

# Review of the HTML5 API

Computer Systems Department, University of Castilla-La Mancha  
Albacete, Spain  
[felix.albertos@uclm.es](mailto:felix.albertos@uclm.es)



5th June 2018

# Index

📖 Introduction

📖 API

📖 Work in Progress

📖 Hands on

📖 Conclusions

# Index

 Introduction

 API

 Evolution

 Hands on

 Conclusions

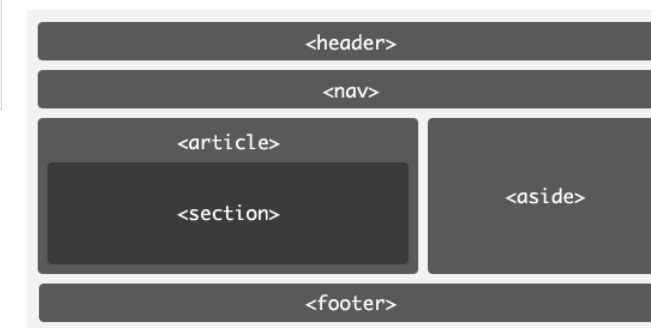
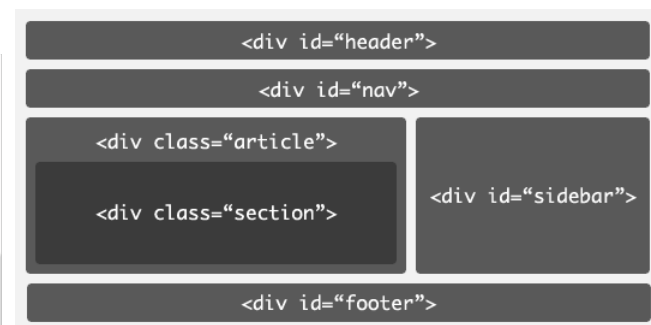
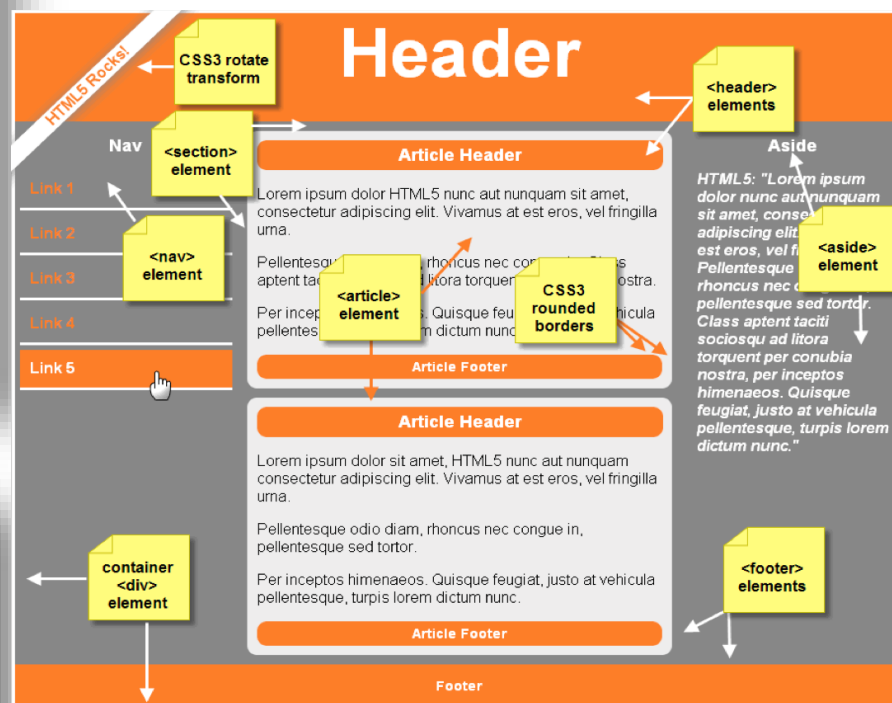
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
  <title>Mi Web en HTML4.01</title>
  <link type="text/css" rel="stylesheet" href="estilo.css">
  <script type="text/javascript">
    //JavaScript
  </script>

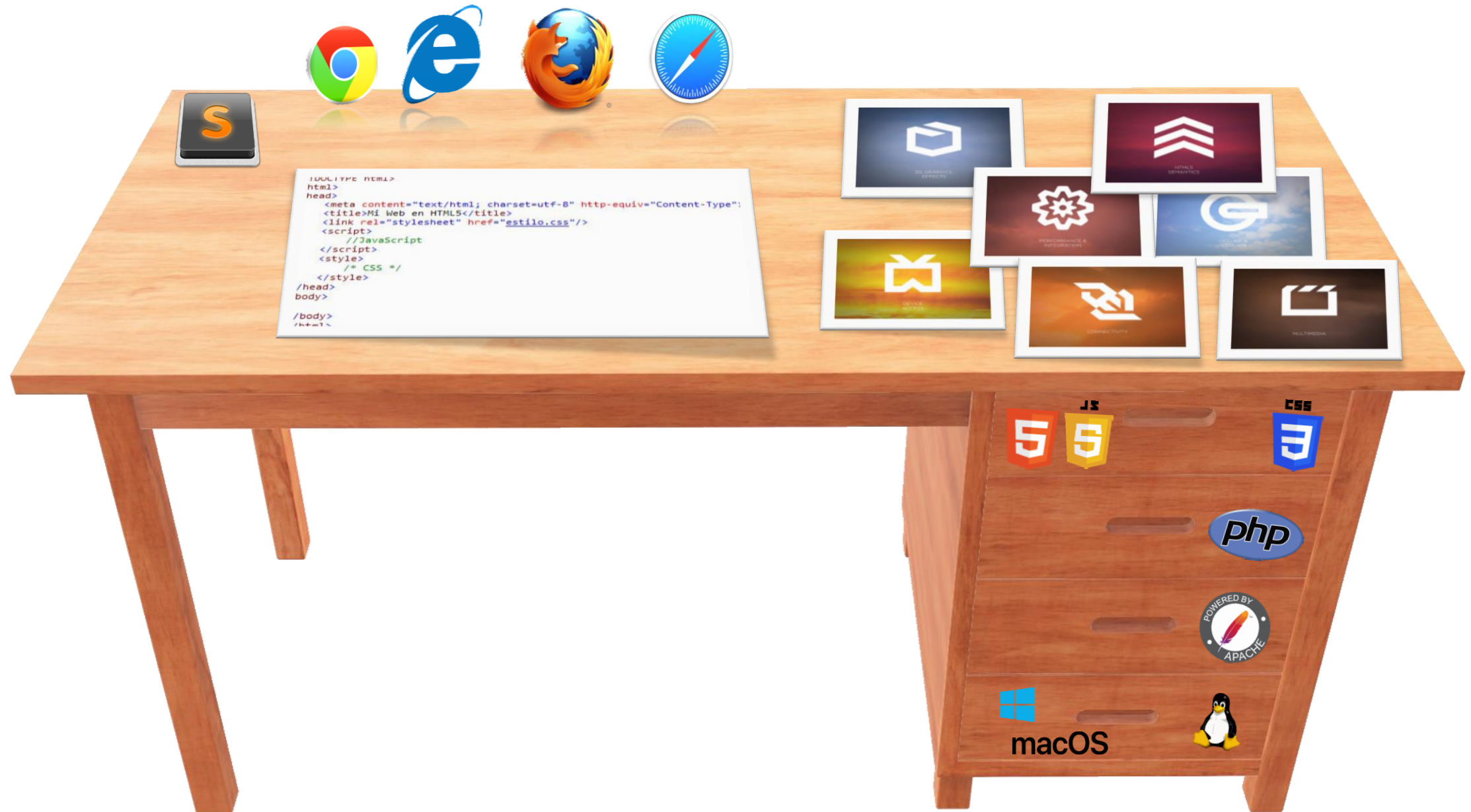
  <style type="text/css">
    /* CSS */
  </style>
</head>
<body>

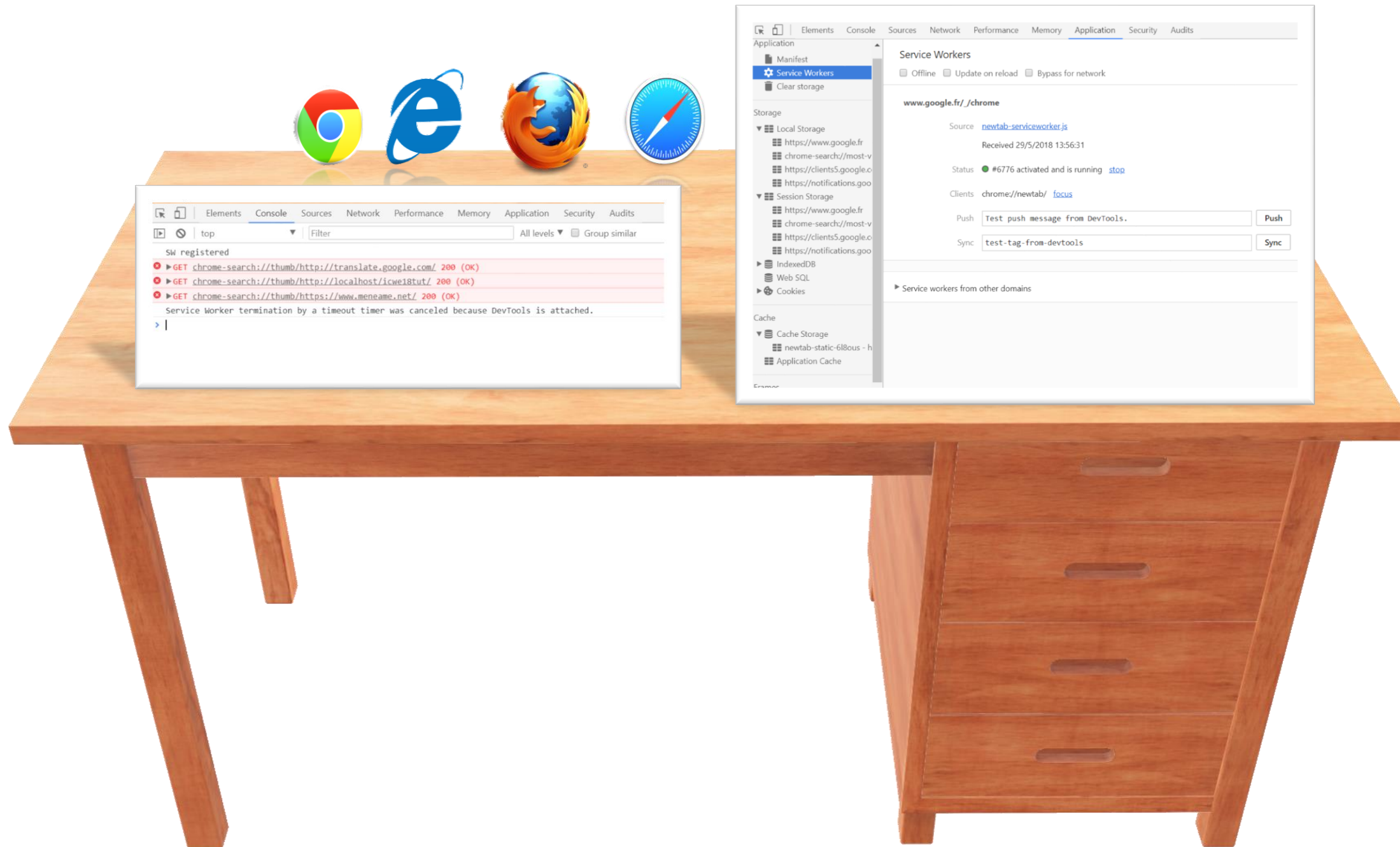
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
  <title>Mi Web en HTML5</title>
  <link rel="stylesheet" href="estilo.css"/>
  <script>
    //JavaScript
  </script>
  <style>
    /* CSS */
  </style>
</head>
<body>

