

HTML 5 API'S

O que é uma API

Aplication Programming Interface



**“User Interfaces” permitem aos
utilizadores interagirem com
“Software”(programas)**



API's permitem a programas (& scripts) a
comunicarem entre si



**Enquanto developer's, não precisamos de
perceber como determinada aplicação ou
programa faz alguma coisa ...**



... Desde que saibamos como “**pedir**” a
esse programa para fazer “algo” e
processar a sua resposta



Uma **API**, diz-nos como utilizar um determinado programa, assim como o formato que poderemos esperar na resposta desse pedido



HTML5 & JS API's

A evolução dos devices vieram permitir aos browser's aceder a:

Funcionalidades do hardware
(GPS, acelerómetro, câmaras, bateria, ...)

Novas funcionalidades de “software”
(storage, file API, canvas, ...)

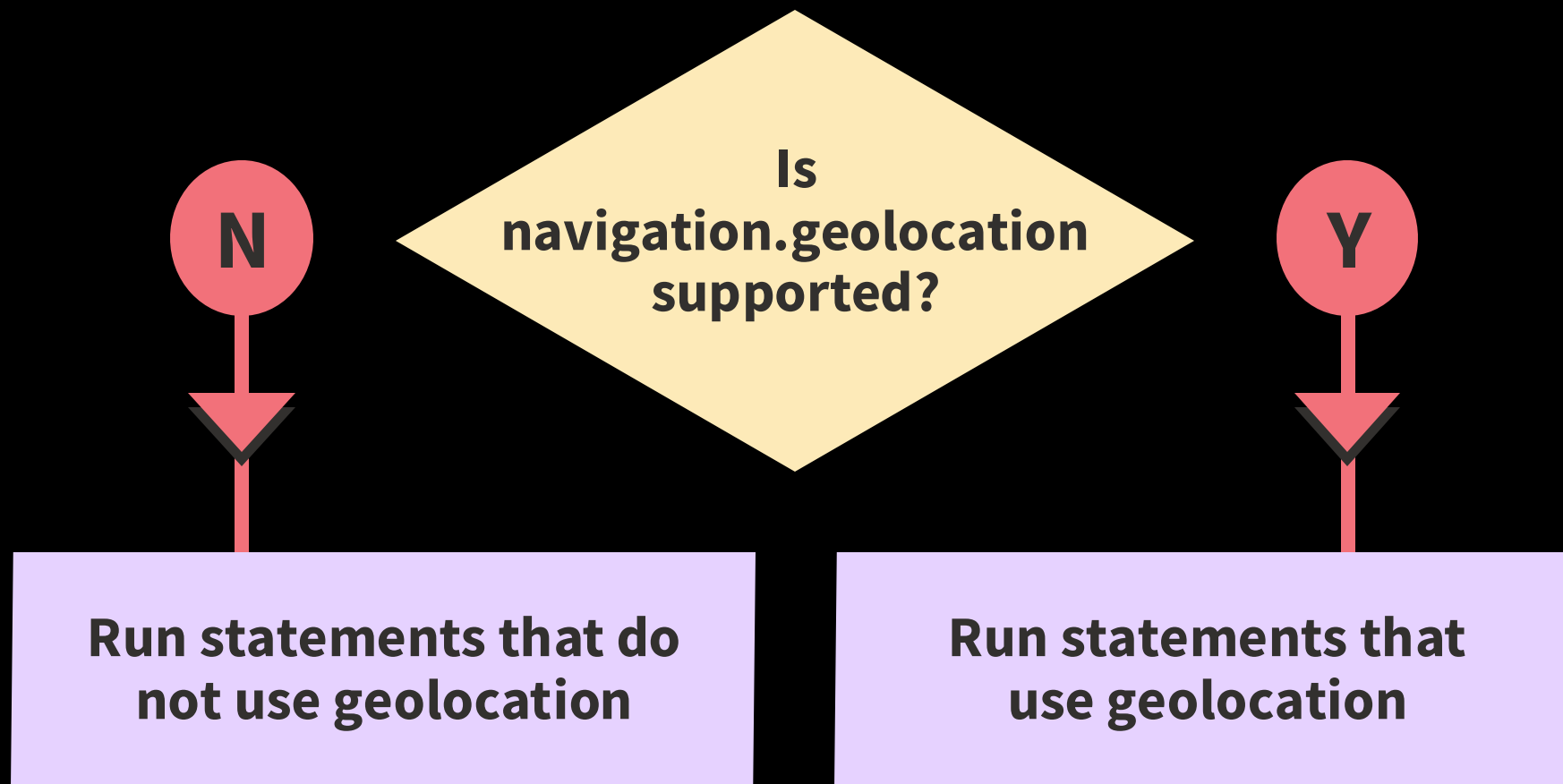


... about bringing the **right content** to the **right person** at the **right time**, in the **right context**, and making sure it's **accurate**, **up-to-date**, and **appropriate** through ongoing engagement



Feature Detection

Verificação se determinado “browser” suporta ou disponibiliza determinada funcionalidade





```
if (Modernizr.geolocation){  
    // executar qualquer tarefa  
} else {  
    // feature nao suportada..  
}
```



Exemplo...



Geolocation

Aceder após autorização á localização do utilizador.



```
function lerCoordenadas() {  
    if (Modernizr.geolocation) {  
        navigator.geolocation.getCurrentPosition(sucesso);  
    }  
}
```



Canvas

Desenho 2D e 3D no browser



```
let contexto = document.querySelector('canvas').getContext('2d');  
  
contexto.fillStyle='rgb(0,255,0)';  
contexto.fillRect(10,10,100,80);  
contexto.strokeStyle = 'rgb(0,100,0)';  
contexto.lineWidth = 5 ;  
contexto.strokeRect(10,10,100,80);
```



Page Visibility

Verificar se uma página ou separador do Browser estão em foco ou não



```
document.addEventListener('visibilitychange', pageChanged, false);

