



HTML5

Avanzado



Data

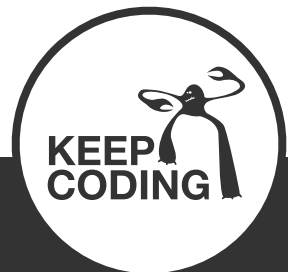


Permite almacenar datos en los elementos del DOM y acceder fácilmente a ellos desde JS

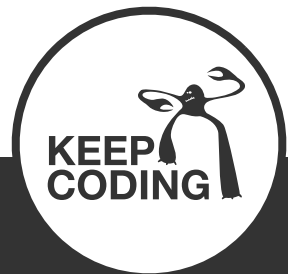


```
<div id="flower" data-leaves='47' data-plant-height='2.4m'></div>
```

```
<script>  
  var plant = document.getElementById('flower');  
  console.log(plant.dataset.leaves);  
  console.log(plant.dataset.plantHeight);  
  // 'plant-height' -> 'plantHeight'  
  plant.dataset.plantHeight = '3.6m';  
</script>
```



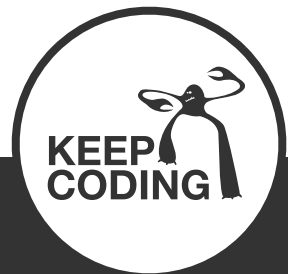
Scalable Vector Graphics (SVG)





SVG es:

- Lenguaje de marcado XML
- Representa formas y trazos
- Gráficos vectoriales



Estructura básica del SVG

```
<svg version="1.1"
  baseProfile="full"
  width="300" height="200"
  xmlns="http://www.w3.org/2000/svg">

  <rect width="100%" height="100%" fill="red" />
  <g>
    <circle cx="150" cy="100" r="80" fill="green" />

    <text x="150" y="125" font-size="60" text-
anchor="middle" fill="white">SVG</text>
  </g>

</svg>
```



Importar un SVG en nuestros proyectos

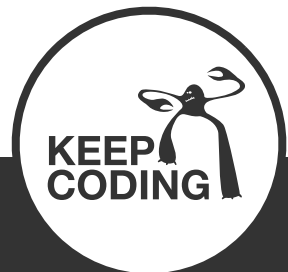
En el propio HTML

Con `<object>`

```
<object data="image.svg" type="image/svg+xml" />
```

Con `<iframe>`

```
<iframe src="image.svg"></iframe>
```



Formas básicas

```
<?xml version="1.0" standalone="no"?>
<svg width="200" height="250" version="1.1" xmlns="http://www.w3.org/2000/svg">

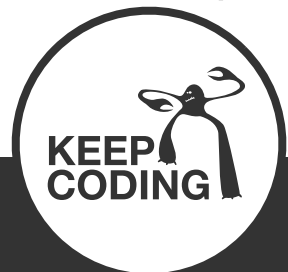
  <rect x="10" y="10" width="30" height="30" stroke="black" fill="transparent" stroke-
width="5"/>
  <rect x="60" y="10" rx="10" ry="10" width="30" height="30" stroke="black"
fill="transparent" stroke-width="5"/>

  <circle cx="25" cy="75" r="20" stroke="red" fill="transparent" stroke-width="5"/>
  <ellipse cx="75" cy="75" rx="20" ry="5" stroke="red" fill="transparent" stroke-width="5"/>

  <line x1="10" x2="50" y1="110" y2="150" stroke="orange" stroke-width="5"/>
  <polyline points="60 110 65 120 70 115 75 130 80 125 85 140 90 135 95 150 100 145"
stroke="orange" fill="transparent" stroke-width="5"/>

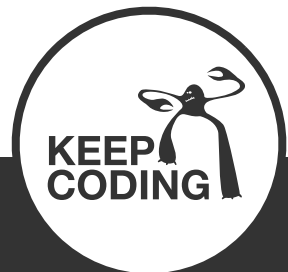
  <polygon points="50 160 55 180 70 180 60 190 65 205 50 195 35 205 40 190 30 180 45 180"
stroke="green" fill="transparent" stroke-width="5"/>

  <path d="M20,230 Q40,205 50,230 T90,230" fill="none" stroke="blue" stroke-width="5"/>
</svg>
```



Ventajas:

- Escalable
- Compresión
- Indexable
- Generable
- Alta resolución
- Estándar abierto
- Modificable



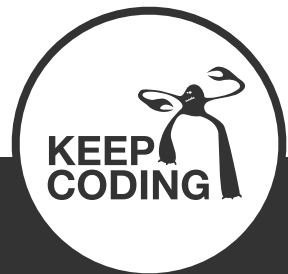
Ejemplos

https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Getting_Started

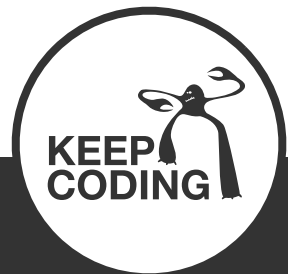
https://www.w3schools.com/graphics/svg_intro.asp

<http://caniuse.com/#search=svg>

<https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Paths>



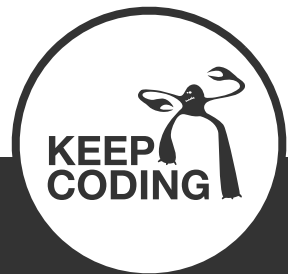
Canvas





Canvas es:

- Un elemento rectangular donde se dibujan gráficos
- Solo es el contenedor, no los gráficos en sí
- Requiere Javascript para dibujar en él



Elemento `<canvas>` y objeto `context`

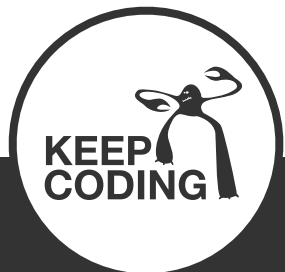
Context es el objeto que tiene todas las propiedades y métodos para dibujar

```
<!-- Elemento canvas -->
<canvas id="my-canvas"></canvas>

// Obteniendo el contexto
<script>

    var element = document.getElementById("my-canvas");
    var context = element.getContext("2d");

</script>
```



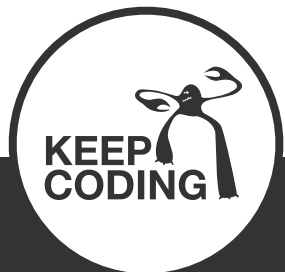
Objeto **context** para 2D y 3D

Un contexto **2d** ofrece herramientas para dibujar gráficos en dos dimensiones

```
<script>
  var element = document.getElementById("my-canvas");
  var context = element.getContext("2d");
</script>
```

Un contexto **webgl** ofrece herramientas para dibujar gráficos en tres dimensiones

```
<script>
  var element = document.getElementById("my-canvas");
  var context = element.getContext("webgl");
</script>
```

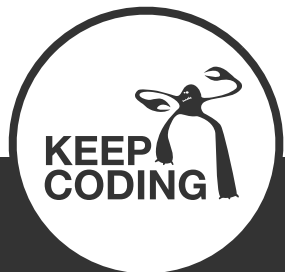




El elemento `<canvas>` no está soportado en las versiones 8 y anteriores de Internet Explorer.