</body>
</html>
```





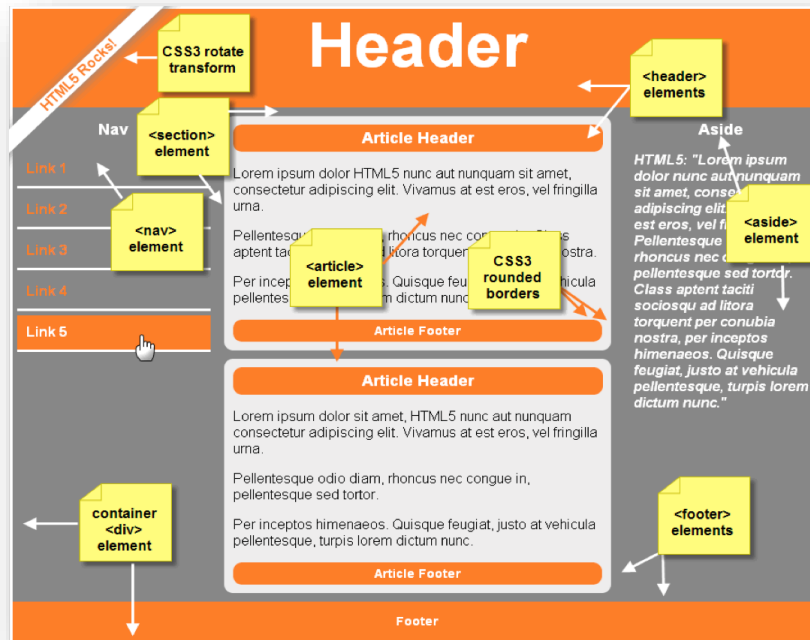






# Semantics

- Giving meaning to structure





## Offline & Storage

- App Cache
- Local Storage
- Session Storage
- Indexed DB
- File API
- Cache Storage



## Device Access

- Geolocation API
- Audio/Video input
- Contacts/Events
- Tilt Orientation



## Connectivity & Communication

- Server Sent Events
- Web Sockets
- WebRTC
- Cross-document messaging
- Channel messaging
- Broadcasting to other browsing contexts



# Multimedia

- Audio
- Video



## 3D Graphics, Effects

- SVG
- Canvas
- WebGL
- CSS3 3D
- Frame Animation



## Performance & Integration

- Web Workers
- XMLHttpRequest 2
- Service Workers

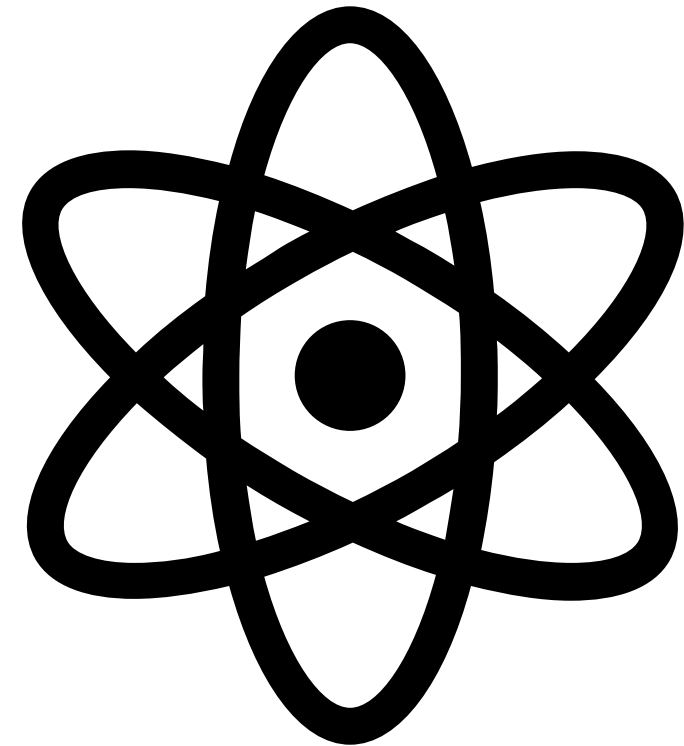




Web Page

Web Application

Native Web Application



# Index

 Introduction

 API

 **Work in Progress**

 Hands on

 Conclusions

# Can I use service workers

? ⚙ Settings

2 results found

## Service Workers 📄 - WD

Usage % of all users  
Global 79.31% + 0.19% = 79.49%

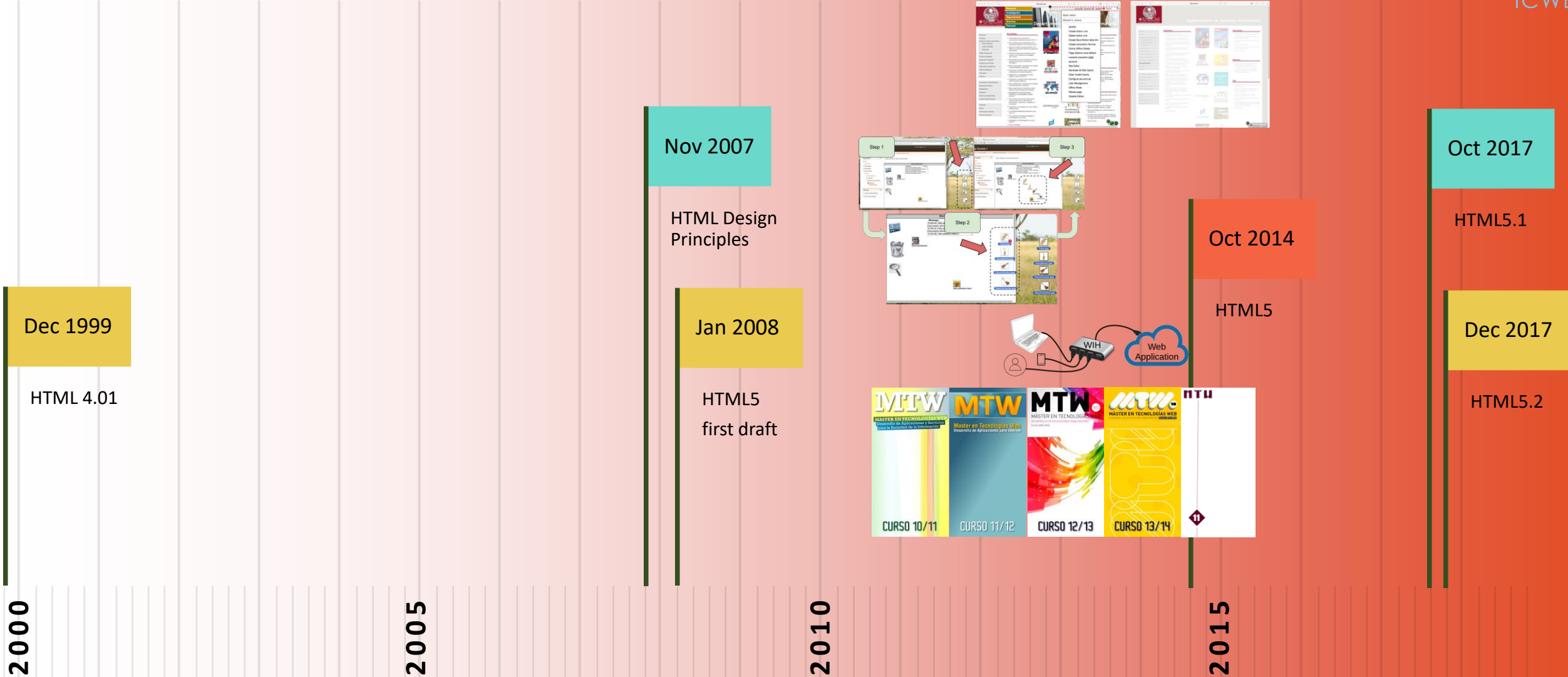
Method that enables applications to take advantage of persistent background processing, including hooks to enable bootstrapping of web applications while offline

# Can I use service workers

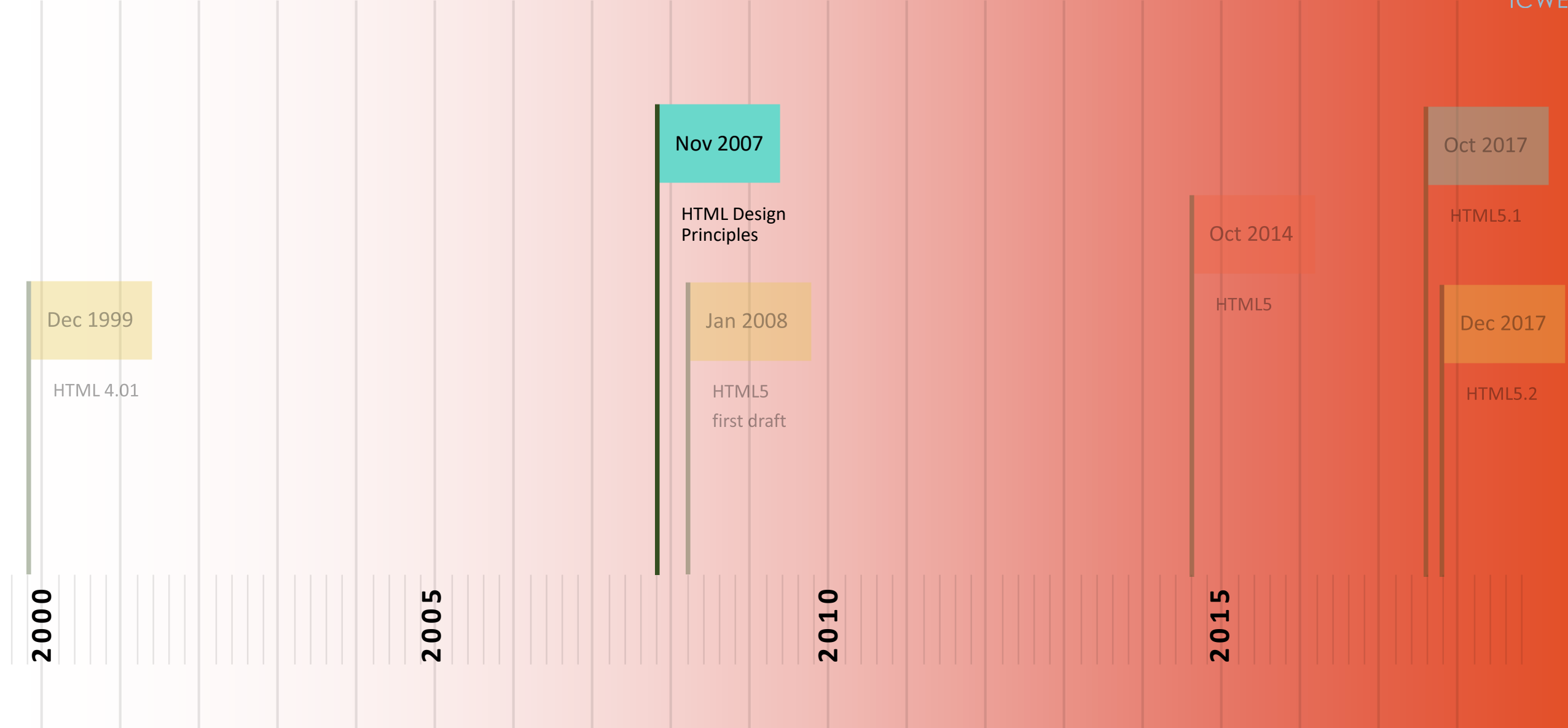
? ⚙ Settings

			49						
			64						
	216	59	65	11	10.3				
					11.2				4
11	17	60	66	11.1	11.3	all	66	11.8	6.2
	18	61	67	TP					
		62	68						
			69						

# Timeline



# Timeline



# Design Principles

- Compatibility
- Utility
- Interoperability
- Universal Access



## HTML Design Principles

W3C Working Draft 26 November 2007

**This Version:**

<http://www.w3.org/TR/2007/WD-html-design-principles-20071126/>



# Design Principles

## Compatibility Utility Interoperability Universal Access

- Support Existing Content
  - Is there a significant quantity of **existent content** on **popular websites intended for consumption** on the **public web** currently working as intended in **multiple popular user agents**?

List of websites

Site	Domain	Alexa top 50 global sites (As of May 16, 2018)
Google	google.com	1 ( )
YouTube	youtube.com	2 ( )
Facebook	facebook.com	3 ( )
Baidu	baidu.com	4 ( )

# Design Principles

**Compatibility** Utility Interoperability Universal Access

- Degrade Gracefully
  - Web content can degrade gracefully in older or less capable user agents, even when making use of new elements, attributes, APIs and content models.
  - Web browsers:
    - Current top mainstream
    - Highly popular old Web browsers
    - Top user agents designed to meet specific needs or address specialized markets

Graceful Degradation



# Design Principles

**Compatibility** Utility Interoperability Universal Access

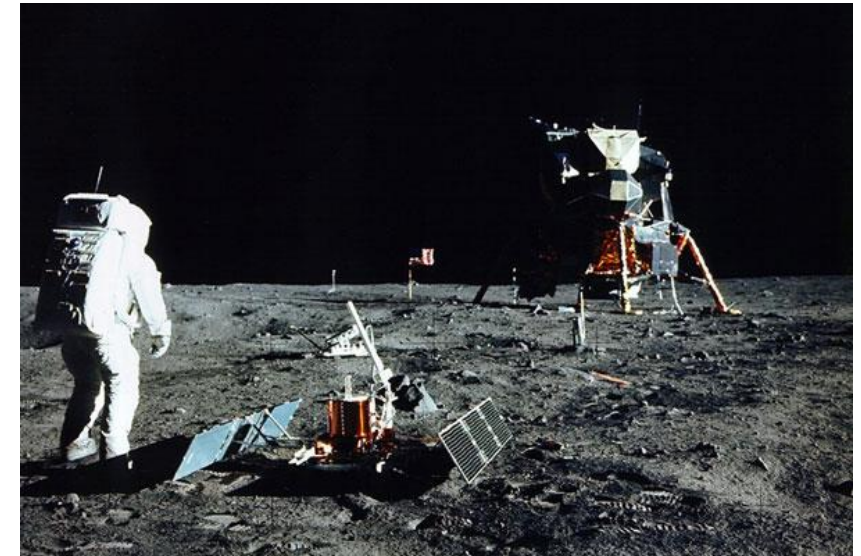
- Do not Reinvent the Wheel
- Pave the Cowpaths
- Evolution Not Revolution



# Design Principles

Compatibility **Utility** Interoperability Universal Access

- Solve Real Problems
- Priority of Constituencies
  - Users>authors>implementors>specifiers>theoretical purity
- Secure By Design
  - Communicating between documents from different sites is useful, but an unrestricted version could put user data at risk.



# Design Principles

Compatibility **Utility** Interoperability Universal Access

- Separation of Concerns
  - Content <--> Presentation
  - article element do not defines details of how it is displayed
  - b and I elements are widely used
- DOM Consistency



# Design Principles

Compatibility Utility **Interoperability** Universal Access

- Well-defined Behaviour
- Avoid Needless Complexity
- Handle Errors



THINGS TO DO TODAY	
Date _____	COMPLETED
1) _____	<input type="checkbox"/>
2) _____	<input type="checkbox"/>
3) _____	<input type="checkbox"/>
4) _____	<input type="checkbox"/>
5) _____	<input type="checkbox"/>
6) _____	<input type="checkbox"/>
7) _____	<input type="checkbox"/>
8) _____	<input type="checkbox"/>
9) _____	<input type="checkbox"/>
10) _____	<input type="checkbox"/>



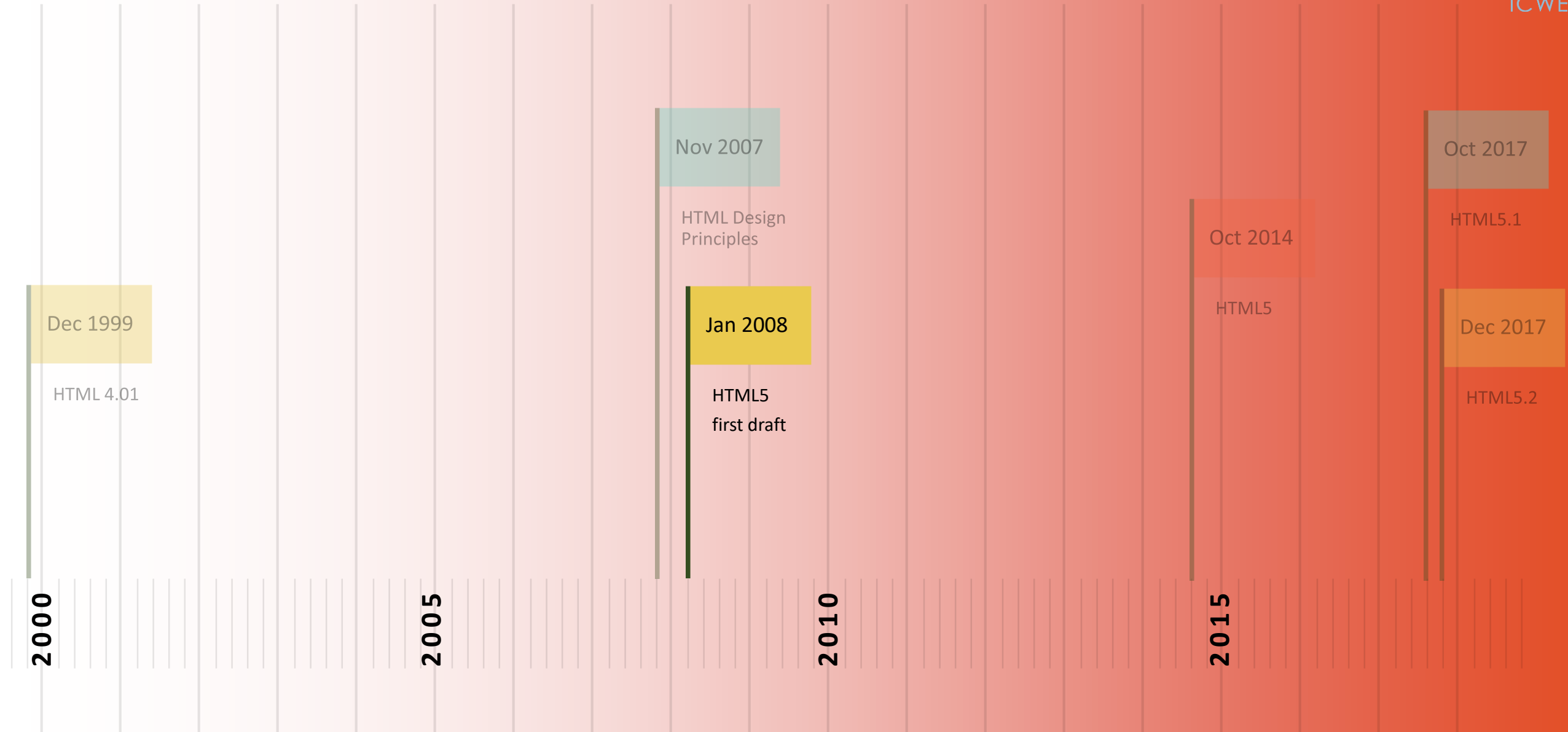
# Design Principles

## Compatibility Utility Interoperability **Universal Access**

- Media Independence
  - work across different platforms, devices and media
- Support World Languages
- Accessibility
  - Access by everyone regardless of ability is essential



# Timeline



# HTML 5 first working draft released



## W3C Publishes HTML 5 Draft, Future of Web Content

### Web Community Forges Next HTML Standard in Public W3C Forum

**Contact Americas, Australia --**

Ian Jacobs, <[ij@w3.org](mailto:ij@w3.org)>, +1.718.260.9447 or +1.617.253.2613

**Contact Europe, Africa and the Middle East --**

Marie-Claire Forgue, <[mcf@w3.org](mailto:mcf@w3.org)>, +33.492.38.75.94

**Contact Asia --**

Yasuyuki Hirakawa <[chibao@w3.org](mailto:chibao@w3.org)>, +81.466.49.1170

(also available in [French](#) and [Japanese](#); see also [translations in other languages](#))

---

[W3C Press Release Archive](#)

---

<http://www.w3.org/> -- 22 January 2008 -- W3C today published an early draft of [HTML 5](#), a major revision of the markup language for the Web. The [HTML Working Group](#) is creating HTML 5 to be the open, [royalty-free](#) specification for rich Web content and Web applications. The group operates entirely in public with nearly five hundred [participants](#), including representatives from W3C Members ACCESS, AOL, Apple, Google, IBM, Microsoft, Mozilla, Nokia, and Opera.

"HTML is of course a very important standard," said Tim Berners-Lee, author of the first version of HTML and W3C Director. "I am glad to see that the community of developers, including browser vendors, is working together to create the best possible path for the Web. To integrate the input of so many people is hard work, as is the challenge of balancing stability with innovation, pragmatism with idealism."

# The standard is in continuous development

- Added elements
- Removed elements
- Changed elements



## Obsolete

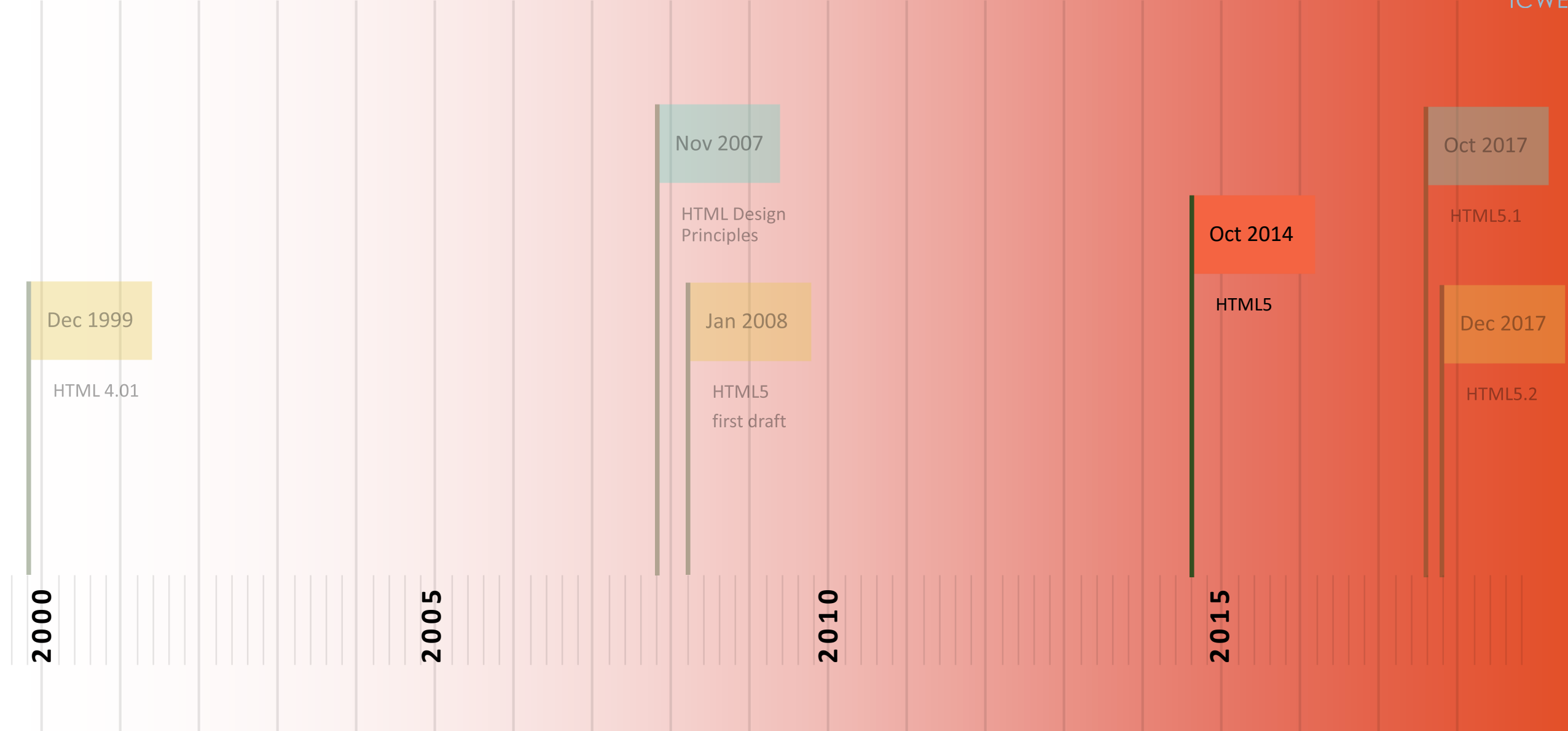
This feature is obsolete. Although it may still work in some browsers, its use is discouraged since it could be removed at any time. Try to avoid using it.



***This feature has been removed. Please fix your Web sites and applications.***

This method was removed in Chrome 43, and Firefox 56.

# Timeline



## Timeline



### HTML5

A vocabulary and associated APIs for HTML and XHTML

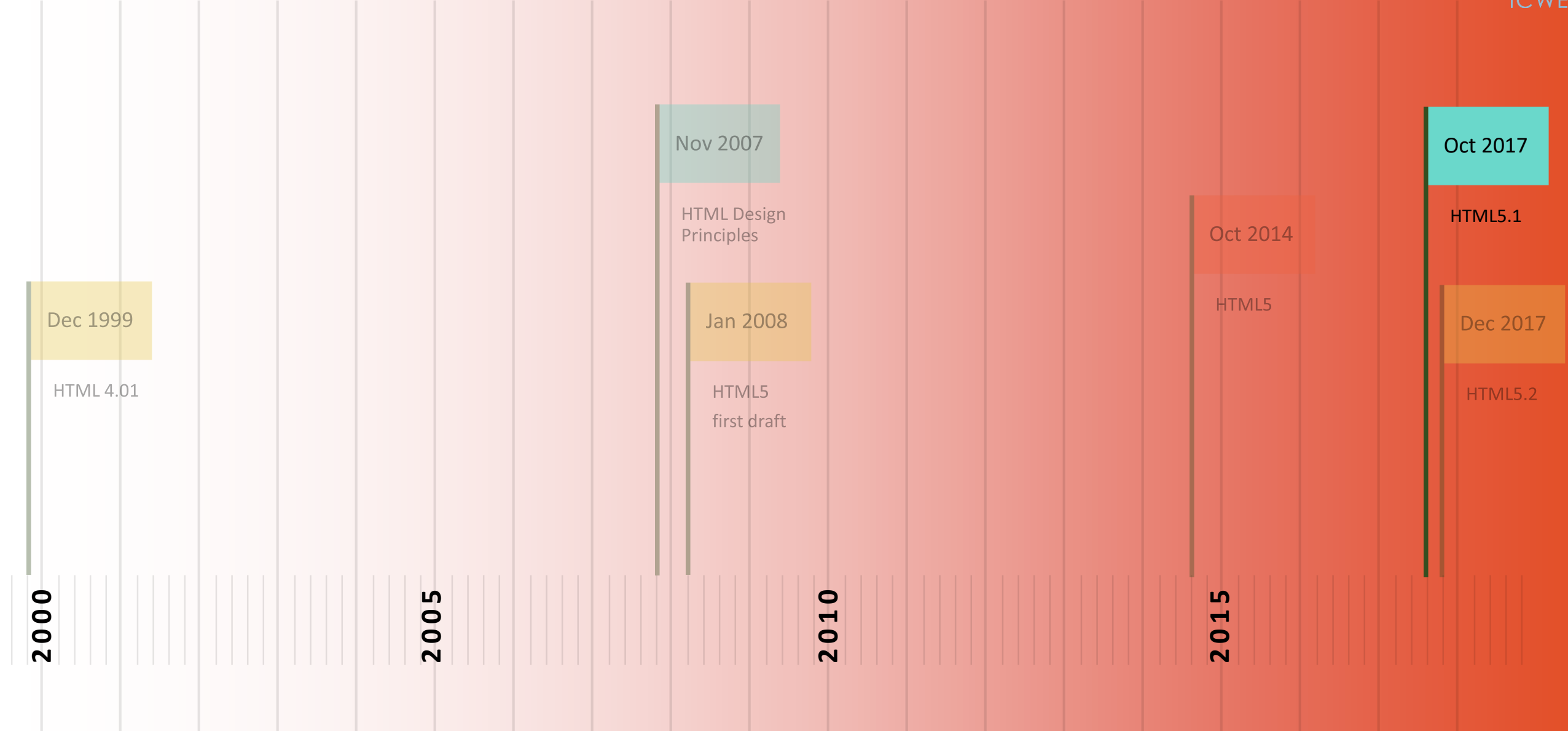
W3C Recommendation 28 October 2014

**This Version:**

<http://www.w3.org/TR/2014/REC-html5-20141028/>



# Timeline



# HTML 5.1

W3C Recommendation 3 October 2017 (2nd Edition)

## HTML 5.1 2nd Edition

W3C Recommendation 3 October 2017



**This version:**

<https://www.w3.org/TR/2017/REC-html51-20171003/>

# HTML 5.1

## Features added (some ...)

- **picture and srcset attributes allow responsive image selection**
- menuitem and type="context" attribute value enable authors to add functionality to the browser's context menu
- **requestAnimationFrame API allows for more efficient animation**
- enqueueJob and nextJob help explain Promise resolution in terms of microtasks
- HTMLMediaElement and srcObject objects
- event-source-error, event-track-error and event-track-load events for media fetching
- **history.scrollRestoration to control where a users' view is directed when navigating through their history**
- **add "noopener" to rel and window to allow for browsing contexts to be separated**
- **nonce attribute on script and style to support the use of Content Security Policy**

# HTML 5.1

## Features added (some ...)

**picture and srcset attributes allow responsive image selection**




- Srcset & sizes
  - Identical images
  - Larger or smaller depending on the device
  - For saving bandwidth
- To test:
  - Use incognito mode in chrome
  - Load with selected size and resize using mobile view
  - From the smallest width, increase width and notice image resolution changes
  - When page with more quality has been loaded, it always show it because it is intended for saving data

# HTML 5.1

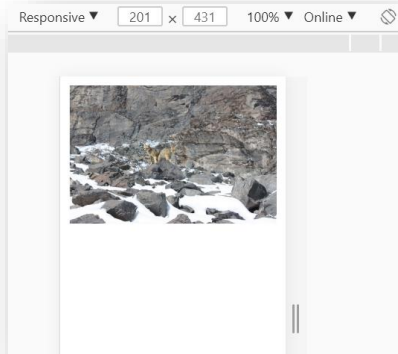
## Features added (some ...)

