins
- Webcam
- Microfone

Media API - Suporte

- Chrome 29+
- Firefox 24+
- Opera 18+

Media API - Verificar disponibilidade

```
function getUserMediaAvailable() {  
    return !!( navigator.getUserMedia ||  
                navigator.webkitGetUserMedia ||  
                navigator.mozGetUserMedia  
            );  
}
```

Demo: http://bit.ly/VC_HTML5_Demo40

Media API - Pedir acesso à webcam

```
getUserMedia({  
  video: true  
},  
callbackSucesso,  
callbackErro)
```

Media API - Captar vídeo

```
getUserMedia({  
    video: true  
}, function(stream) {  
    videoElement.src = window.URL.createObjectURL(stream);  
}, callbackError)
```

Demo: http://bit.ly/VC_HTML5_Demo41

Media API - Video a reter

- Pode ser capturado e feito o stream para um elemento video na página;
- É possível gravar o vídeo em ficheiro webm;
- É possível enviar o vídeo para o servidor, e guardá-lo ou processá-lo;
- Alguns navegadores permitem escolher a fonte de video;

Media API - Audio

- Usado especialmente para WebRTC (Web Real Time Communications);
- API ainda em Draft e bastante complexa;
- Pode ser usada para processar audio;
- Pode ser usada para captar audio;

Media API - Recursos de audio

- http://bit.ly/VC_HTML5_Audio1
 - Live input effects
- http://bit.ly/VC_HTML5_Audio2
 - Audio recorder
- http://bit.ly/VC_HTML5_Audio3
 - Pitch detector