Por su lado, WebGL no está soportado en todos los navegadores. Si necesitas un contexto 3D, cerciórate de comprobar que lo tienes disponible:

```
if (window.WebGLRenderingContext) {...}
```



Ejemplos

En 2D:

<http://www.html5canvastutorials.com>

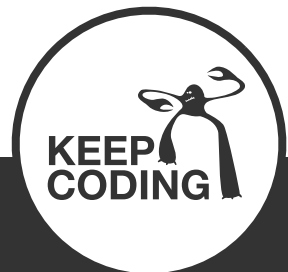
<http://www.effectgames.com/demos/canvacycle/>

<http://cheatsheetworld.com/programming/html5-canvas-cheat-sheet/>

En 3D:

http://learningwebgl.com/blog/?page_id=1217

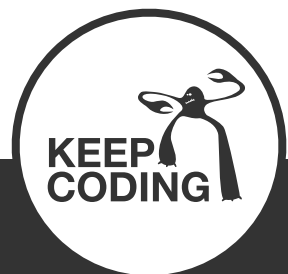
<http://www.chromeexperiments.com/webgl/>



Geolocalización



- Se ofrece la API **navigator.geolocation**
- Su uso requiere autorización por parte del usuario
- La precisión es más alta en terminales con GPS



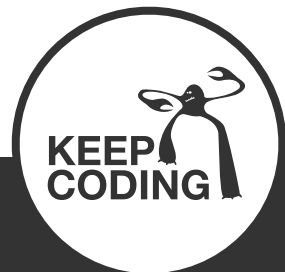
getCurrentPosition

Devuelve la posición actual

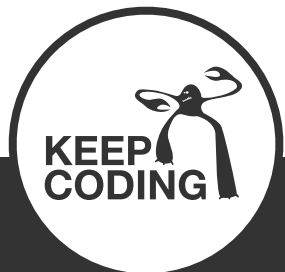
```
<script>
```

```
// El navegador soporta geolocalización
if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(
        successHandler, // manejador de respuesta correcta
        errorHandler, // manejo de errores
        options // opciones
    );
}
// El navegador no soporta geolocalización
else { /* Código*/ }
```

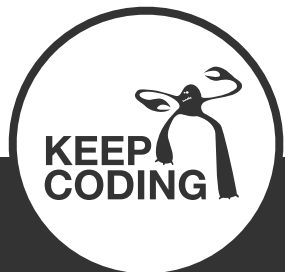
```
</script>
```



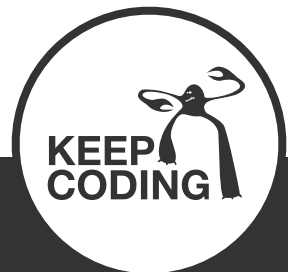
```
function successHandler(data) {  
    var coords = data.coords;  
    console.log("Latitud", coords.latitude);  
    console.log("Longitud", coords.longitude);  
    console.log("Precisión", coords.accuracy);  
    console.log("Altitud", coords.altitude);  
    console.log("Precisión de altitud", coords.altitudeAccuracy);  
    // The heading as degrees clockwise from North  
    console.log("Preguntar a marinero", coords.heading);  
    console.log("Velocidad", coords.speed);  
}
```



```
function errorHandler(error) {
  switch(error.code) {
    case error.PERMISSION_DENIED:
      alert("User denied the request for Geolocation.");
      break;
    case error.POSITION_UNAVAILABLE:
      alert("Location information is unavailable.");
      break;
    case error.TIMEOUT:
      alert("The request to get user location timed out.");
      break;
    case error.UNKNOWN_ERROR:
      alert("An unknown error occurred.");
      break;
  }
}
```



```
var options = {  
    enableHighAccuracy: true, // Más precisión o no  
    timeout: 5000, // Infinito por defecto  
    maximumAge: 0 // Máximo tiempo de caché  
};
```



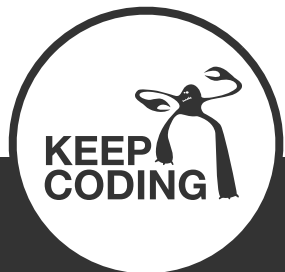
watchPosition() & clearWatch()

Sirve para detectar cuando se llega a una posición

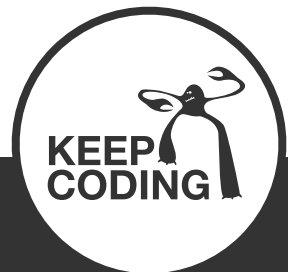
```
<script>
```

```
// El navegador soporta geolocalización
if (navigator.geolocation) {
    var watchId = navigator.geolocation.watchPosition(
        successHandler, // manejador de respuesta correcta
        errorHandler, // manejo de errores
        options // opciones
    );
}
// El navegador no soporta geolocalización
else { /* Código*/ }
```

```
</script>
```

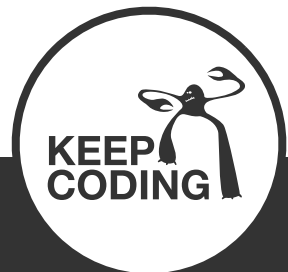



```
function successHandler(data) {  
  var crd = data.coords;  
  if (target.latitude === crd.latitude  
    && target.longitude === crd.longitude) {  
    console.log('Has llegado!');  
    navigator.geolocation.clearWatch(watchId);  
  }  
}
```



Google Maps Embed API

```
<iframe
  width="600"
  height="450"
  frameborder="0" style="border:0"
  src="https://www.google.com/maps/embed/v1/place?key=YOUR_API_KEY
    &q=Lat,Long" allowfullscreen>
</iframe>
```



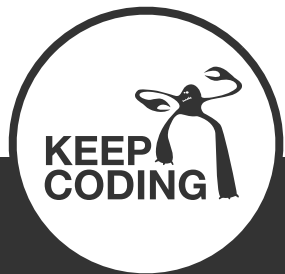
Web Storage





Web Storage es:

- Un almacén local de pares key-value de strings
- Más seguro y rápido que las tradicionales cookies
 - Cada página tiene acceso solo a sus datos locales
 - Los datos no se incluyen en cada petición



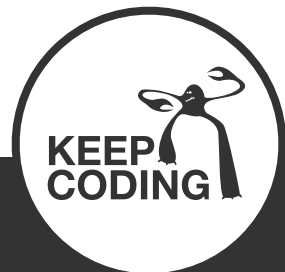
API localStorage

El almacén localStorage no tiene fecha de expiración

```
<script>
```

```
// El navegador soporta Web Storage
if (typeof(Storage) !== "undefined") {
  // Setter
  localStorage.setItem("pageSize", 10);
  // Getter
  var size = Number(localStorage.getItem("pageSize"));
}
// El navegador no soporta Web Storage
else { /* Código*/ }
```

```
</script>
```



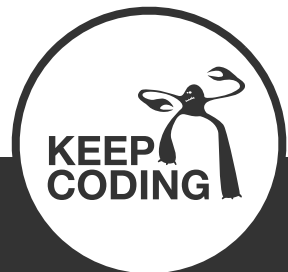
API sessionStorage

El almacén sessionStorage expira con la sesión actual, al cerrar la ventana del navegador