```

```

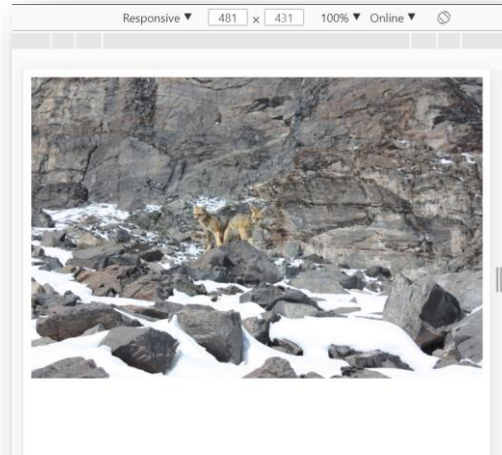
	fox480w.jpg	137 KB
	fox800w.jpg	296 KB
	foxfull.jpg	4.556 KB

picture and srcset attributes allow responsive image selection



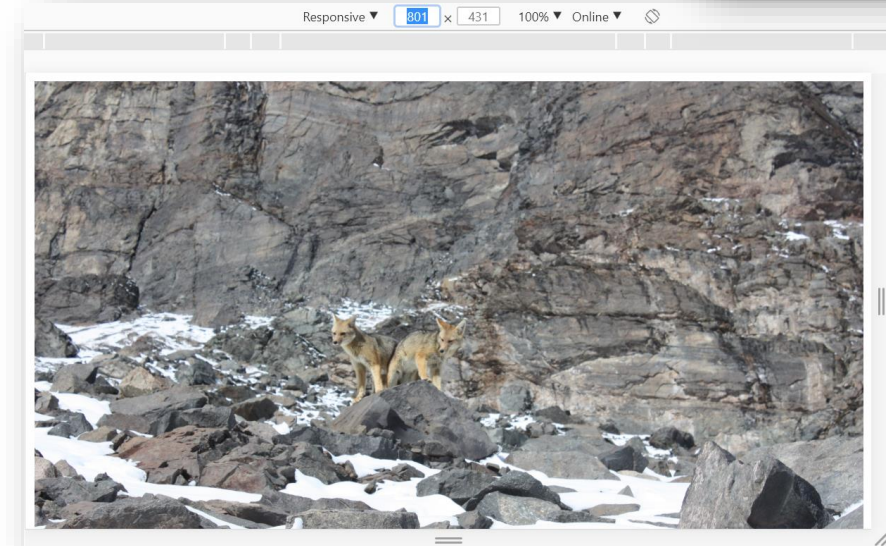
ResponsivImages\_srcset.html  
fox480w.jpg

25 Images  
3,43 MB



ResponsivImages\_srcset.html  
fox480w.jpg  
fox800w.jpg

25 Images  
7,4 MB



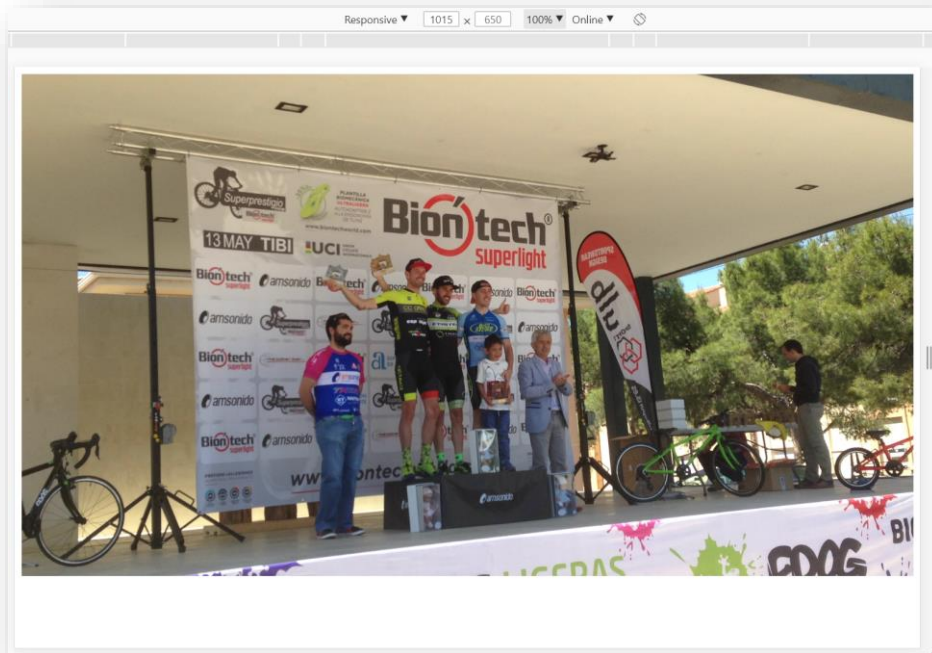
ResponsivImages\_srcset.html  
fox480w.jpg  
fox800w.jpg  
foxfull.jpg

25 Images  
113,9 MB

# HTML 5.1

## Features added (some ...)

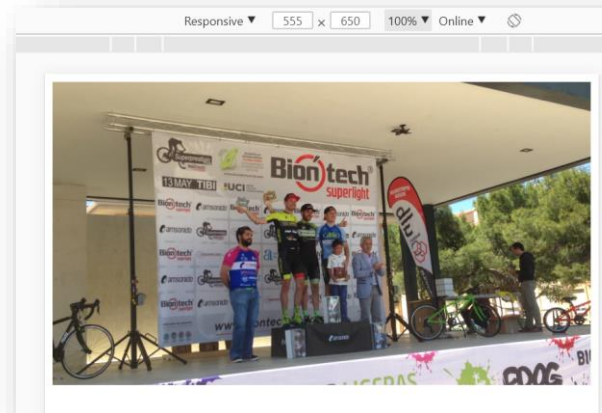
picture and srcset attributes allow responsive image selection



```

```

```
<picture>  
<source media="(max-width: 849px)" srcset="img/WinnerPodiumClose.jpg">  
<source media="(min-width: 850px)" srcset="img/WinnerPodium.jpg">  
  
</picture>
```



# HTML 5.1

## Features added (some ...)

### **requestAnimationFrame API allows for more efficient animation**

- Execution of code on the **next available screen repaint**
- Usually 60 times per second
  - Generally match the display refresh rate in most web browsers as per W3C recommendation
- `window.requestAnimationFrame(callback)`
  - callback takes one single argument which indicates the current time (in milliseconds)

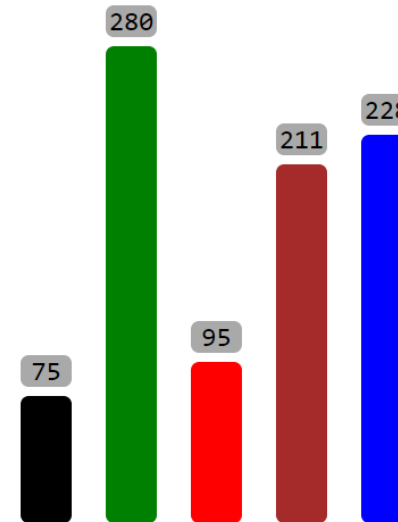
# HTML 5.1

## Features added (some ...)

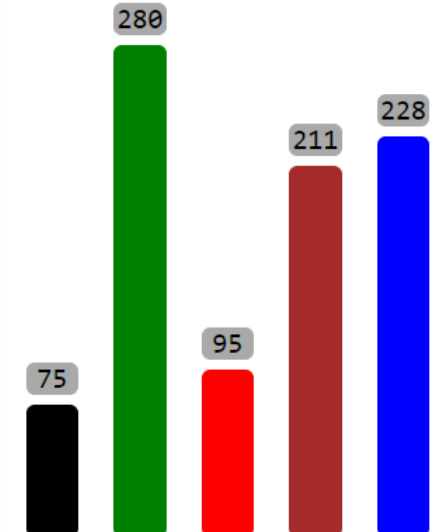
**requestAnimationFrame** API allows for more efficient animation

```
function step(timestamp) {  
  frames++;  
  if (!start){  
    start = timestamp;  
    lastTime = timestamp;  
  }else{  
    showFrameTime.innerHTML = timestamp-lastTime;  
    fps = (1000/(timestamp-lastTime));  
    lastTime = timestamp;  
    showFPS.innerHTML = fps;  
    showValue.innerHTML = frames;  
  }  
  graph.forEach(function(e){  
    e.setValue(frames);  
  });  
  if (frames<limit) {  
    window.requestAnimationFrame(step);  
  }  
}  
window.requestAnimationFrame(step);
```

Frame Time: 33.379000000000815  
FPS: 29.958956229964215  
Frame: 300



Frame Time: 16.668000000000575  
FPS: 59.995200383967216  
Frame: 300





# HTML 5.1

## Features added (some ...)

**history.scrollRestoration to control where a users' view is directed when navigating through their history**

- `history.scrollRestoration = 'auto';`
  - The browser restores the previous scroll position
- `history.scrollRestoration = 'manual';`
  - The scroll is not restored by the browser

# HTML 5.1

## Features added (some ...)

add "noopener" to rel and window to allow for browsing contexts to be separated

```
<p>Pick the yellow peach that looks like a sunset with its red, orange, and pink coat skin, peel it off with your teeth. Sink them into unripened... Please, <a href=
"..\\magazine\\index.html" rel="noopener" target="_blank" >visit my site</a> for more
information :)</p>
```

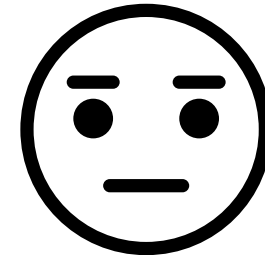


# HTML 5.1

## Features added (some ...)

add "noopener" to rel and window to allow for browsing contexts to be separated

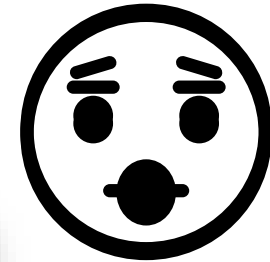
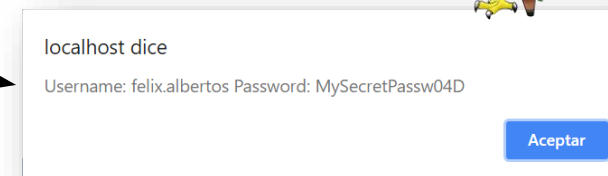
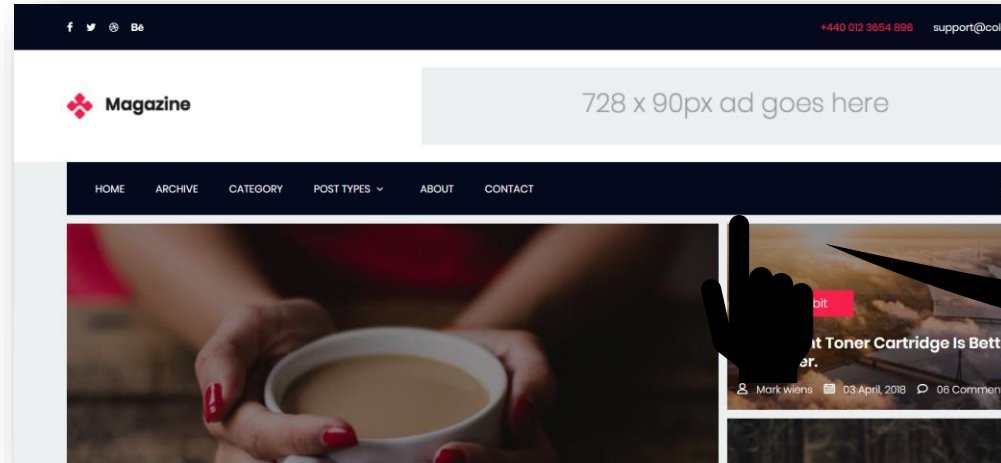
```
<p>Pick the yellow peach that looks like a sunset with its red, orange, and pink coat skin, peel it off with your teeth. Sink them into unripened... Please, <a href=
"..\\magazine\\index.html" target="_blank" >visit my site</a> for more information :)</p>
```



# HTML 5.1

## Features added (some ...)

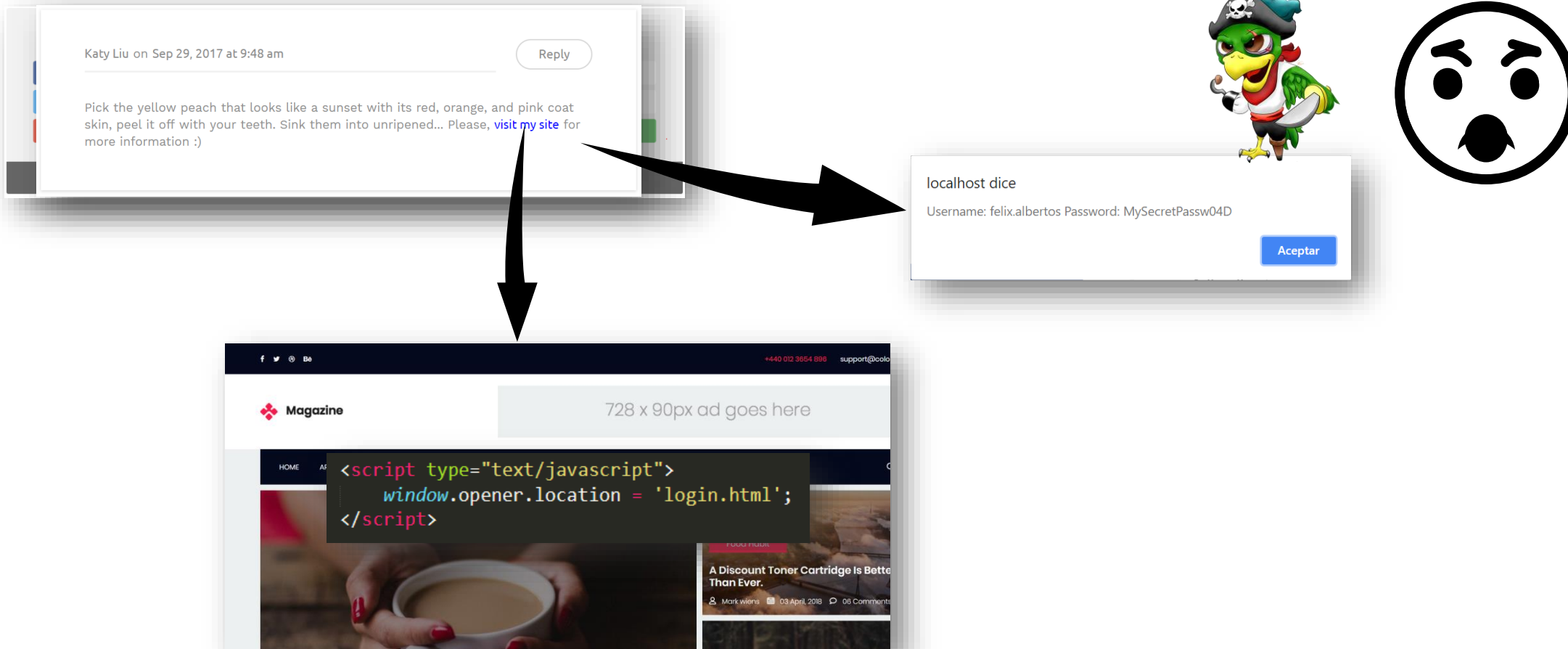
add "noopener" to rel and window to allow for browsing contexts to be separated



# HTML 5.1

## Features added (some ...)

add "noopener" to rel and window to allow for browsing contexts to be separated

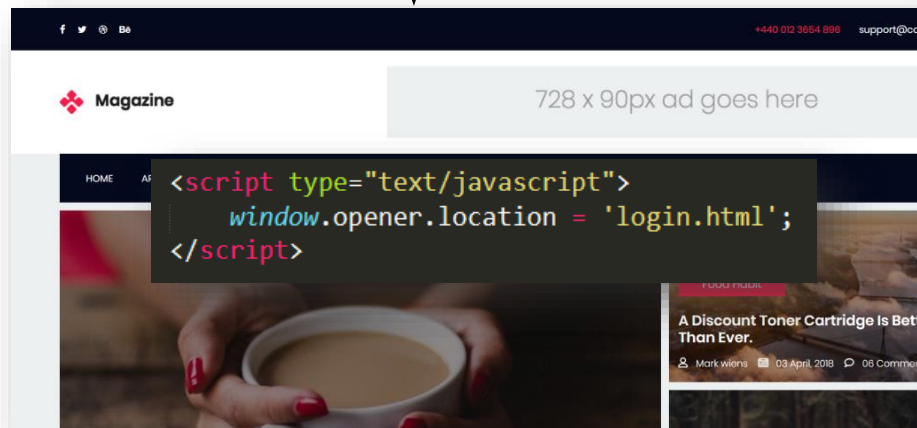
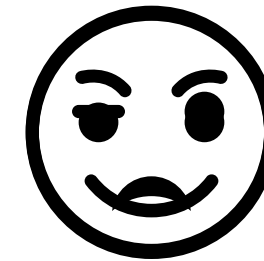
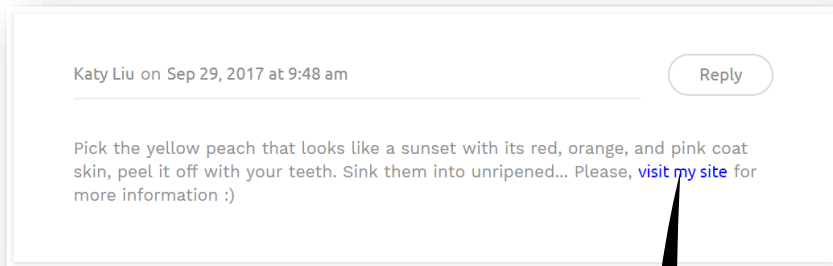


# HTML 5.1

## Features added (some ...)