function pageChanged(){
    if(document.hidden){
        console.log('utilizador mudou de eseparador...')
    } else {
        console.log('voltou a minha pagina.... ')
    }
}
```



Online / Offline

Verificar se e tomar decisões se um determinado “device” perde ou readquire uma ligação á internet



```
window.addEventListener('offline', estado, false);
window.addEventListener('online', estado, false);

function estado(){
    if(navigator.onLine){
        console.log('esta ligado.....')
    } else {
        console.log('OFFFFFFFFF.... ')
    }
}
```



Drag & Drop

Permite a aplicação a possibilidade de arrastar e largar “objetos” numa aplicação web



```
items.addEventListener('dragstart', onDrag, false);
function onDrag(e){
    novoItem = e.target;
}

cesto.addEventListener('drop', onDrop, false);
function onDrop(e){
    cesto.innerHTML += ``;
    let numero = document.querySelectorAll('#cesto img').length;
    info.textContent = "Items: " + numero;
}
```



Web Storage

Permite aos browser's guardar dados no utilizador



```
let pessoaStr = JSON.stringify(pessoa);

window.localStorage.setItem('pessoa', pessoaStr);

let pessoaJSON = JSON.parse(window.localStorage.getItem('pessoa'));
console.log("Pessoa: ", pessoaJSON);
```



File API

Fornecer informação sobre ficheiros locais assim como permite aceder a determinados conteúdos



```
fileEl.addEventListener('change', onFileChange, false);
function onFileChange(e){

    let ficheiro = e.target.files[0];
    fileInfo.innerHTML = `
        <hr />
        Name: ${ficheiro.name} <br />
        Type: ${ficheiro.type} <br />
        Size: ${ficheiro.size} <br />
        <hr />
    `;
}
```



Fetch

Permite efetuar chamadas assíncronas via http, com a utilização de Promises



```
fetch('pessoa.json')  
  .then( response => response.json() )  
  .then( data => info.textContent = JSON.stringify(data) )  
  .catch(error => console.log('ERR0: ', error));
```



Demos ...