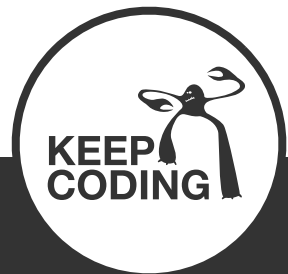
```
<script>
```

```
// El navegador soporta Web Storage
if (typeof(Storage) !== "undefined") {
    // Getter
    var pages = Number(sessionStorage.getItem("pagesNavigated")) + 1;
    // Setter
    sessionStorage.setItem("pagesNavigated", pages);
}
// El navegador no soporta Web Storage
else { /* Código*/ }
```

```
</script>
```



Web Workers



Los Web Workers:

- Ejecutan código Javascript en segundo plano
- No afectan al rendimiento de la página
- No están pensados para tareas simples, sino para tareas que hagan uso intensivo de CPU



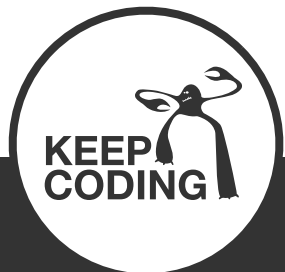
API Web Workers

Ejemplo sencillo de uso de Web Workers

<script>

```
// El navegador soporta Web Workers
if (typeof(Worker) !== "undefined") {
    // Ejecuta el script my_background_task.js en segundo plano
    var worker = new Worker("my_background_task.js");
}
// El navegador no soporta Web Workers
else { /* Código*/ }
```

</script>



API Web Workers

Acciones posibles con los Web Workers

<script>

```
// Teniendo el siguiente Web Worker
var worker = new Worker("my_background_task.js");

// Podemos terminar su ejecución
worker.terminate();
worker = undefined;

// Recibir notificaciones
worker.onmessage = function(event) {
    // En event.data tenemos el dato notificado
};
```

</script>



API Web Workers

Envío de notificaciones al Web Worker

```
// Documento my_background_task.js

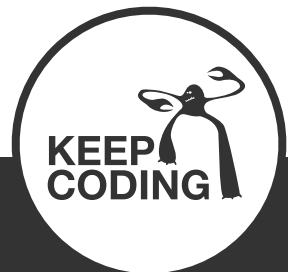
function fibonacci(num) {

    var result;

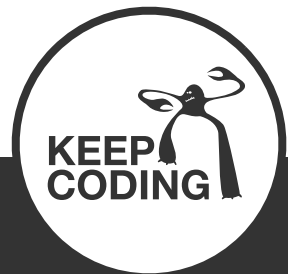
    // Código para calcular el fibonacci de num

    postMessage(result);
}

fibonacci(36);
```

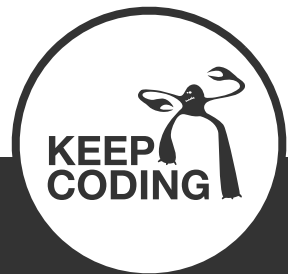


Web Services



Los Web Services:

- Tipo específico de Web Worker
- Ejecutan código Javascript en segundo plano
- No afectan al rendimiento de la página
- Como proxy de las comunicaciones
- Control sobre la experiencia offline

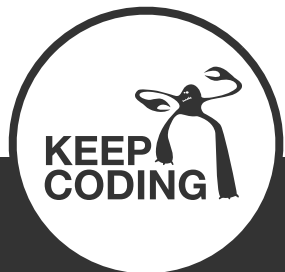


Registrar un Web Service

```
<script>
```

```
if ('serviceWorker' in navigator) {  
  window.addEventListener('load', function() {  
    navigator.serviceWorker.register('/sw.js').then(function(registration) {  
      // Registration was successful  
      console.log('ServiceWorker registration successful with scope: ', registration.scope);  
    }, function(err) {  
      // registration failed :(  
      console.log('ServiceWorker registration failed: ', err);  
    });  
  });  
}
```

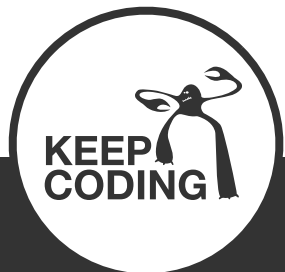
```
</script>
```



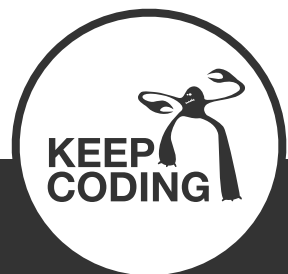
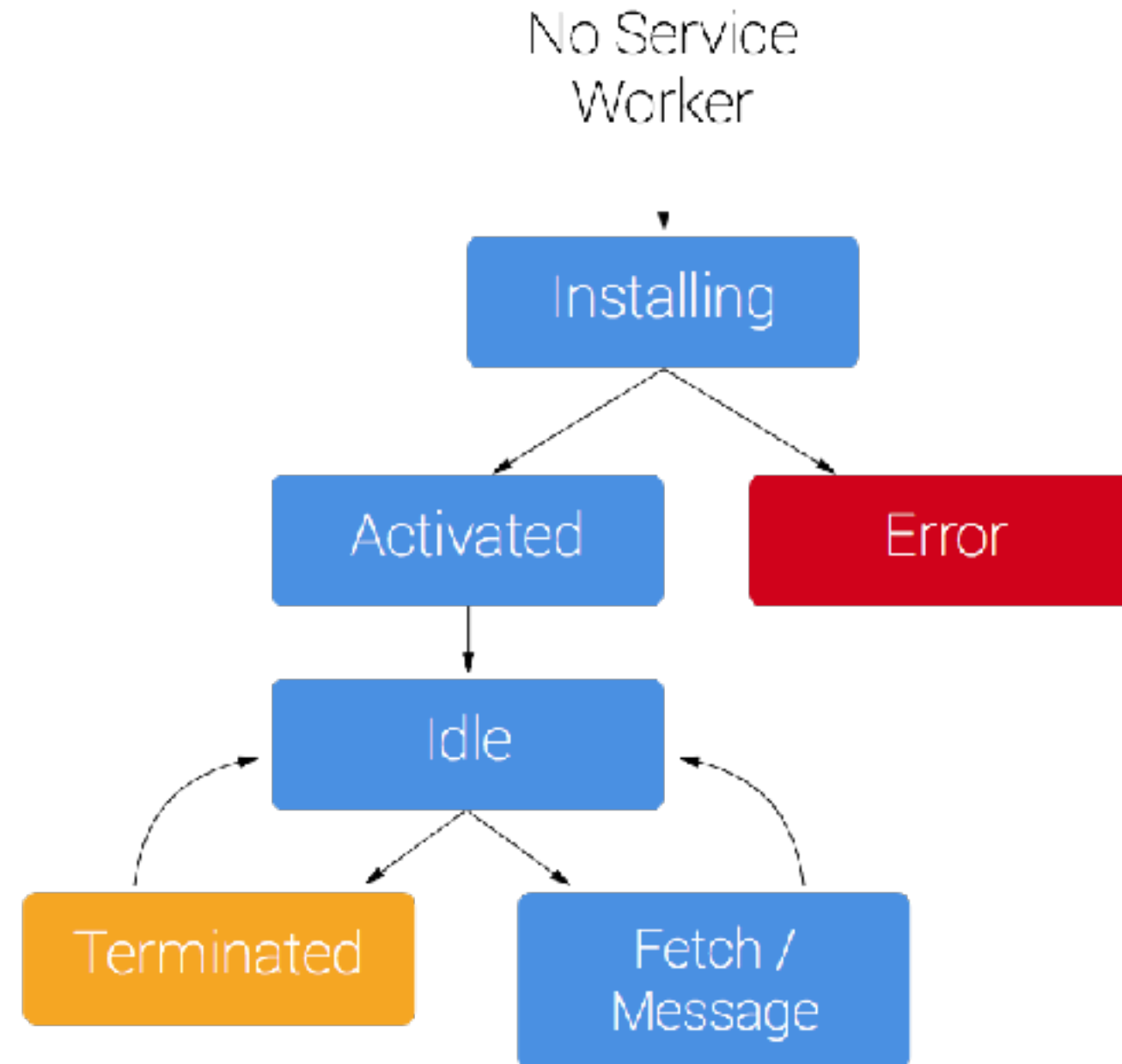
Ejemplo Web Service

```
var CACHE_NAME = 'my-site-cache-v1';
var urlsToCache = [
  '/',
  '/styles/main.css',
  '/script/main.js'
];

self.addEventListener('install', function(event) {
  // Perform install steps
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(function(cache) {
        console.log('Opened cache');
        return cache.addAll(urlsToCache);
      })
  );
});
```