```
<p>Pick the yellow peach that looks like a sunset with its red, orange, and pink coat skin, peel it off with your teeth. Sink them into unripened... Please, <a href=
"..\\magazine\\index.html" rel="noopener" target="_blank" >visit my site</a> for more
information :)</p>
```

add "noopener" to rel and window to allow for browsing contexts to be separated



# HTML 5.1

## Features added (some ...)

### **nonce attribute on script and style to support the use of Content Security Policy**

- What is Content Security Policy?
  - Security layer to detect and mitigate attacks
    - Cross Site Scripting (XSS)
    - Data Injection Attacks

# HTML 5.1

## Features added (some ...)

### **nonce attribute on script and style to support the use of Content Security Policy**

- How to enable it?
  - HTTP header
    - Content-Security-Policy: policy
  - or <meta> element
    - `<meta http-equiv="Content-Security-Policy" content="default-src 'self'; img-src https://*; child-src 'none';">`



# HTML 5.1

## Features added (some ...)

### nonce attribute on script and style to support the use of Content Security Policy

- script-src policy to allow sources
  - Content-Security-Policy: script-src <source> <source>;
- For example:
  - CSP header:
    - Content-Security-Policy: script-src https://www.felixalbertos.com/
  - Script
    - `<script src="http://www.felixalbertos.com/script/foobar/js/boot.js"></script>`



# HTML 5.1

## Features added (some ...)

### nonce attribute on script and style to support the use of Content Security Policy

- Among other values, source could be
  - nonce-<base64-value>
  - A whitelist for specific inline scripts using a cryptographic once (**number used once**)
    - Dynamically generated, for each requested page, if it is static ... :(
  - For example:
    - Content-Security-Policy: script-src 'nonce-kjl93204ids'
    - `<script nonce="kjl93204ids">`  
    // JavaScript code  
    </script>

# HTML 5.1

## Features removed (some ...)

- appCache
  - Offline Web Applications, almost ...
- Media Controllers
- command API
  - A command is the abstraction behind menu items, buttons, and links. Once a command is defined, other parts of the interface can refer to the same command, allowing many access points to a single feature
- usemap attribute on object

### Why?

- Sometimes nobody, or only one (Media Controllers ! ), has implemented it
- Implementation is inconsistent
- It doesn't work as expected

# HTML 5.1

## Features removed (some ...)

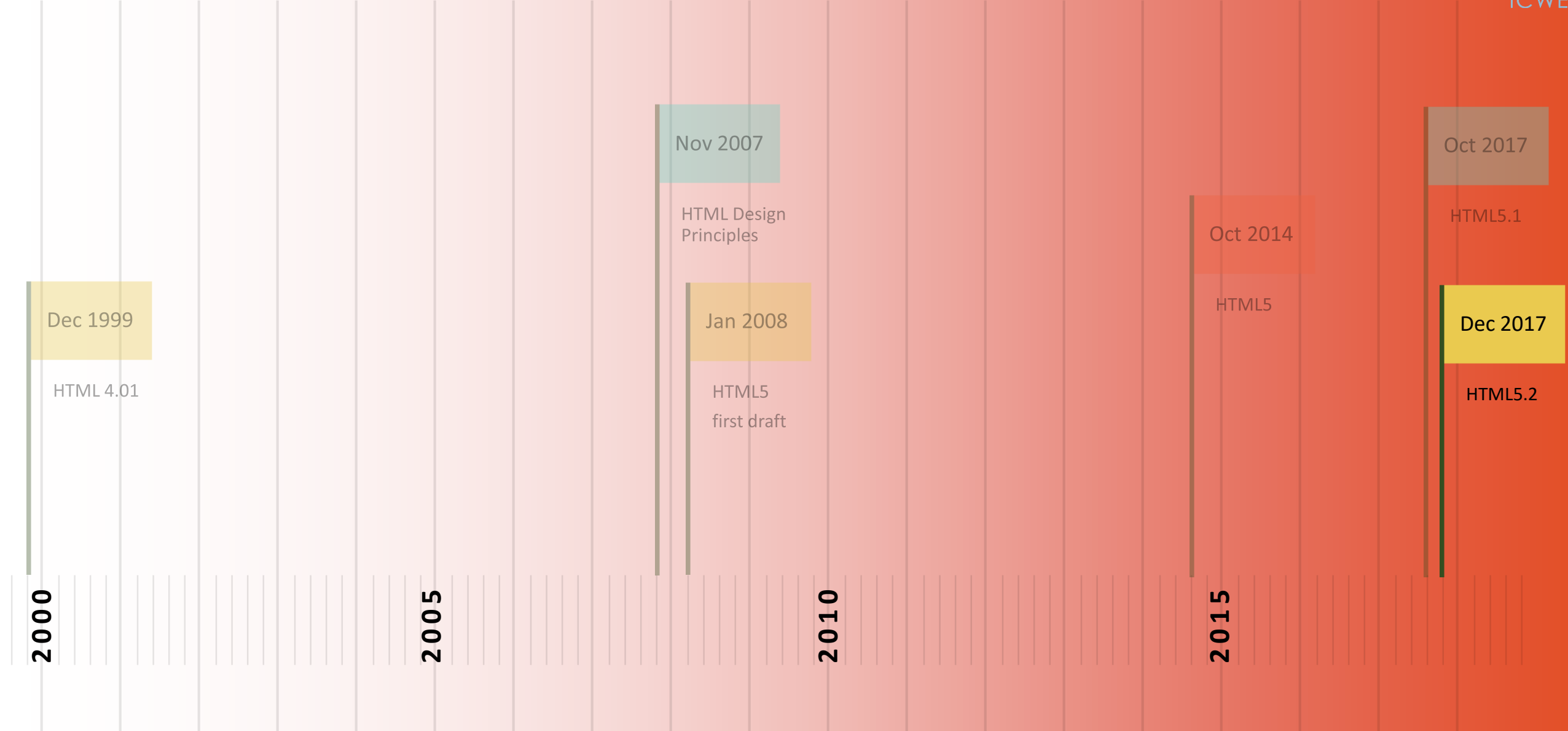
- appCache
  - Not able to decide if serving cached or remote content
  - Not having control of which files are being processed
  - If something fails, it totally breaks

# HTML 5.1

## Features changed (some ...)

- `accesskey` takes a single character as a value (as in HTML 4)
  - accessibility
- header and footer elements can be nested, if each level is within a sectioning element
- mousewheel event is called `wheel`
- `img` and related elements support `width="0"`
- `.tfoot` and `.createTfoot()` always insert at the end of a table

# Timeline



## HTML 5.2

W3C Recommendation 14 December 2017

# HTML 5.2

W3C Recommendation, 14 December 2017



**This version:**

<https://www.w3.org/TR/2017/REC-html52-20171214/>

## HTML 5.2

### Features added (some ...)

- **<dialog> element**
- Integration with the JavaScript module system of [[ECMA-262](#)]
- Update ARIA reference to [[wai-aria-1.1](#)], which introduces new features to improve accessibility
- **allowpaymentrequest attribute of iframe, for integration with the Payment Request API [[PAYMENT-REQUEST](#)]**
- **allow-presentation value for the sandbox attribute of iframe, for integration with the Presentation API [[PRESENTATION-API](#)]**
- Define the about:html-kind URL for MP4 media track integration



## HTML 5.2

### Features added (some ...)

#### <dialog> element

- The dialog element represents a part of an application that a user interacts with to perform a task, for example a dialog box, inspector, or window
- The open attribute indicates that the dialog is active and available for interaction. When the open attribute is not set, it shouldn't be shown to the user
  - The method `show()` / `showModal()` opens the dialog
- Closing the dialog
  - method `close([return_value])`
  - Property `returnValue` of the dialog receives the value in `return_value`
- When it is used with integrated forms, the submit button closes the dialog

# HTML 5.2

## Features added (some ...)

### <dialog> element

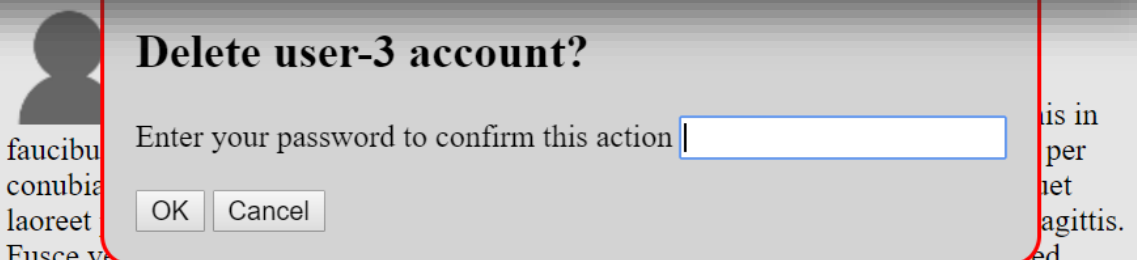
```
myDialog = document.getElementById("confirmDeleteAccount");
userNameDialog = document.getElementById("userNameDialog");
myDialog.addEventListener('close', dialogClosedEvent);
```

```
function openDialog(id, modal){
  userID = id;
  userNameDialog.innerHTML=id;
  document.getElementById("password").value="";
  if(modal){
    myDialog.showModal();
  }else{
    myDialog.show();
  }
}
```

```
function closeDialog(v){
  if(v){
    if(document.getElementById("password").value==pwd){
      var elem = document.getElementById(userID);
      elem.parentNode.removeChild(elem);
      myDialog.close();
    }
  }else{
    myDialog.close();
  }
}
```

condimentum ante ultricies aliquam vel vel metus. Duis tincidunt risus eu leo congue, a

```
<dialog id="confirmDeleteAccount">
  <h2>Delete <span id="userNameDialog"></span> account?</h2>
  <p>
    Enter your password to confirm this action <input type="password" id="password">
  </p>
  <button onclick="closeDialog(true);">OK</button>
  <button onclick="closeDialog(false);">Cancel</button>
</dialog>
```



```
<button onclick="openDialog('user-1',true);">Delete account</button>
```

# HTML 5.2

## Features added (some ...)

- Integration with the JavaScript module system of [\[ECMA-262\]](#)

### JavaScript

Some parts of the language described by this specification only support JavaScript as the underlying scripting language. [\[ECMA-262\]](#)

#### NOTE:

The term "JavaScript" is used to refer to ECMA262, rather than the official term ECMAScript, since the term JavaScript is more widely known. Similarly, the [MIME type](#) used to refer to JavaScript in this specification is `text/javascript`, since that is the most commonly used type, [despite it being an officially obsoleted type](#) according to RFC 4329. [\[RFC4329\]](#)

## HTML 5.2

### Features added (some ...)

- **allowpaymentrequest attribute of iframe, for integration with the Payment Request API** [[PAYMENT-REQUEST](#)]
  - To indicate that a cross-origin iframe is allowed to invoke the payment request API, the allowpaymentrequest attribute can be specified on the iframe element.
- Payment-Request
  - This specification standardizes an API to allow merchants (i.e. web sites selling physical or digital goods) to utilize one or more payment methods with minimal integration. User agents (e.g., browsers) facilitate the payment flow between merchant and user.

# HTML 5.2

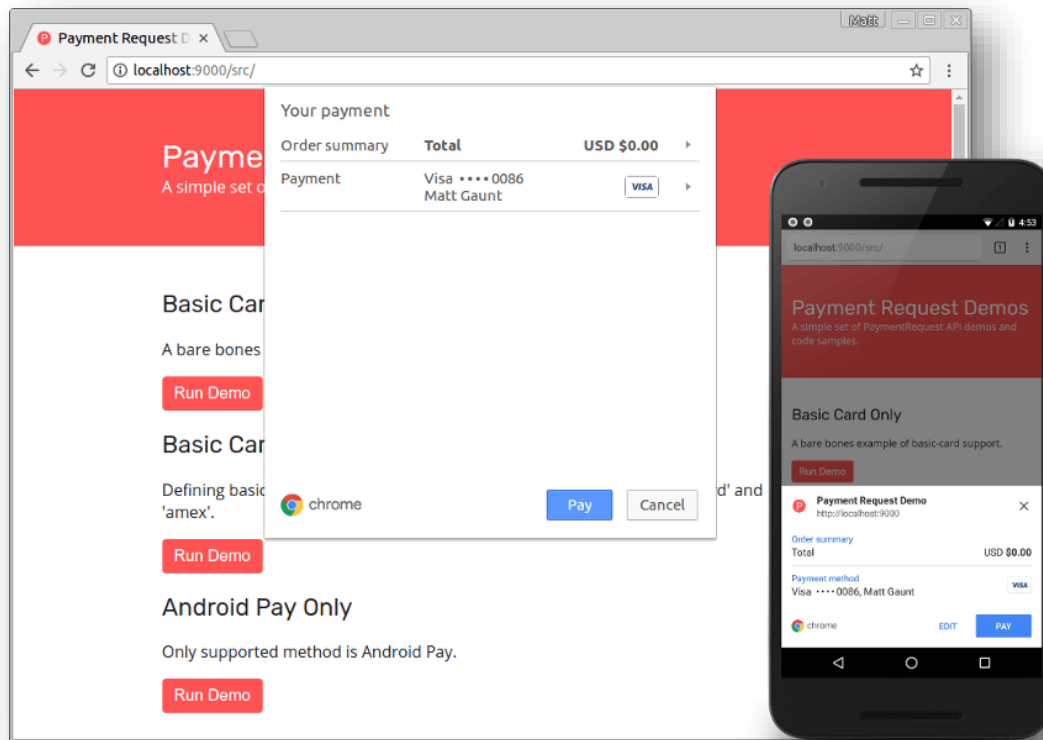
## Features added (some ...)

- **allowpaymentrequest** attribute of **iframe**, for integration with the Payment Request API [[PAYMENT-REQUEST](#)]
- Payment-Request
  - Fast purchase experience
  - Consistent experience on every site (that supports the API)
  - Accessibility
  - Credentials management
  - Consistent error handling

# HTML 5.2

## Features added (some ...)

- `allowpaymentrequest` attribute of `iframe`, for integration with the Payment Request API [[PAYMENT-REQUEST](#)]
- Payment-Request



Source: <https://developers.google.com/web/fundamentals/payments/deep-dive-into-payment-request>

## HTML 5.2

### Features added (some ...)

- **allow-presentation** value for the **sandbox** attribute of **iframe**, for integration with the **Presentation API** [[PRESENTATION-API](#)]
  - Allows embedders to have control over whether an **iframe** can start a presentation session
- **Presentation API**
  - Enable Web content to access presentation displays and use them for presenting Web content
  - Supported types of multimedia devices include both displays which are wired using HDMI, DVI, or the like, or wireless, using DLNA, Chromecast, AirPlay, or Miracast.



## HTML 5.2

### Features added (some ...)

- allow-presentation value for the sandbox attribute of iframe, for integration with the Presentation API  
[[PRESENTATION-API](#)]
- Presentation API





# HTML 5.2

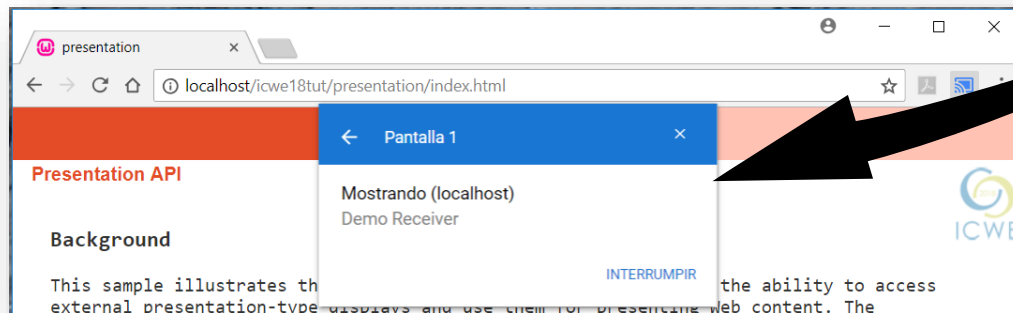
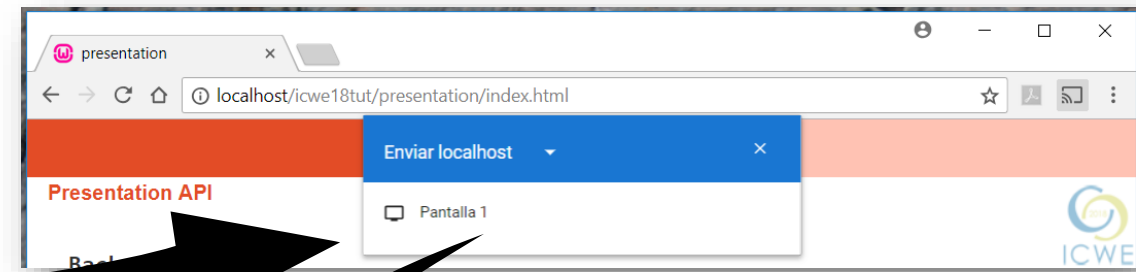
## Features added (some ...)

- allow-presentation value for the sandbox attribute of iframe, for integration with the Presentation API [\[PRESENTATION-API\]](#)
- Presentation API