Ciclo de vida Web Services

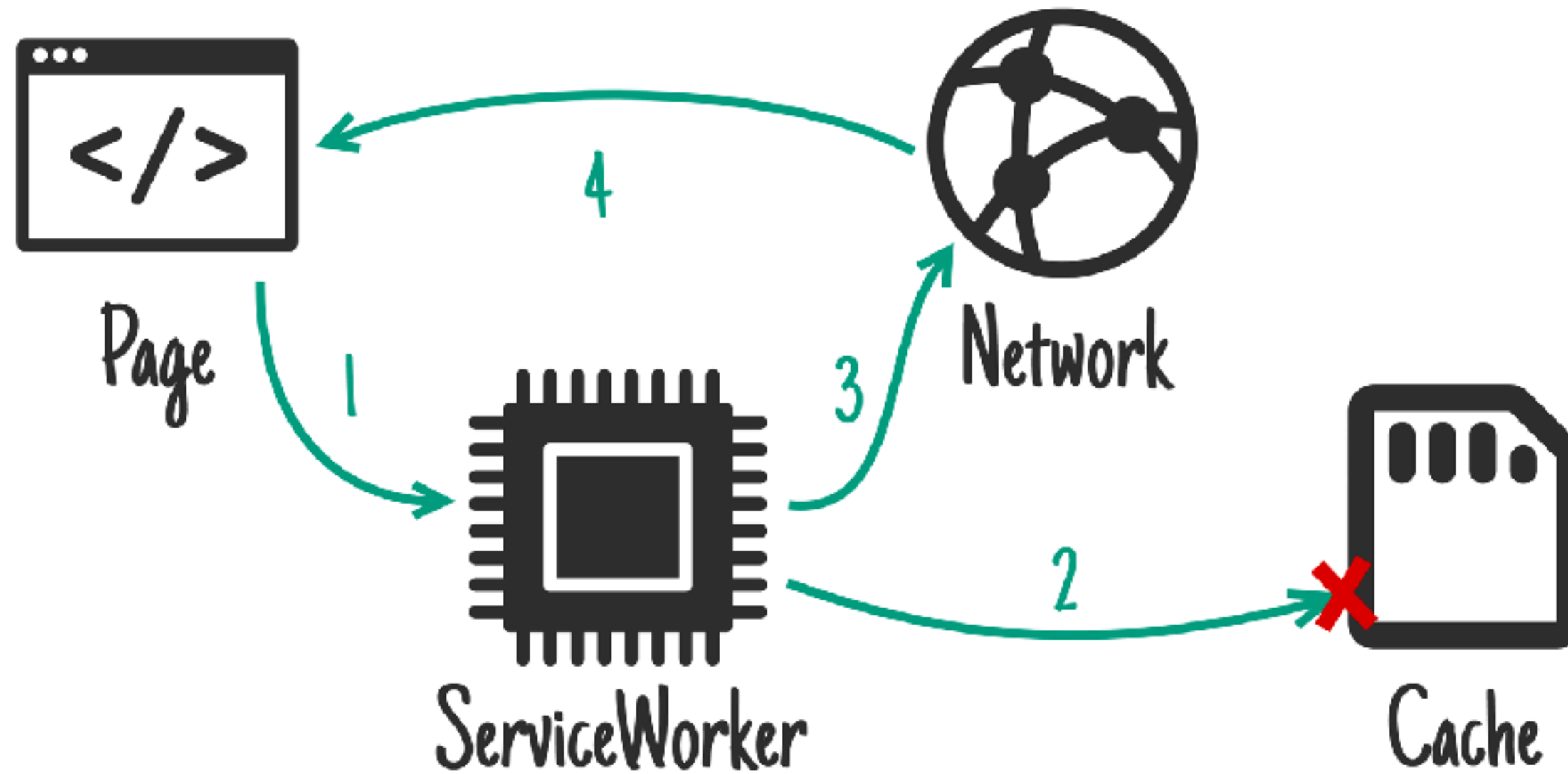


Web Services as Network Cache



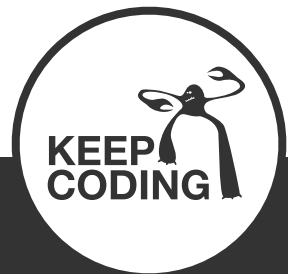
<https://jakearchibald.com/2014/offline-cookbook/>

Cache, network fallback

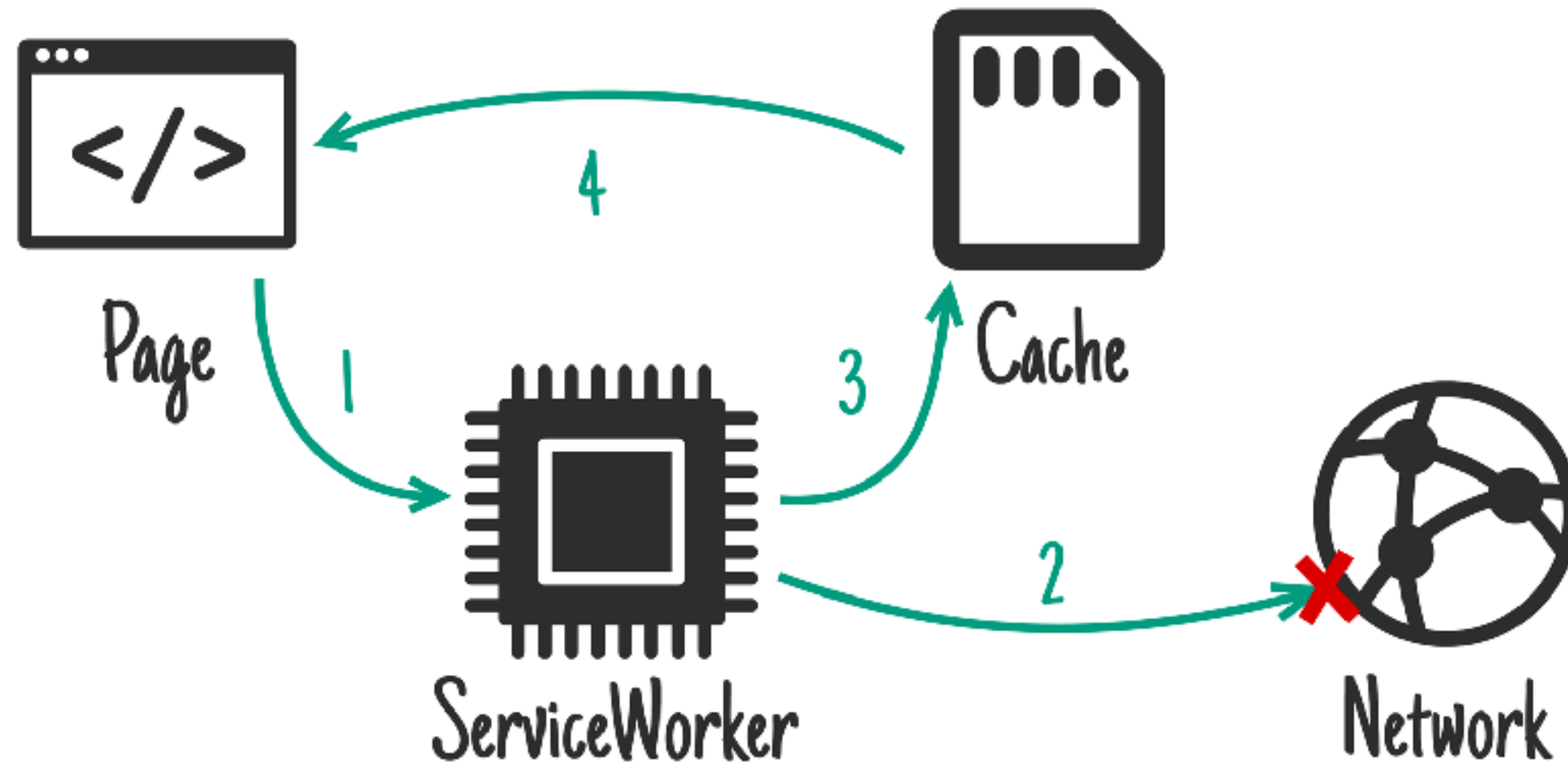


Cache, network fallback

```
self.addEventListener('fetch', function(event) {  
  event.respondWith(  
    caches.match(event.request).then(function(response) {  
      return response || fetch(event.request);  
    })  
  );  
});
```



Network, cache fallback

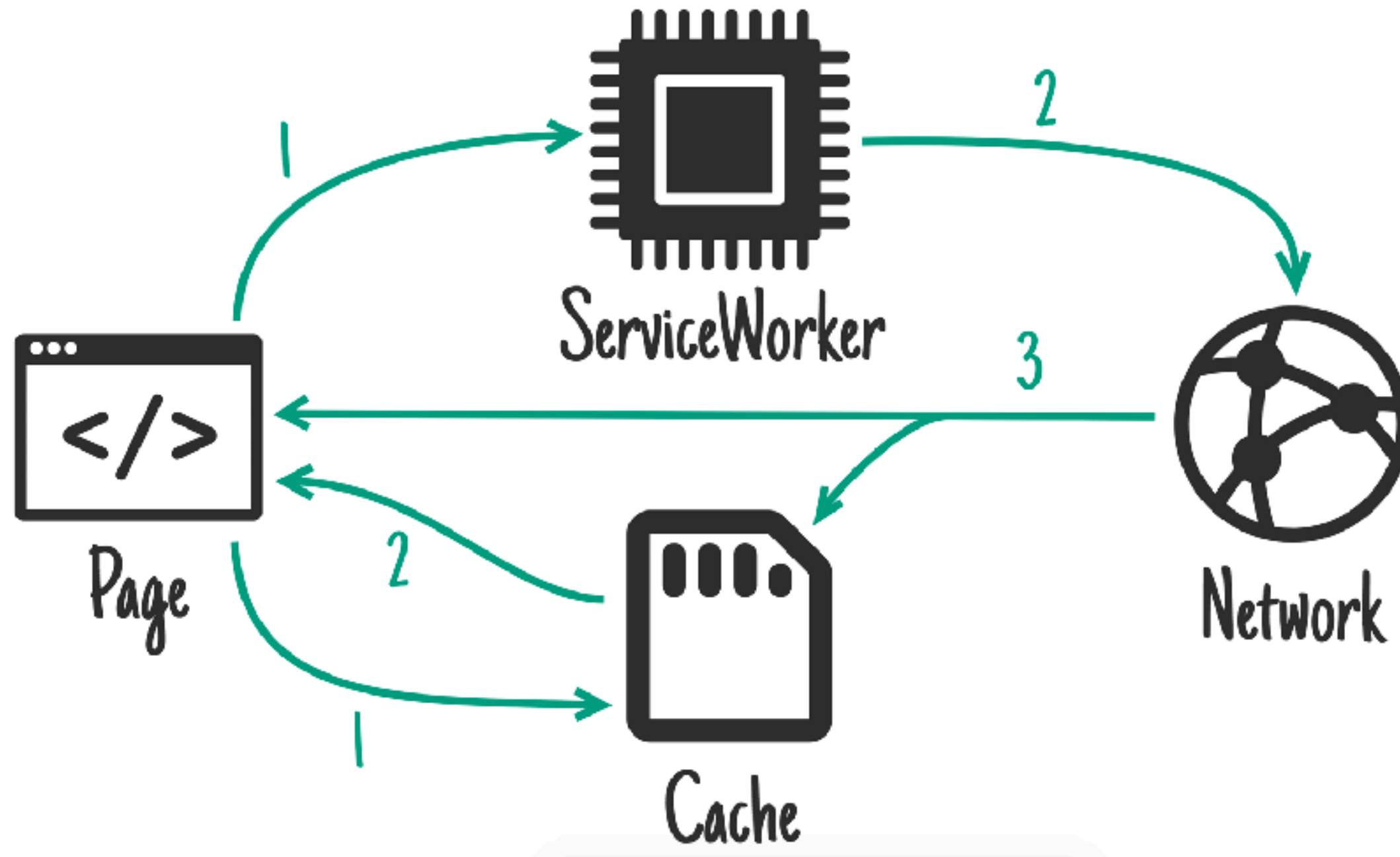


Network, cache fallback

```
self.addEventListener('fetch', function(event) {  
  event.respondWith(  
    fetch(event.request).catch(function() {  
      return caches.match(event.request);  
    })  
  );  
});
```