```
const presentationRequest = new PresentationRequest(['receiver/index.html']);

// Make this presentation the default one when using the "Cast" browser menu.
navigator.presentation.defaultRequest = presentationRequest;
```

```
document.querySelector('#start').addEventListener('click', function() {
  log('Starting presentation request...');
  presentationRequest.start();
});
```



```
presentationRequest.addEventListener('connectionavailable', function(event) {
  presentationConnection = event.connection;
  presentationConnection.addEventListener('close', function() {
    log('> Connection closed.');
```

Source: <https://googlechrome.github.io/samples/presentation-api/index.html>

# HTML 5.2

## Features added (some ...)

- allow-presentation value for the sandbox attribute of iframe, for integration with the Presentation API  
[[PRESENTATION-API](#)]
- Presentation API

```
document.querySelector('#sendMessage').addEventListener('click', function() {  
  const message = document.querySelector('#message').value.trim();  
  const lang = document.body.lang || 'en-US';  
  
  log('Sending "' + message + '"...');  
  presentationConnection.send(JSON.stringify({message, lang}));  
});
```

```
document.addEventListener('DOMContentLoaded', function() {  
  if (navigator.presentation.receiver) {  
    navigator.presentation.receiver.connectionList.then(list => {  
      list.connections.map(connection => addConnection(connection));  
      list.addEventListener('connectionavailable', function(event) {  
        addConnection(event.connection);  
      });  
    });  
  }  
});
```