Cache then network fallback



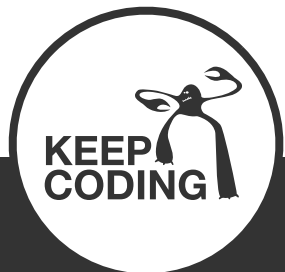
Cache then network fallback (web)

```
var networkDataReceived = false;

startSpinner();

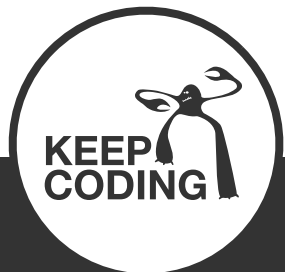
// fetch fresh data
var networkUpdate = fetch('/data.json').then(function(response) {
  return response.json();
}).then(function(data) {
  networkDataReceived = true;
  updatePage();
});

// fetch cached data
caches.match('/data.json').then(function(response) {
  if (!response) throw Error("No data");
  return response.json();
}).then(function(data) {
  // don't overwrite newer network data
  if (!networkDataReceived) {
    updatePage(data);
  }
}).catch(function() {
  // we didn't get cached data, the network is our last hope:
  return networkUpdate;
}).catch(showErrorMessage).then(stopSpinner);
```

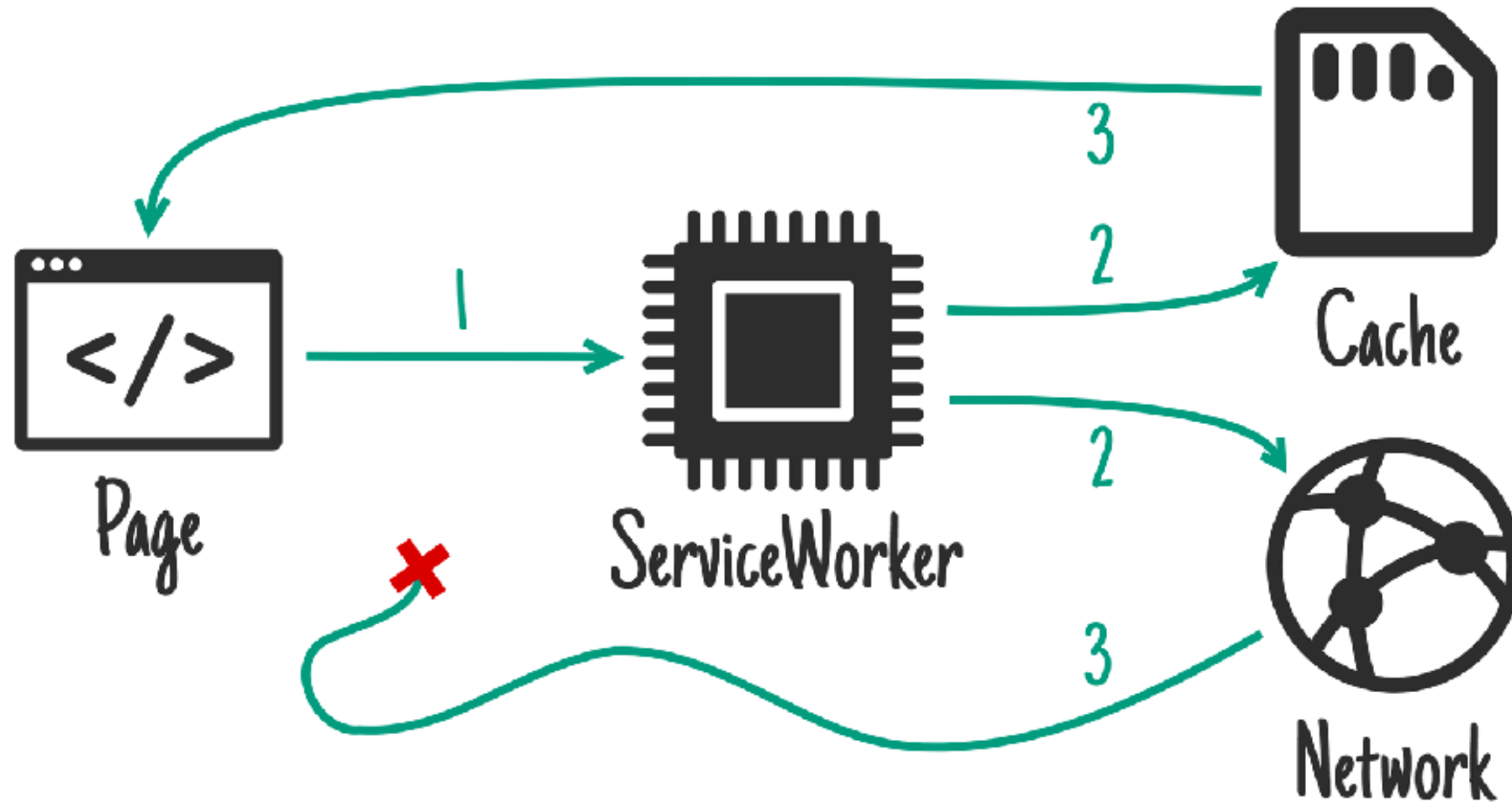


Cache then network fallback (web service)

```
self.addEventListener('fetch', function(event) {  
  event.respondWith(  
    caches.open('mysite-dynamic').then(function(cache) {  
      return fetch(event.request).then(function(response) {  
        cache.put(event.request, response.clone());  
        return response;  
      });  
    });  
  });  
});
```



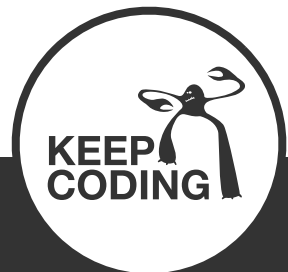
Cache, network race



Cache, network race

```
// Promise.race is no good to us because it rejects if
// a promise rejects before fulfilling. Let's make a proper
// race function:
function promiseAny(promises) {
  return new Promise((resolve, reject) => {
    // make sure promises are all promises
    promises = promises.map(p => Promise.resolve(p));
    // resolve this promise as soon as one resolves
    promises.forEach(p => p.then(resolve));
    // reject if all promises reject
    promises.reduce((a, b) => a.catch(() => b))
      .catch(() => reject(Error("All failed")));
  });
};

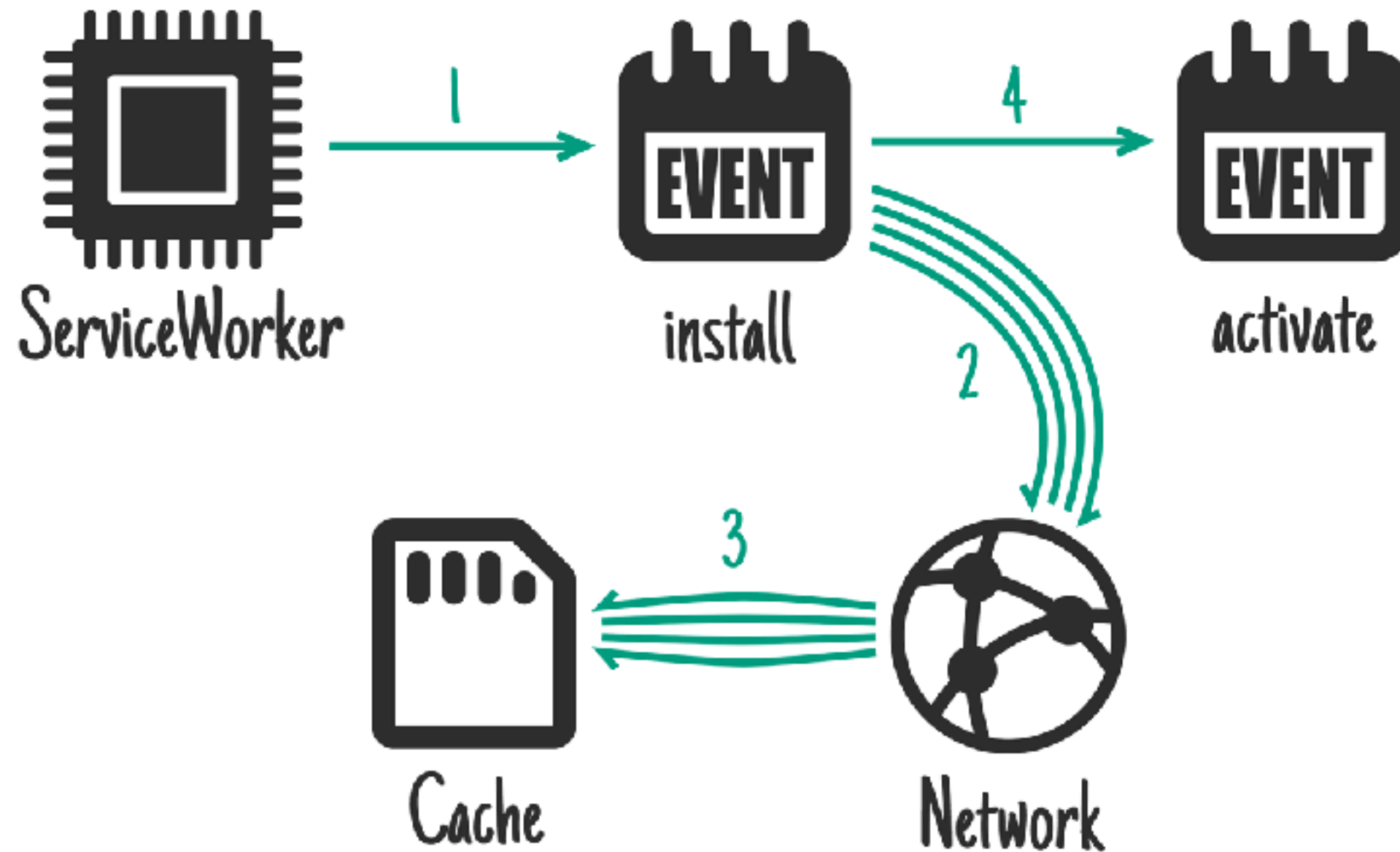
self.addEventListener('fetch', function(event) {
  event.respondWith(
    promiseAny([
      caches.match(event.request),
      fetch(event.request)
    ])
  );
});
```



Web Services as webapp Cache

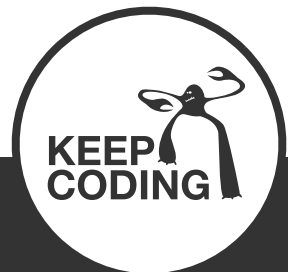


On install - como dependencia

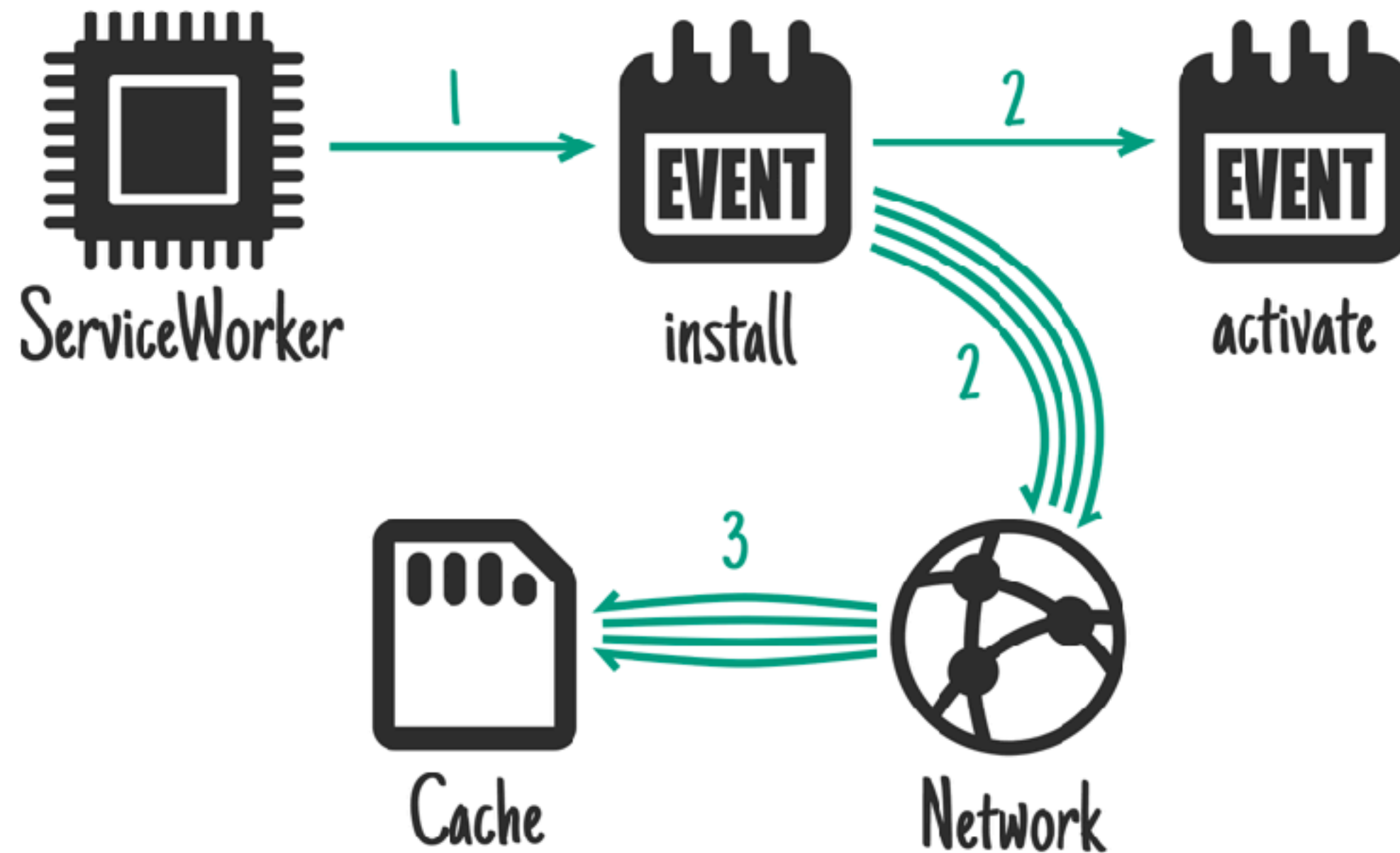


On install - como dependencia

```
self.addEventListener('install', function(event) {  
  event.waitUntil(  
    caches.open('mysite-static-v3').then(function(cache) {  
      return cache.addAll([  
        '/css/whatever-v3.css',  
        '/css/imgs/sprites-v6.png',  
        '/css/fonts/whatever-v8.woff',  
        '/js/all-min-v4.js'  
        // etc  
      ]);  
    })  
  );  
});
```