```
connection.addEventListener('message', function(event) {  
  messageIdx++;  
  const data = JSON.parse(event.data);  
  const logString = 'Message ' + messageIdx + ' from connection #' +  
    connection.connectionId + ': ' + data.message;  
  addMessage(logString, data.lang);  
  maybeSetFruit(data.message);  
  connection.send('Received message ' + messageIdx);  
});
```

Source: <https://googlechrome.github.io/samples/presentation-api/index.html>

## HTML 5.2

### Features removed (some ...)

- keygen, menu and menuitem (added in 5.1) elements
- inputmode attribute for textual input elements, and the dropzone attributes
- showModalDialog method

## HTML 5.2

### Constructions now valid HTML

- `<style>` within the `<body>`
- multiple `<main>` elements in the DOM, so long as only one is visible to the user
  - The main content area of a document includes content that is unique to that document and excludes content that is repeated across a set of documents such as site navigation links, copyright information, site logos and banners and search forms (unless the document or application's main function is that of a search form).

why? Accessibility ...

# Index

 Introduction

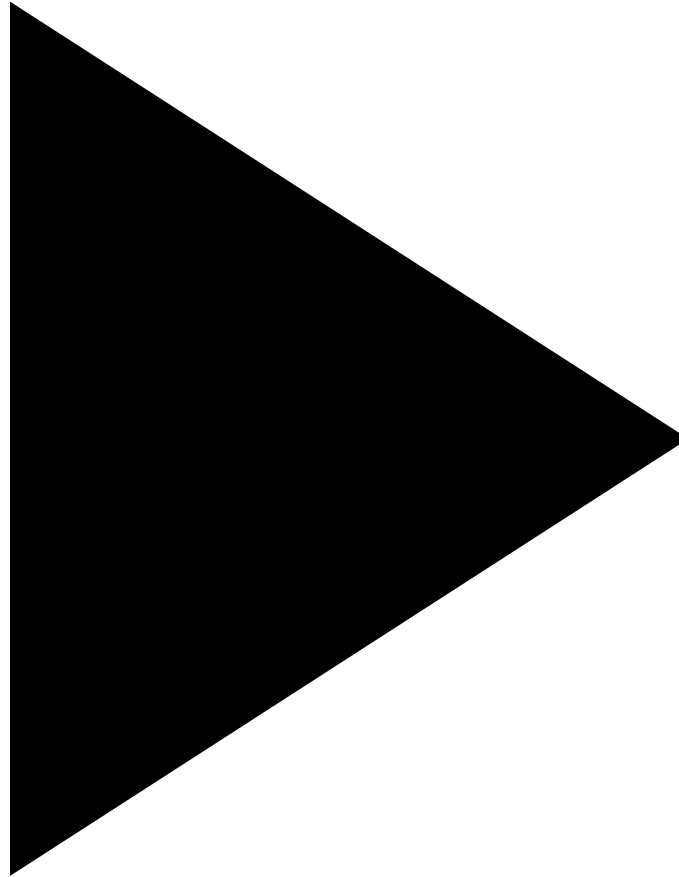
 API

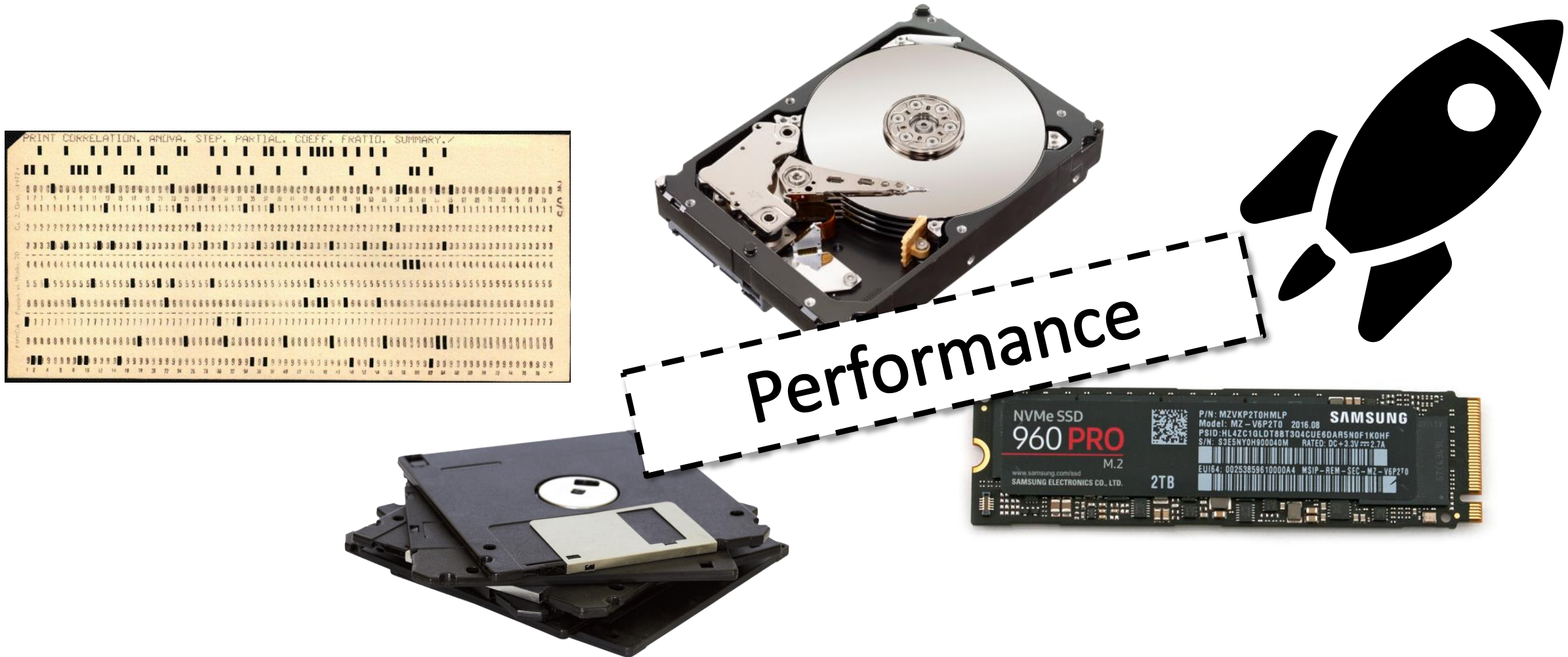
 Evolution

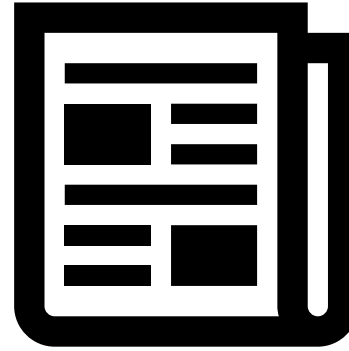
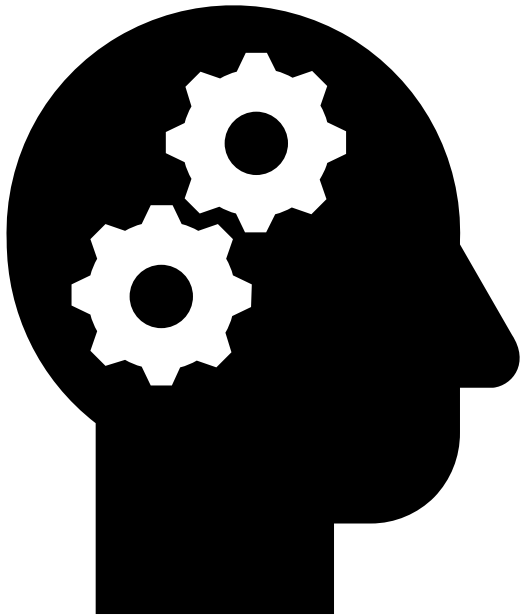
 Hands on

 Conclusions

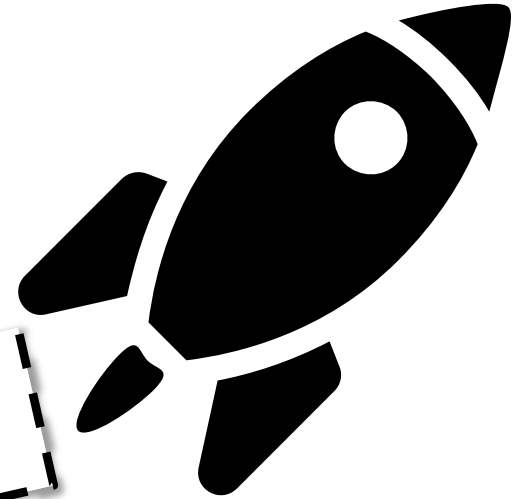
## Canvas, Drag Files, Local Storage







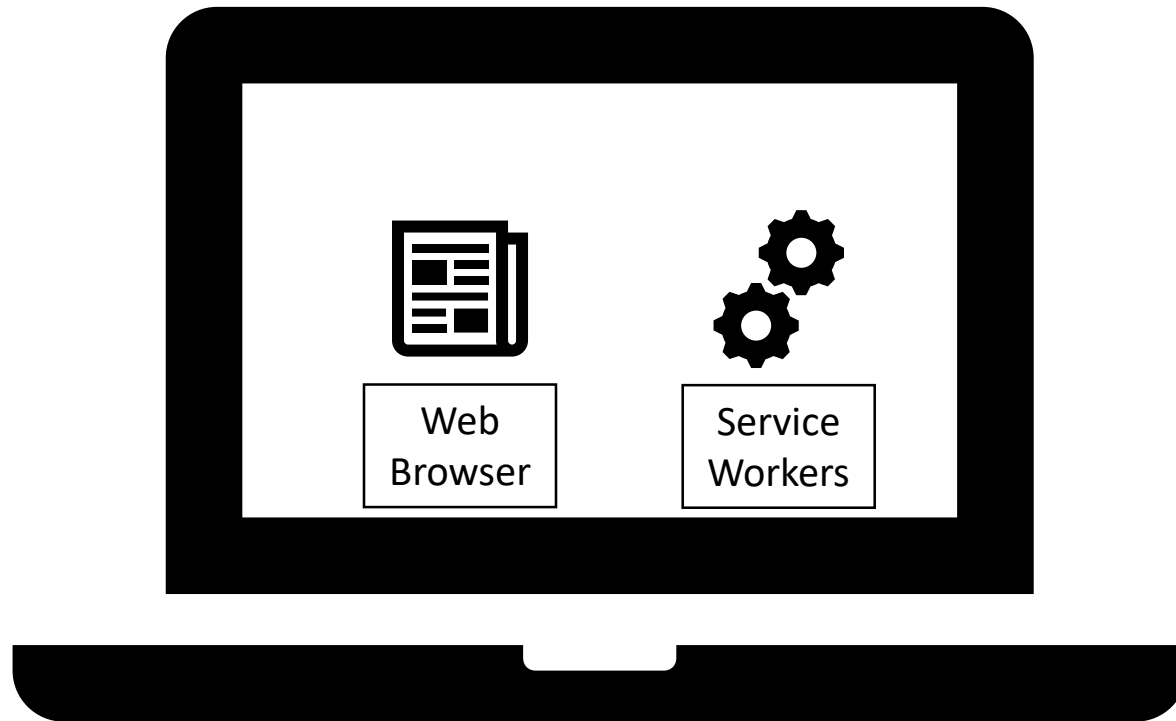
**Taking the Control**





## What

- Proxy servers that sit between web applications, the browser, and the network



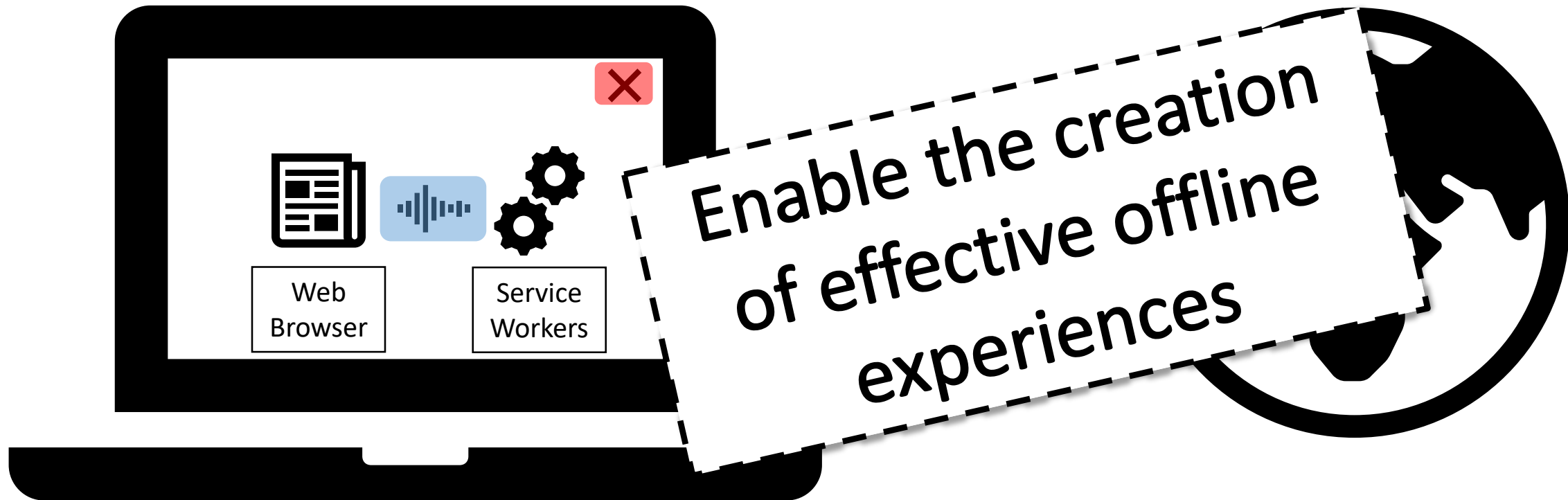
## How

- Intercept network requests and take appropriate action based on whether the network is **available**, updating assets residing on the server\*

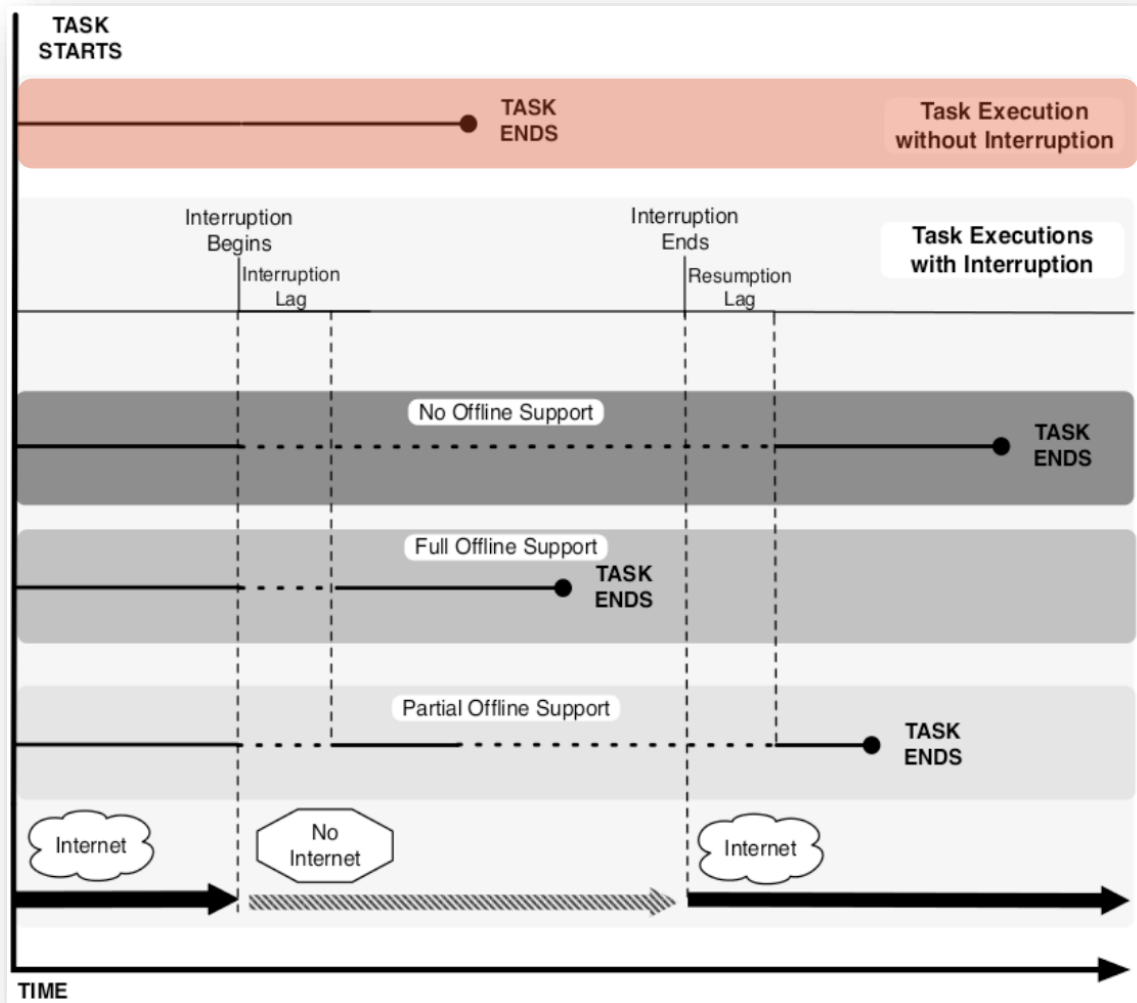


## How

- Intercept network requests and take appropriate action based on whether the network is available **or not**, assets locally cached can be used



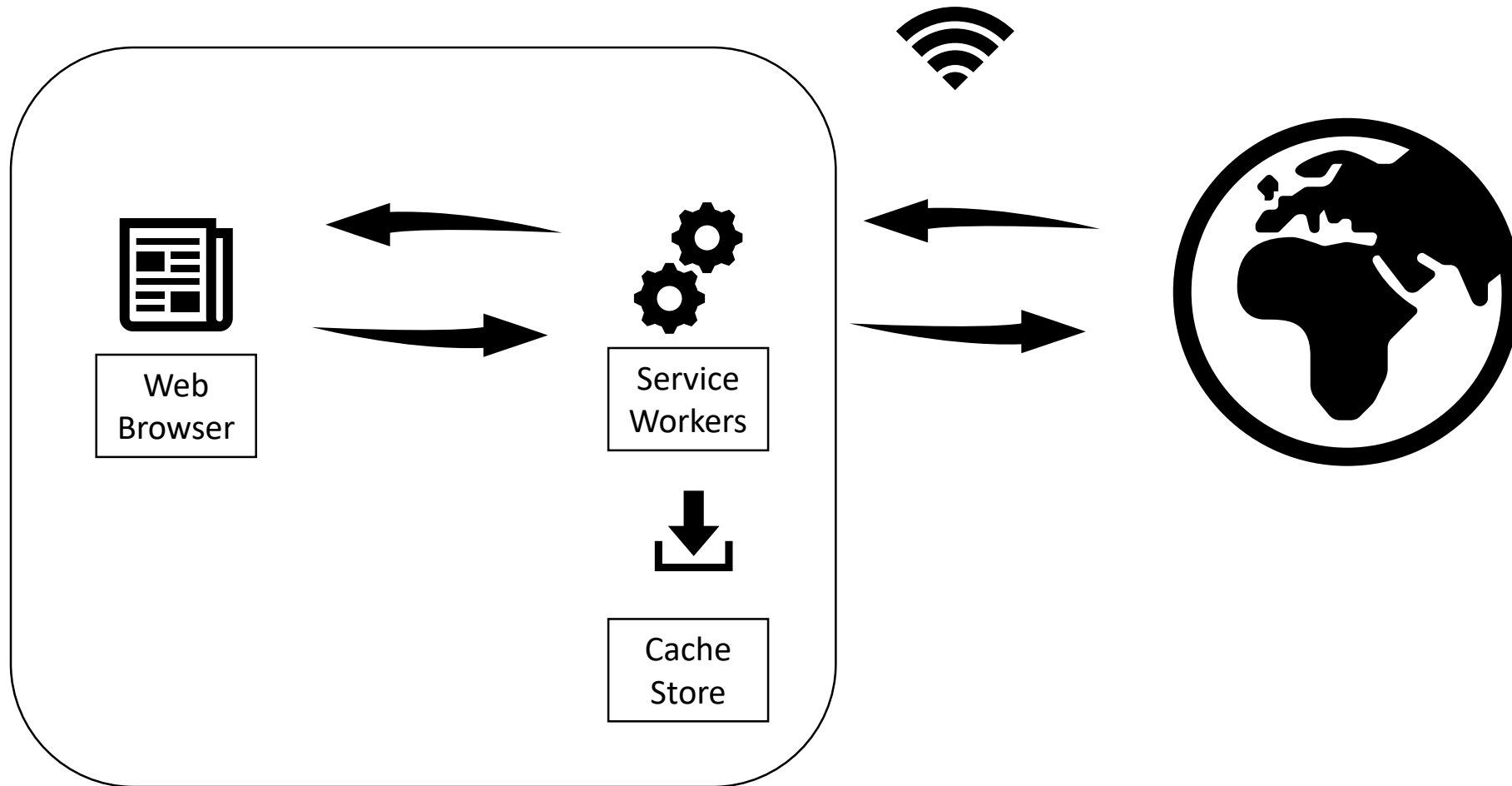
# How



Enable the creation  
of effective offline  
experiences

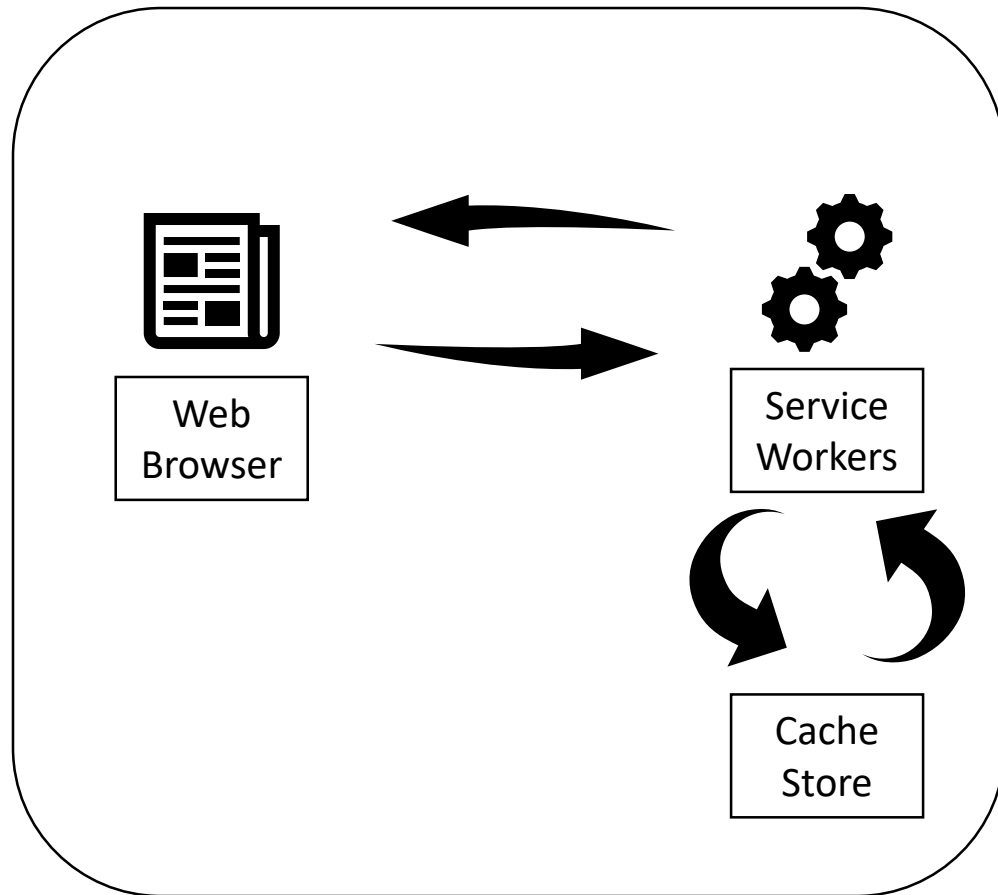
## Scenarios

- Updating assets from the server



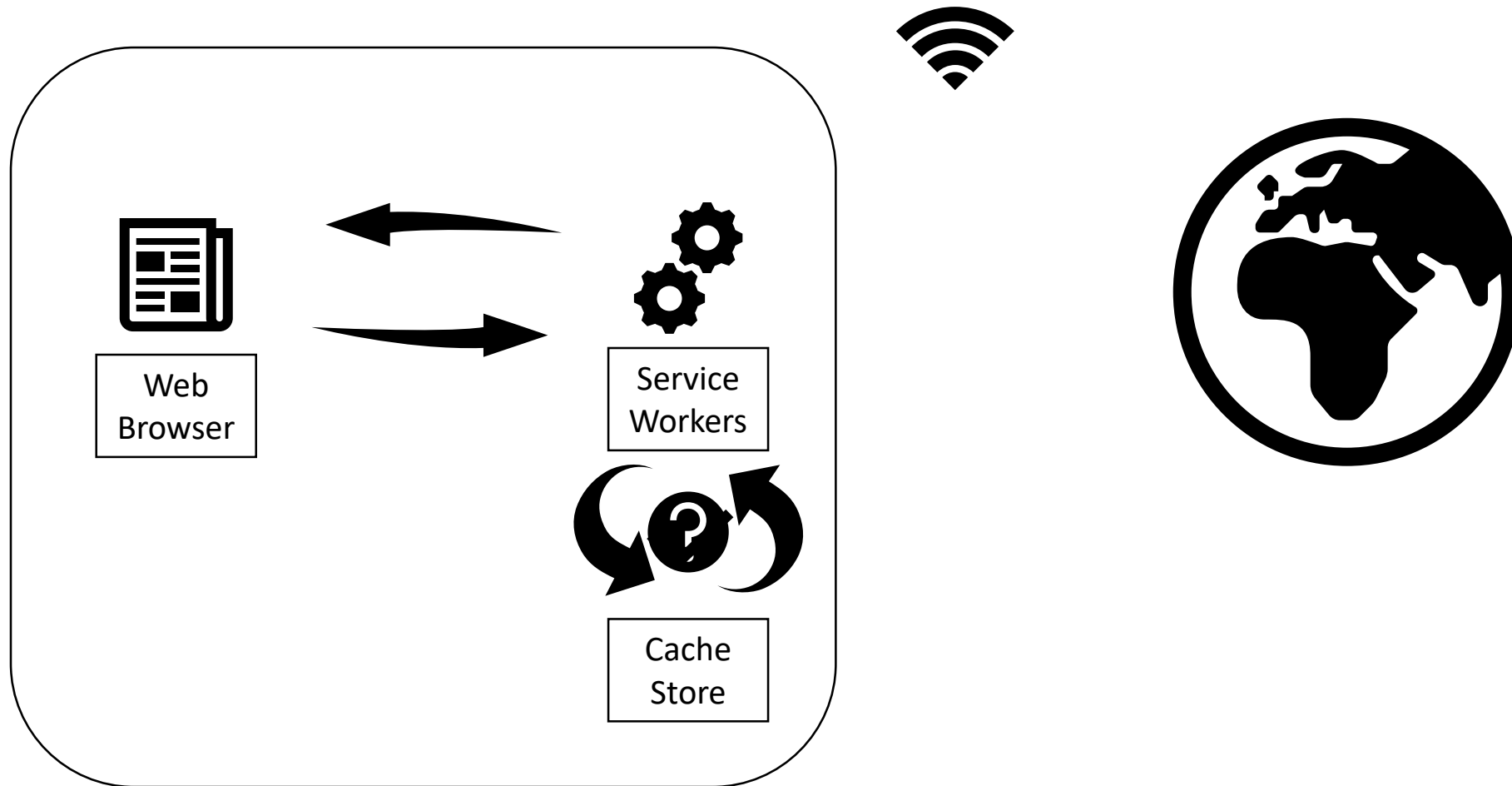
## Scenarios

- Working offline



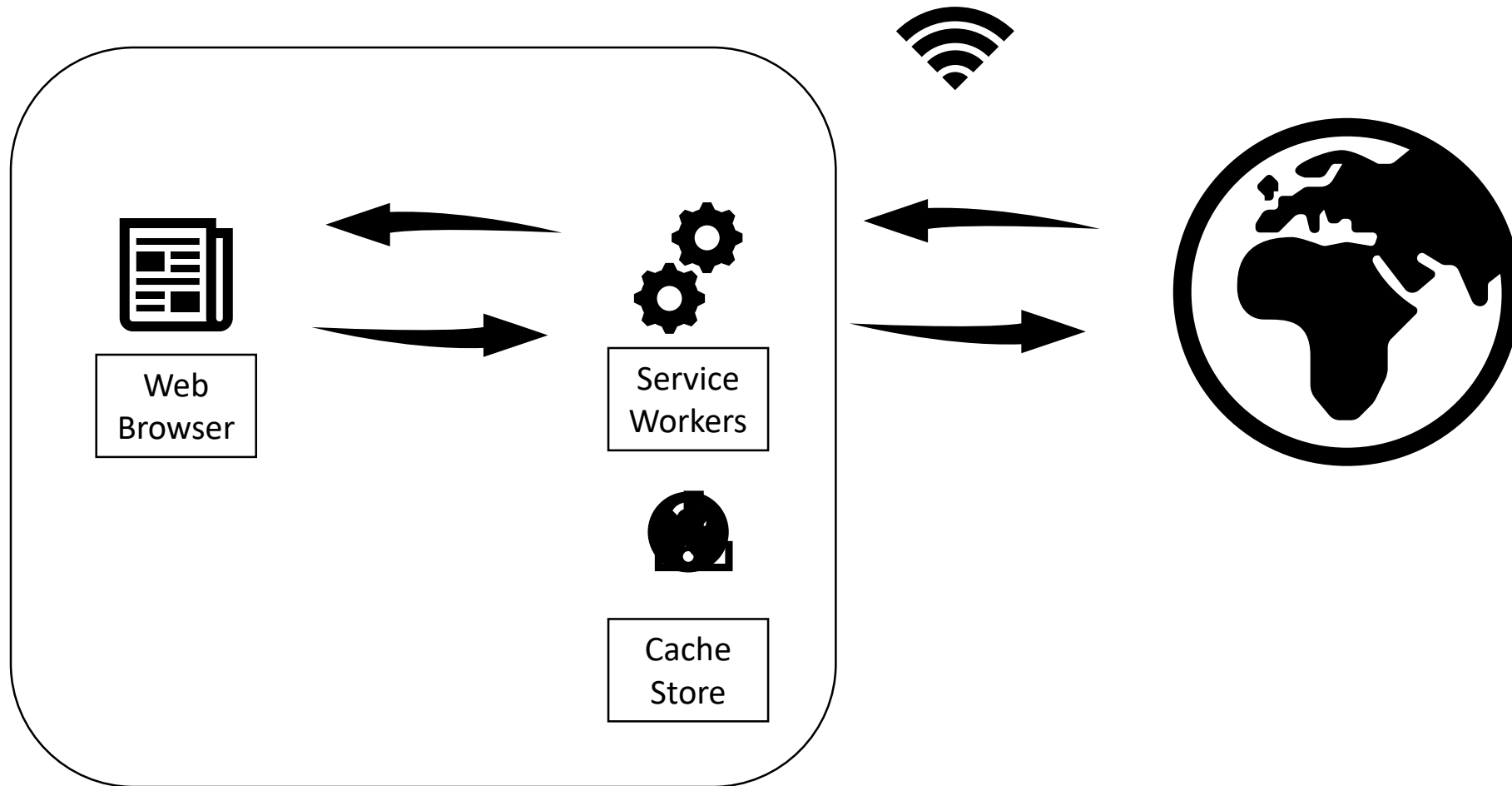
## Scenarios

- Check if asset is cached (yes)



## Scenarios

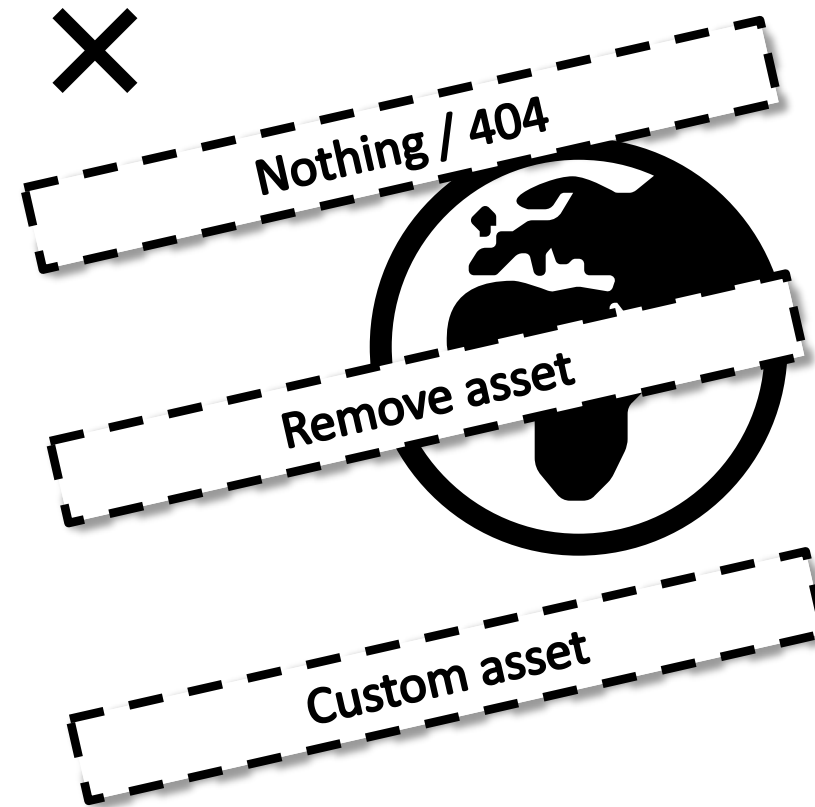
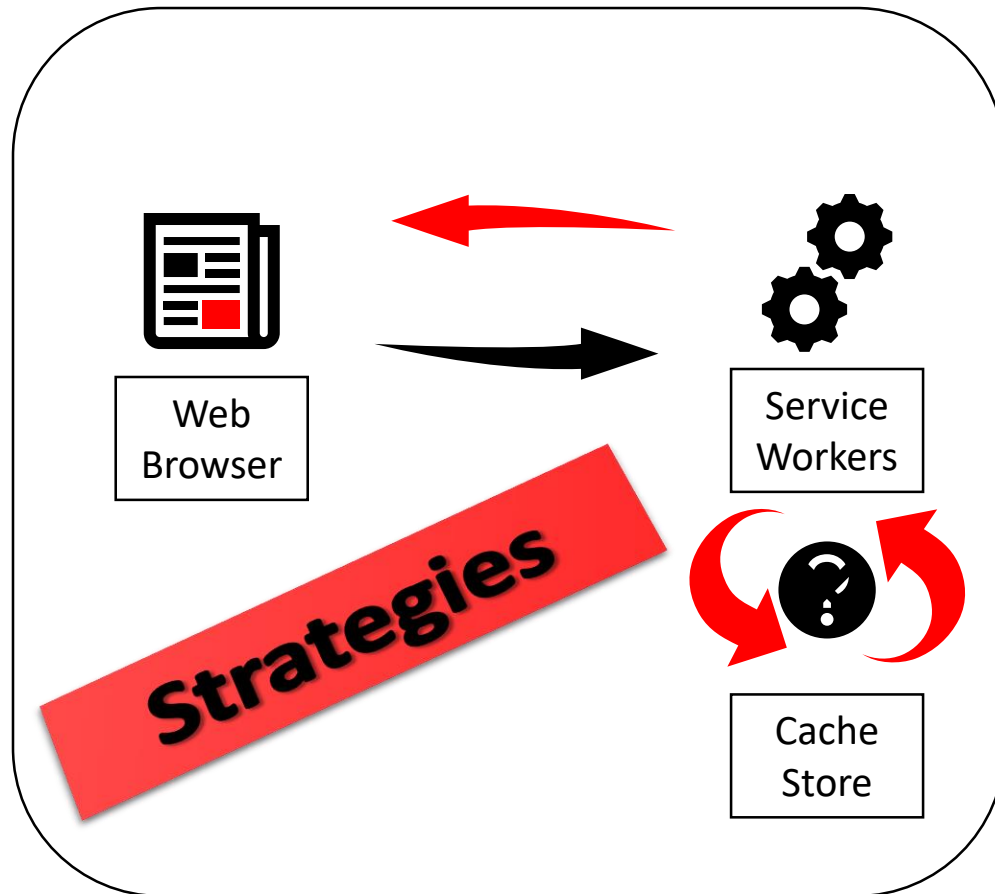
- Check if asset is cached (no)





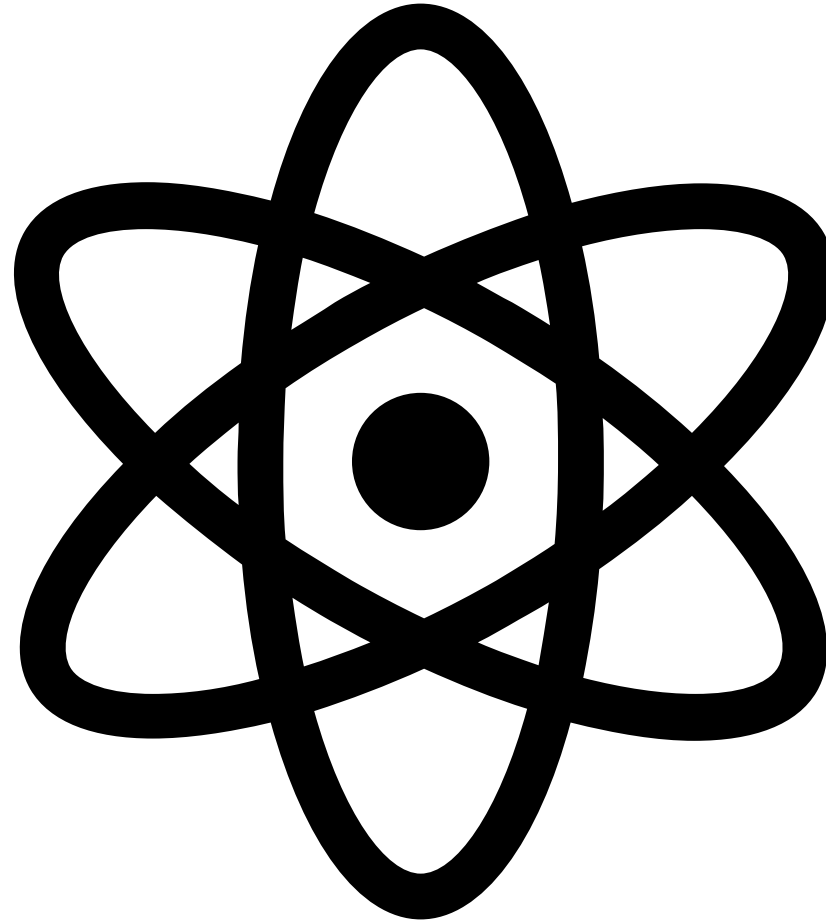
## Scenarios

- Check if asset is cached (no)



# Lifecycle

- Lifecycle



Register

Configuration

Installing

Waiting

Active

Fetch

## Lifecycle

- Register Service Workers in the Web Application



```
<script>
  navigator.serviceWorker.register('serviceWorker.js')
    .then(reg => console.log('SW registered!', reg))
    .catch(err => console.log('SW not registered', err));
</script>
```

**Register**

Configuration

Installing

Waiting

Active

Fetch



# Attention



**You don't have access to window objects**



**Service worker file run only over HTTPS server or your localhost**



**Service worker file must registered at the root directory of your website**

# Lifecycle

✂ (Configuring the Service Worker)

📋 Installing

The SW is registered in the browser for that site

🕒 Waiting

The SW is waiting for closing all the clients



✓ Active

Precious SW (if exists) is gone, new SW is able to control clients

Register

Configuration

Installing

Waiting

Active

Fetch

# Lifecycle

- Some configuration for the Web Application



```
var BASEdir = '/icwe18tut/ServiceWorkers/';
var cacheName = 'v0';
var offlineHTML = BASEdir+'content-not-available.html';
var offlineIMG = BASEdir+'img/offlineImage.png';

////////////////////////////////////
// Fixed, Incremental, Free --> Schemas for offline navigation
var incrementalSchema = true;
////////////////////////////////////

// Assets
```

Register

Configuration

Installing

Waiting

Active

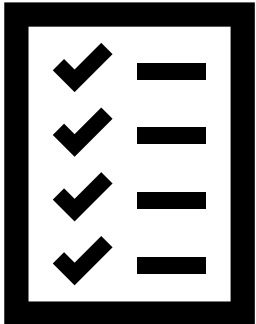
Fetch

Schema	Available elements	Initial elements	Does it take into account user actions?
Fixed	Fixed	Yes	No
Free	Variable	No	Yes
Incremental	Fixed + Variable	Yes	Yes

```
BASEdir+'content-not-available.html';
}
```

# Lifecycle

- Events
  - install



```
self.addEventListener('install', function(e){
  console.log("[ServiceWorker] Installed");

  //////////////////////////////////////
  // IMPORTANT !!!
  // Avoid waiting for closing all the clients
  // Remove if data consistence is important !!!
  //self.skipWaiting(); // <-----
  //////////////////////////////////////

  e.waitUntil(
    caches.open(cacheName).then(function(cache){
      console.log("[ServiceWorker] Caching cacheFiles");
      return cache.addAll(cacheFiles);
    }).catch(error =>{
      console.log('ERROR: ' + error);
    })
  )
});
```

[Register](#)[Configuration](#)[Installing](#)[Waiting](#)[Active](#)[Fetch](#)

# Lifecycle

- Events
  - activate

```
self.addEventListener('activate', function(e){
  console.log("[ServiceWorker] Activated");

  e.waitUntil(
    caches.keys().then(function(cacheNames){
      return Promise.all(cacheNames.map(function(thisCacheName){
        if(thisCacheName !== cacheName){
          console.log("[ServiceWorker] Removing Cached Files from", thisCacheName);
          return caches.delete(thisCacheName);
        }
      }));
    })
  );
});
```

Register

Configuration

Installing

Waiting

**Active**

Fetch



# Lifecycle

- Events
  - fetch

```
self.addEventListener('fetch', function(event) {
  console.log('[ServiceWorker] Fetch', event.request.url);
  event.respondWith(
    caches.match(event.request)
      .then(function(response) {
        if(response) {
          console.log("From cache");
          return response;
        } else {
          console.log("From network");
          return fetch(event.request).then(function(response) {
            if(incrementalSchema) {
              caches.open(cacheName).then(function(cache) {
                cache.put(event.request, response.clone());
              });
            }
            return response.clone();
          }).catch(error => {
            var ext = event.request.url.split('.').pop();
            if( ['html', 'htm', 'php'].indexOf(ext) >= 0) {
              return caches.match(offlineHTML);
            } else if( ['png', 'jpg'].indexOf(ext) >= 0) {
              return caches.match(offlineIMG);
            } else {
              return caches.match(offlineHTML);
            }
          });
        }
      })
  );
});
```

Register

Configuration

Installing

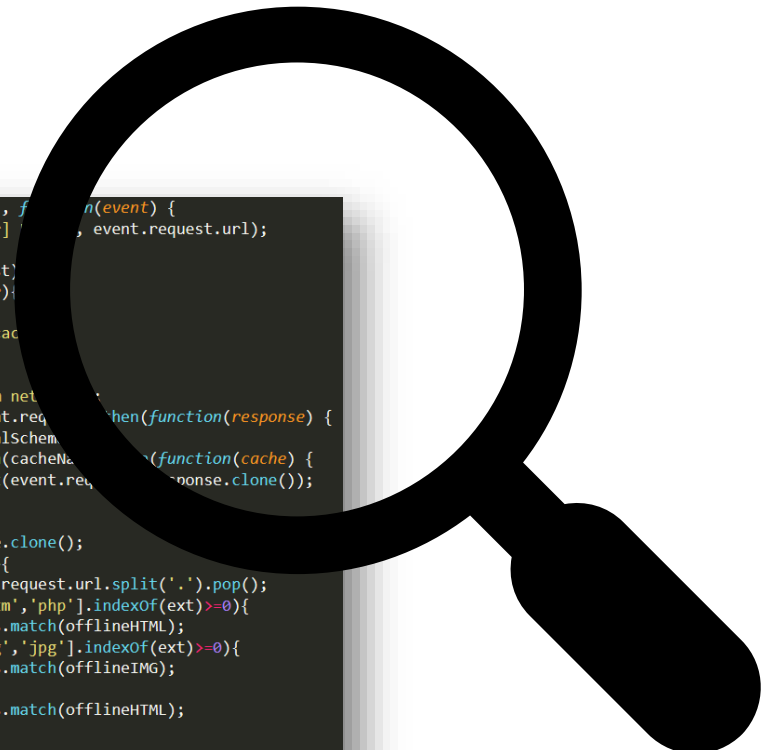
Waiting

Active

**Fetch**

# Lifecycle

- Dive into the fetch Event



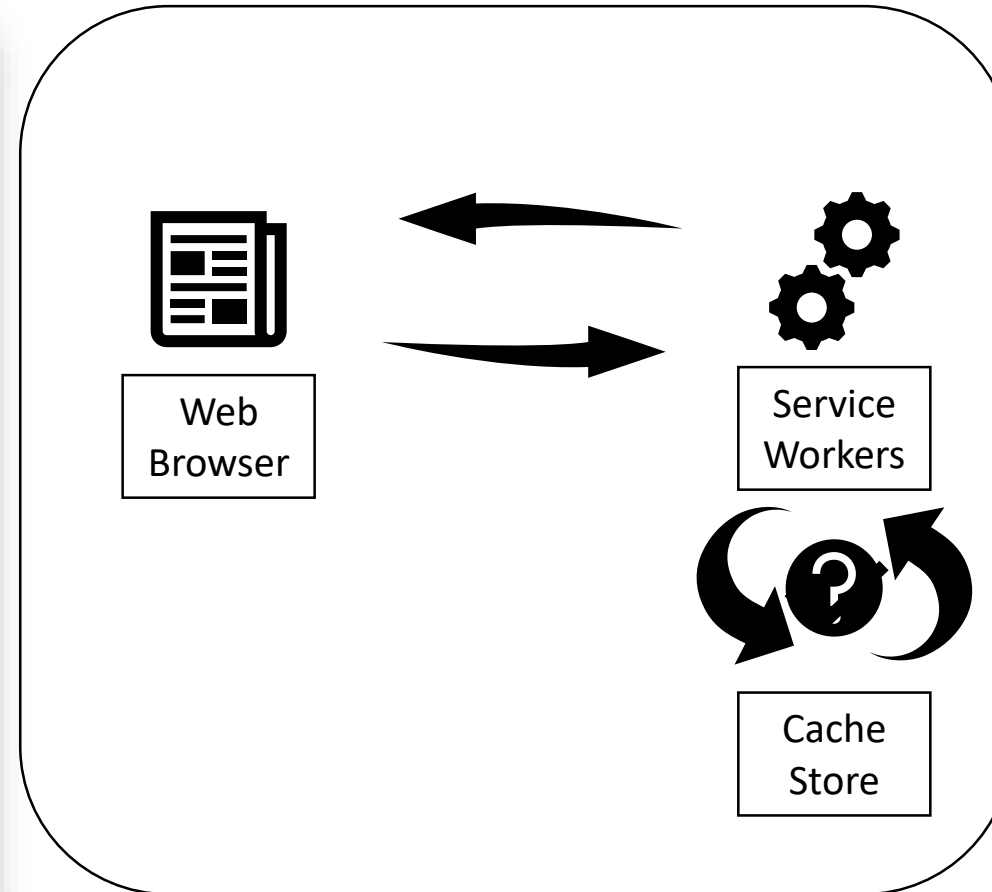
```
self.addEventListener('fetch', function(event) {
  console.log('[ServiceWorker] fetch', event.request.url);
  event.respondWith(
    caches.match(event.request)
      .then(function(response) {
        if(response){
          console.log("From cache", event.request.url);
          return response;
        } else {
          console.log("From network", event.request.url);
          return fetch(event.request).then(function(response) {
            if(!!response && response.ok) {
              caches.open(cacheName).then(function(cache) {
                cache.put(event.request, response.clone());
              });
            }
            return response.clone();
          }).catch(error => {
            console.log('Fetch error', error);
            var ext = event.request.url.split('.').pop();
            if( ['html', 'htm', 'php'].indexOf(ext) >= 0){
              return caches.match('offlineHTML');
            } else if( ['png', 'jpg'].indexOf(ext) >= 0){
              return caches.match('offlineIMG');
            } else {
              return caches.match('offlineHTML');
            }
          });
        }
      })
  );
});
```

[Register](#)[Configuration](#)[Installing](#)[Waiting](#)[Active](#)[Fetch](#)

## Scenarios

- Check if asset is cached (yes)

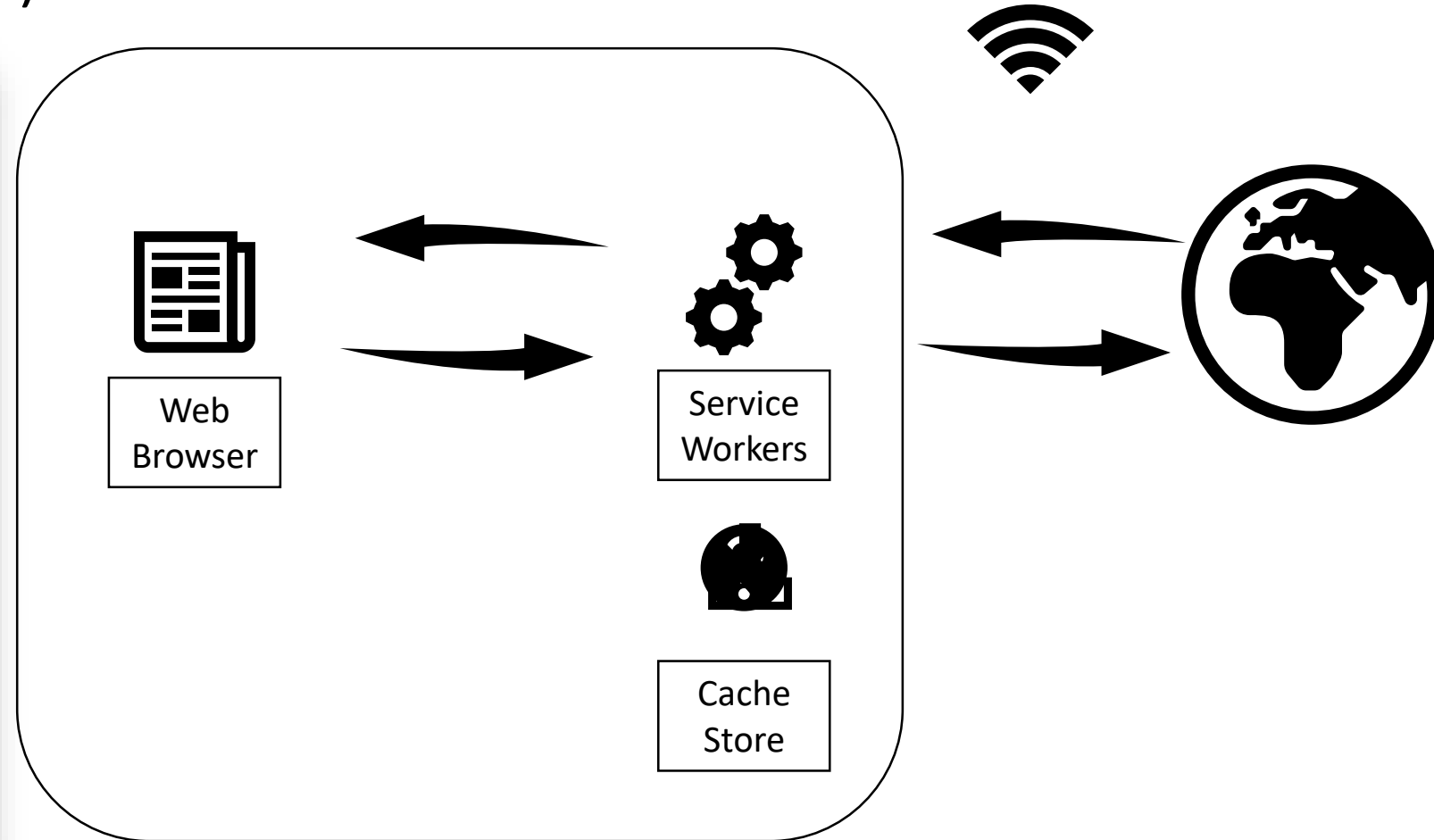
```
self.addEventListener('fetch', function(event) {  
  console.log('[ServiceWorker] Fetch', event.request.url);  
  event.respondWith(  
    caches.match(event.request)  
      .then(function(response) {  
        if(response) {  
          console.log("From cache");  
          return response;  
        } else {  
          console.log("From network");  
          return fetch(event.request).then(function(response) {  
            if(incrementalSchema) {  
              caches.open(cacheName).then(function(cache) {  
                cache.put(event.request, response.clone());  
              });  
            }  
            return response.clone();  
          }).catch(error => {  
            var ext = event.request.url.split('.').pop();  
            if( ['html', 'htm', 'php'].indexOf(ext) >= 0) {  
              return caches.match(offlineHTML);  
            } else if( ['png', 'jpg'].indexOf(ext) >= 0) {  
              return caches.match(offlineIMG);  
            } else {  
              return caches.match(offlineHTML);  
            }  
          });  
        }  
      });  
    );  
  });  
});
```



## Scenarios

- Check if asset is cached (no)

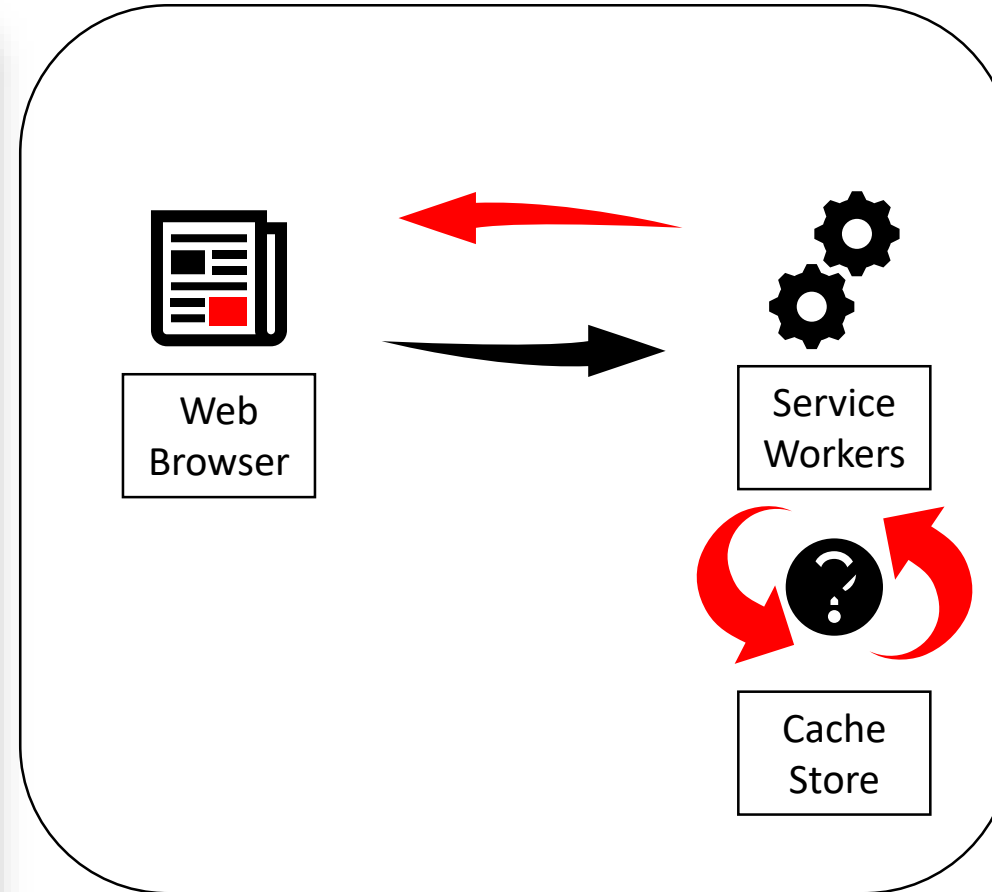
```
self.addEventListener('fetch', function(event) {  
  console.log('[ServiceWorker] Fetch', event.request.url);  
  event.respondWith(  
    caches.match(event.request)  
      .then(function(response) {  
        if(response) {  
          console.log("From cache");  
          return response;  
        } else {  
          console.log("From network");  
          return fetch(event.request).then(function(response) {  
            if(incrementalSchema) {  
              caches.open(cacheName).then(function(cache) {  
                cache.put(event.request, response.clone());  
              });  
            }  
            return response.clone();  
          }).catch(error => {  
            var ext = event.request.url.split('.').pop();  
            if( ['html', 'htm', 'php'].indexOf(ext) >= 0) {  
              return caches.match(offlineHTML);  
            } else if( ['png', 'jpg'].indexOf(ext) >= 0) {  
              return caches.match(offlineIMG);  
            } else {  
              return caches.match(offlineHTML);  
            }  
          });  
        }  
      });  
  });  
});
```



## Scenarios

- Check if asset is cached (no)

```
self.addEventListener('fetch', function(event) {
  console.log('[ServiceWorker] Fetch', event.request.url);
  event.respondWith(
    caches.match(event.request)
      .then(function(response) {
        if(response) {
          console.log("From cache");
          return response;
        } else {
          console.log("From network");
          return fetch(event.request).then(function(response) {
            if(incrementalSchema) {
              caches.open(cacheName).then(function(cache) {
                cache.put(event.request, response.clone());
              });
            }
            return response.clone();
          }).catch(error => {
            var ext = event.request.url.split('.').pop();
            if( ['html', 'htm', 'php'].indexOf(ext) >= 0) {
              return caches.match(offlineHTML);
            } else if( ['png', 'jpg'].indexOf(ext) >= 0) {
              return caches.match(offlineIMG);
            } else {
              return caches.match(offlineHTML);
            }
          });
        }
      })
  );
});
```



## Lifecycle

- Debugging tool (Chrome)
  - Chrome
  - Application
    - Application
      - Service Workers
  - Application
    - Cache
      - Cache Storage

```
var cacheName = 'v0';
```

localhost/icwe18tut/ServiceWorkers

Source [serviceWorker.js](#)


Received 28/5/2018 11:54:52

Status ● #6185 activated and is running [stop](#)

Cache

▼  Cache Storage

 v0 - http://localhost

 Application Cache

```
var cacheName = 'v1';
```

localhost/icwe18tut/ServiceWorkers

Source [serviceWorker.js](#)

Received 28/5/2018 11:54:52

Status ● #6185 activated and is running [stop](#)

● #6186 waiting to activate [skipWaiting](#)


Received 28/5/2018 11:56:43

Cache

▼  Cache Storage

 v0 - http://localhost

 v1 - http://localhost

 Application Cache

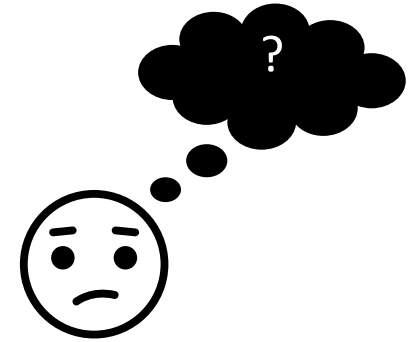
## Service Workers



- Lessons learned
  - Be aware of the Browsers cache
  - <ctrl><f5> updates cache ... or not
    - <f5> shows previous version
  - shift+reload a document (Debugger tools), it'll always load without a controller, which is handy for testing quick CSS & JS changes
  - Only updates when browser closes completely
    - But, self.skipWaiting()!
    - Service Workers work as native applications --> close the browser completely for update them
  - Code Consistency and Data consistency
    - Different browser tabs may run different code --> corrupt client-side database

## Service Workers

- Lessons learned
  - event.request
    - Be careful when using it
      - `console.log(event.request)` ?
    - Only one use
      - Clone
  - Caches.match vs fetch
    - When offline always caches.match (avoid the problems I have for that mistake ;) )
  - Reload the Web application to use the new version





## Service Workers

- The secret life of a Service Worker
  - Service Worker do not ends when the browser is closed
    - Variables ...
  - But if the computer is restarted, the service worker starts again



## Service Workers

- \*Bonus
  - Communication between the Web Application and the Service Worker
    - Using MessageChannel
  - Push notifications
  - Background sync API

# MessageChannel

- Part of the Channel Messaging API
  - Allows the creation of a new message channel
  - Send / Receive data through it via its two MessagePort properties

```
class swMessageClient{
  constructor(){
    this._channel = null;
    this._port1 = null;
    this._port2 = null;
    this._command = null;
    this._message = null;
    this._data = null;
  }

  _ini(){
    this._channel = new MessageChannel();
    this._port1 = this._channel.port1;
    this._port2 = this._channel.port2;
    this._data = {
      command: this._command,
      msg: this._message
    };
  }

  _sendMessage(v){
    navigator.serviceWorker.controller.postMessage(v, [this._port2]);
  }

  exec(v){
    this._message = v;
    this._ini();
    var _this = this;
    return new Promise(function(resolve, reject){
      this._port1.onmessage = function(e){
        if(e.data.error){
          reject(e.data.error);
        }else{
          resolve(e.data);
        }
      };
      _this._sendMessage(JSON.stringify(_this._data));
    });
  }
}
```

Client Side

# MessageChannel

```
class chkFileOnCache extends swMessageClient{
  constructor(){
    super();
    this._command='chkFileOnCache';
  }
}

class chkConnectivity extends swMessageClient{
  constructor(){
    super();
    this._command='chkConnectivity';
  }
}
```

Client Side

```
self.addEventListener('message', function(event){
  var data = JSON.parse(event.data);
  switch(data.command){
    case 'chkFileOnCache':
      caches.match(data.msg)
        .then(function(response){
          if(response){
            event.ports[0].postMessage(true);
          }else{
            event.ports[0].postMessage(false);
          }
        })
      break;
    case 'chkConnectivity':
      var myHeaders = new Headers();
      myHeaders.append('pragma', 'no-cache');
      myHeaders.append('cache-control', 'no-cache');
      var myInit = {
        method: 'GET',
        headers: myHeaders,
      };
      var myRequest = new Request('index.html');
      fetch(myRequest, myInit)
        .then(function(response){
          if(response){
            event.ports[0].postMessage(true);
          }else{
            event.ports[0].postMessage(false);
          }
        })
        .catch(function(error){
          event.ports[0].postMessage(false);
        });
      break;
    default:
      console.log('Service Worker Message Error :'+event.data);
  }
});
```

Service Workers Side

# MessageChannel

- Execution
  - Creation of the MessageChannel
    - (using the previous class)
  - Send the message
    - (exec method)
  - Read the result
    - PromiseValue

```
> var test = new chkFileOnCache();  
< undefined  
  
> test.exec("C1.html");  
< ▼ Promise {<pending>} ⓘ  
  ► __proto__: Promise  
    [[PromiseStatus]]: "resolved"  
    [[PromiseValue]]: true  
  
> test.exec("C2.html");  
< ▼ Promise {<pending>} ⓘ  
  ► __proto__: Promise  
    [[PromiseStatus]]: "resolved"  
    [[PromiseValue]]: false
```

Execution on the Console

```
this._fileOnCache.exec(e.getAttribute(this._getURL)).then(  
  function(m){  
    if(m){  
      // The file exists on the cache  
    }else{  
      // The file doesn't exists on the cache  
    }  
  }  
);
```

Execution on a Web Application

## Push Notifications

- Push Notifications
  - The Notifications API displays notifications to the user
  - It is incredibly powerful and simple to use
  - Where possible, it uses the same mechanisms a native app would use, giving a completely native look and feel

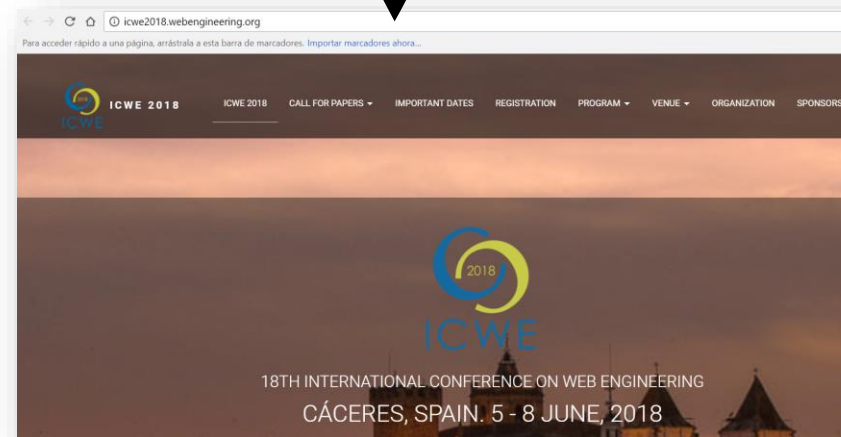
# Push Notifications

- Push Notifications
  - First request permission for sending notifications
    - Notification.requestPermission
  - If granted
    - showNotification method to send notifications

```
Notification.requestPermission(function(result) {  
  if (result === 'granted') {  
    navigator.serviceWorker.ready.then(function(registration) {  
      registration.showNotification('ICWE 2018 Registration', {  
        body: 'Please, complete the registration process.',  
        vibrate: [200, 100, 200, 100, 200, 100, 200],  
        icon: 'img/logoBig.png',  
        tag: 'Registration to ICWE 2018',  
        data: {  
          dateOfArrival: Date.now(),  
          primaryKey: 'registration'  
        }  
      });  
    });  
  }  
});
```

## Push Notifications

- Case Study 1
  - Show an invitation to the user for participating in a conference

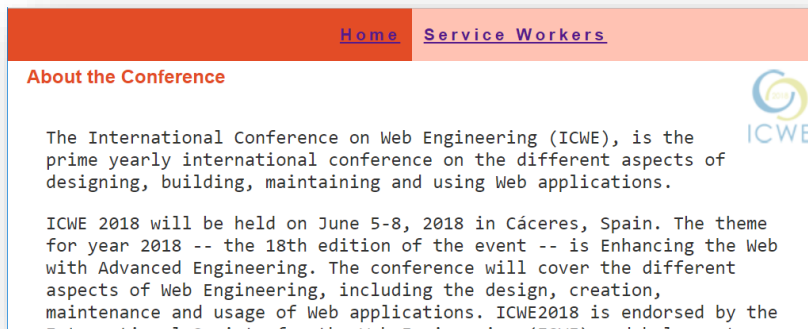




# Push Notifications

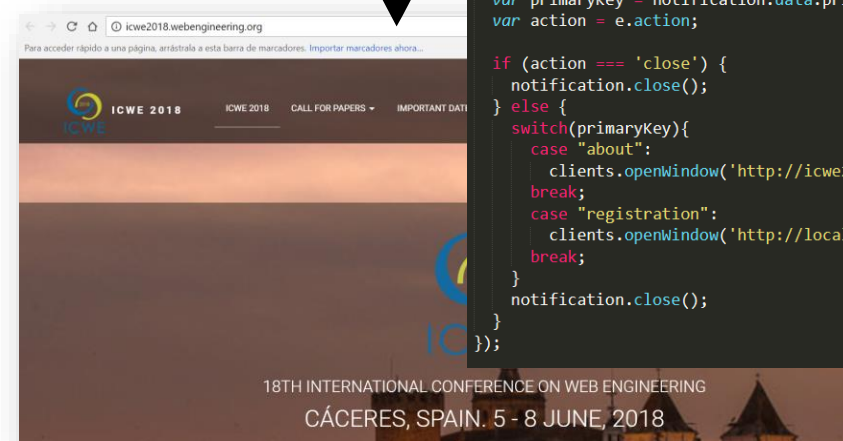
- Case Study 1
  - Show an invitation to the user for participating in a conference

```
self.addEventListener('notificationclose', function(e) {
  var notification = e.notification;
  var primaryKey = notification.data.primaryKey;
  console.log('Closed notification: ' + primaryKey);
  if(primaryKey=="registration" && retryRegistration){
    setTimeout(function(){
      self.registration.showNotification('ICWE 2018 Registration', {
        body: 'Please, complete the registration process.',
        vibrate: [200, 100, 200, 100, 200, 100, 200],
        icon: 'img/logoBig.png',
        tag: 'Registration to ICWE 2018',
        data: {
          dateOfArrival: Date.now(),
          primaryKey: 'registration'
        }
      });
    }, 10000);
  }
});
```



```
function showNotification() {
  Notification.requestPermission(function(result) {
    if (result === 'granted') {
      navigator.serviceWorker.ready.then(function(registration) {
        registration.showNotification('Invitation to ICWE 2018', {
          body: '18TH INTERNATIONAL CONFERENCE ON WEB ENGINEERING CÁCERES, SPAIN. 5 - 8 JUNE, 2018',
          vibrate: [200, 100, 200, 100, 200, 100, 200],
          icon: 'img/logoBig.png',
          tag: 'Invitation to ICWE 2018',
          data: {
            dateOfArrival: Date.now(),
            primaryKey: "about"
          }
        });
      });
    }
  });
}

window.onload = function(){
  showNotification();
}
```



```
self.addEventListener('notificationclick', function(e) {
  var notification = e.notification;
  var primaryKey = notification.data.primaryKey;
  var action = e.action;

  if (action === 'close') {
    notification.close();
  } else {
    switch(primaryKey){
      case "about":
        clients.openWindow('http://icwe2018.webengineering.org/');
        break;
      case "registration":
        clients.openWindow('http://localhost/icwe18tut/ServiceWorkers/RegistrationForm.html');
        break;
    }
    notification.close();
  }
});
```

# Push Notifications

- Case Study 2
  - Remember the user to complete the registration process



# Push Notifications

- Case Study 2
- Remember the

## Registration Form

First name:

Last name:

You are not registered !!!

```
self.addEventListener('notificationclick', function(e) {
  var notification = e.notification;
  var primaryKey = notification.data.primaryKey;
  var action = e.action;

  if (action === 'close') {
    notification.close();
  } else {
    switch(primaryKey){
      case "about":
        clients.openWindow('http://icwe2018.webengineering.org/');
        break;
      case "registration":
        clients.openWindow('http://localhost/icwe18tut/ServiceWorkers/RegistrationForm.html');
        break;
    }
    notification.close();
  }
});
```

```
class retryRegistration extends swMessageClient{
  constructor(){
    super();
    this._command='retryRegistration';
  }
  var formSubmitted = false;
  var retry = null;
  function setRetryRegistration(v){
    setRetryRegistration(false);
    retry = v;
    var DoRetryRegistration = new retryRegistration();
    DoRetryRegistration.exec(retry);
  }
}
```

## ICWE 2018 Registration

Please, complete the registration process.

localhost

```
window.addEventListener("beforeunload", function (event) {
  if(!formSubmitted){
    setRetryRegistration(true);
    showNotification();
  }
});
```

```
self.addEventListener('notificationclose', function(e) {
  var notification = e.notification;
  var primaryKey = notification.data.primaryKey;
  console.log('Closed notification: ' + primaryKey);
  if(primaryKey=="registration" && retryRegistration){
    setTimeout(function(){
      self.registration.showNotification('ICWE 2018 Registration', {
        body: 'Please, complete the registration process.',
        vibrate: [200, 100, 200, 100, 200, 100, 200],
        icon: 'img/logoBig.png',
        tag: 'Registration to ICWE 2018',
        data: {
          dateOfArrival:Date.now(),
          primaryKey:'registration'
        }
      });
    },10000);
  }
});
```

```
function showNotification() {
  if(retry){
    Notification.requestPermission(function(result) {
      if (result === 'granted') {
        navigator.serviceWorker.ready.then(function(registration) {
          registration.showNotification('ICWE 2018 Registration', {
            body: 'Please, complete the registration process.',
            vibrate: [200, 100, 200, 100, 200, 100, 200],
            icon: 'img/logoBig.png',
            tag: 'Registration to ICWE 2018',
            data: {
              dateOfArrival:Date.now(),
              primaryKey:'registration'
            }
          });
        });
      }
    });
  }
}
```

## Registration Form

First name:

Last name:

Thank you for your registration :)

```
function submitRegistration(){
  setRetryRegistration(false);
  formSubmitted = true;
  updateStatus("Thank you for your registration :)");
}
```

# Index

 Introduction

 API

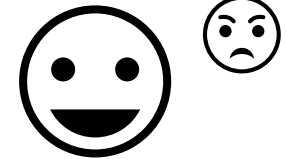
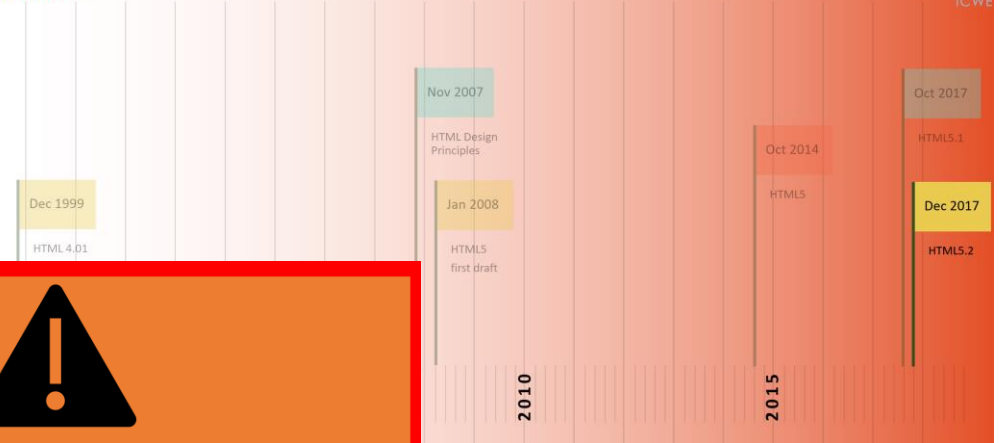
 Evolution

 Hands on

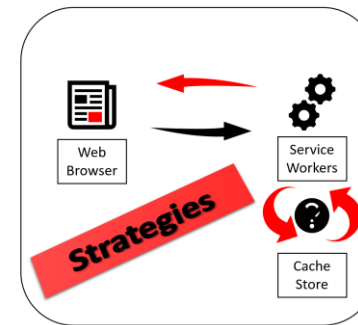
 Conclusions

## Conclusions

### Timeline



- Check if asset is cached (no)



- Why, When, How



*Mastropiero Terrace*



*one of the most interesting  
terraces in Spain to have some  
"tapas"*

# Review of the HTML5 API

Computer Systems Department, University of Castilla-La Mancha  
Albacete, Spain  
[felix.albertos@uclm.es](mailto:felix.albertos@uclm.es)



5th June 2018