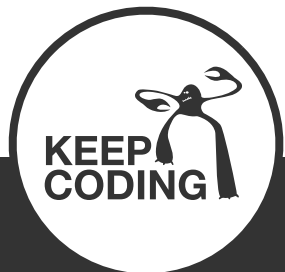


On install - no dependencia

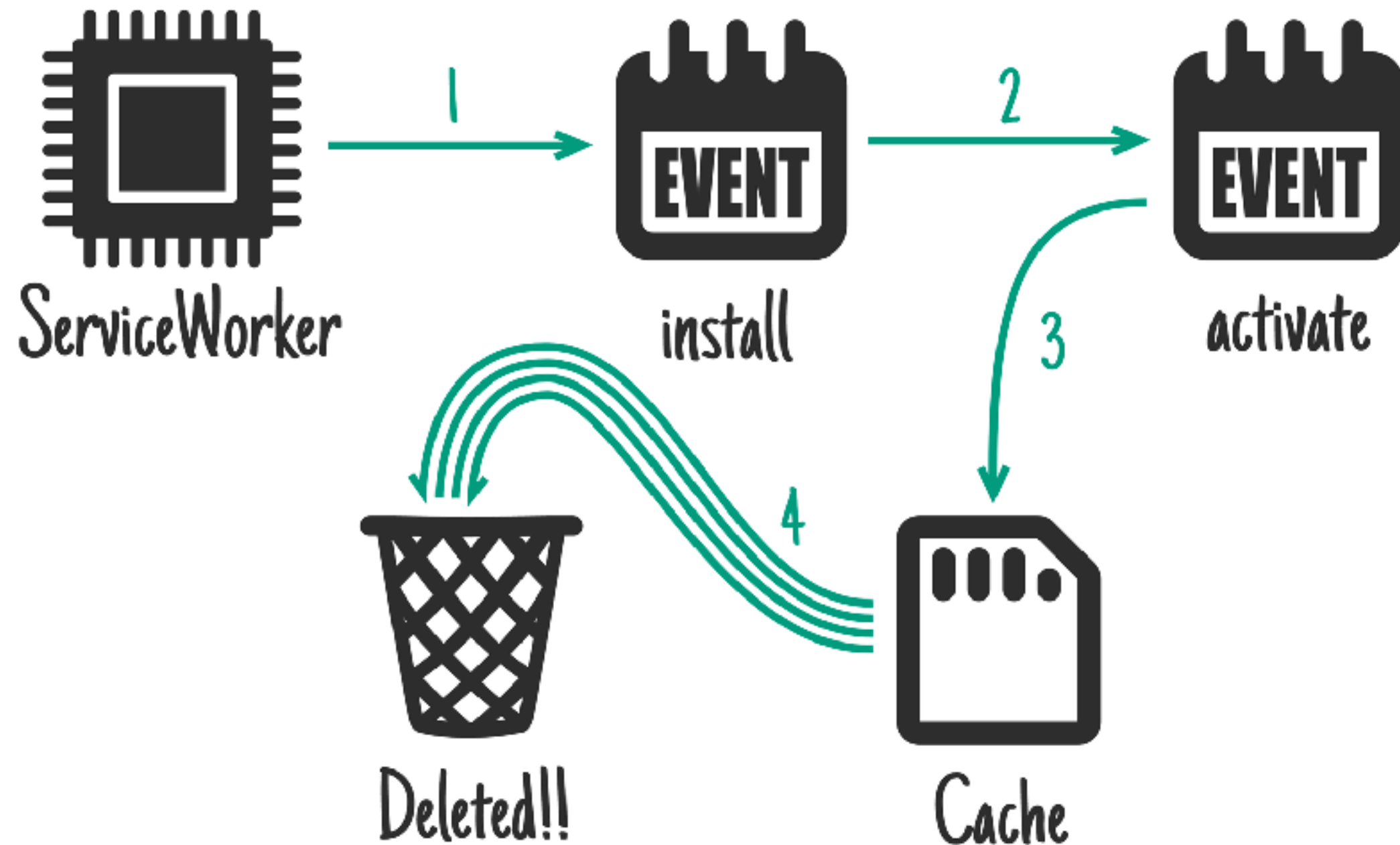


On install - como dependencia

```
self.addEventListener('install', function(event) {  
  event.waitUntil(  
    caches.open('mygame-core-v1').then(function(cache) {  
      cache.addAll(  
        // levels 11-20  
      );  
      return cache.addAll(  
        // core assets & levels 1-10  
      );  
    })  
  );  
});
```



On activate

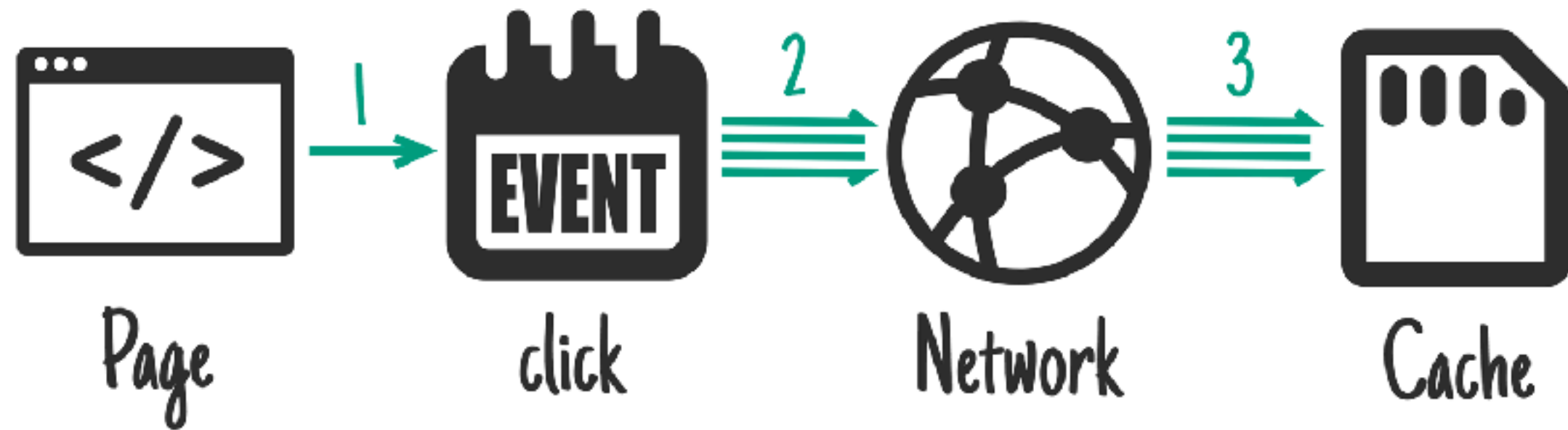


On activate

```
self.addEventListener('activate', function(event) {
  event.waitUntil(
    caches.keys().then(function(cacheNames) {
      return Promise.all(
        cacheNames.filter(function(cacheName) {
          // Return true if you want to remove this cache,
          // but remember that caches are shared across
          // the whole origin
        }).map(function(cacheName) {
          return caches.delete(cacheName);
        })
      );
    })
  );
});
```



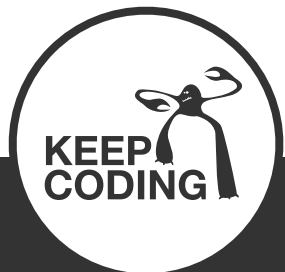
On user interaction



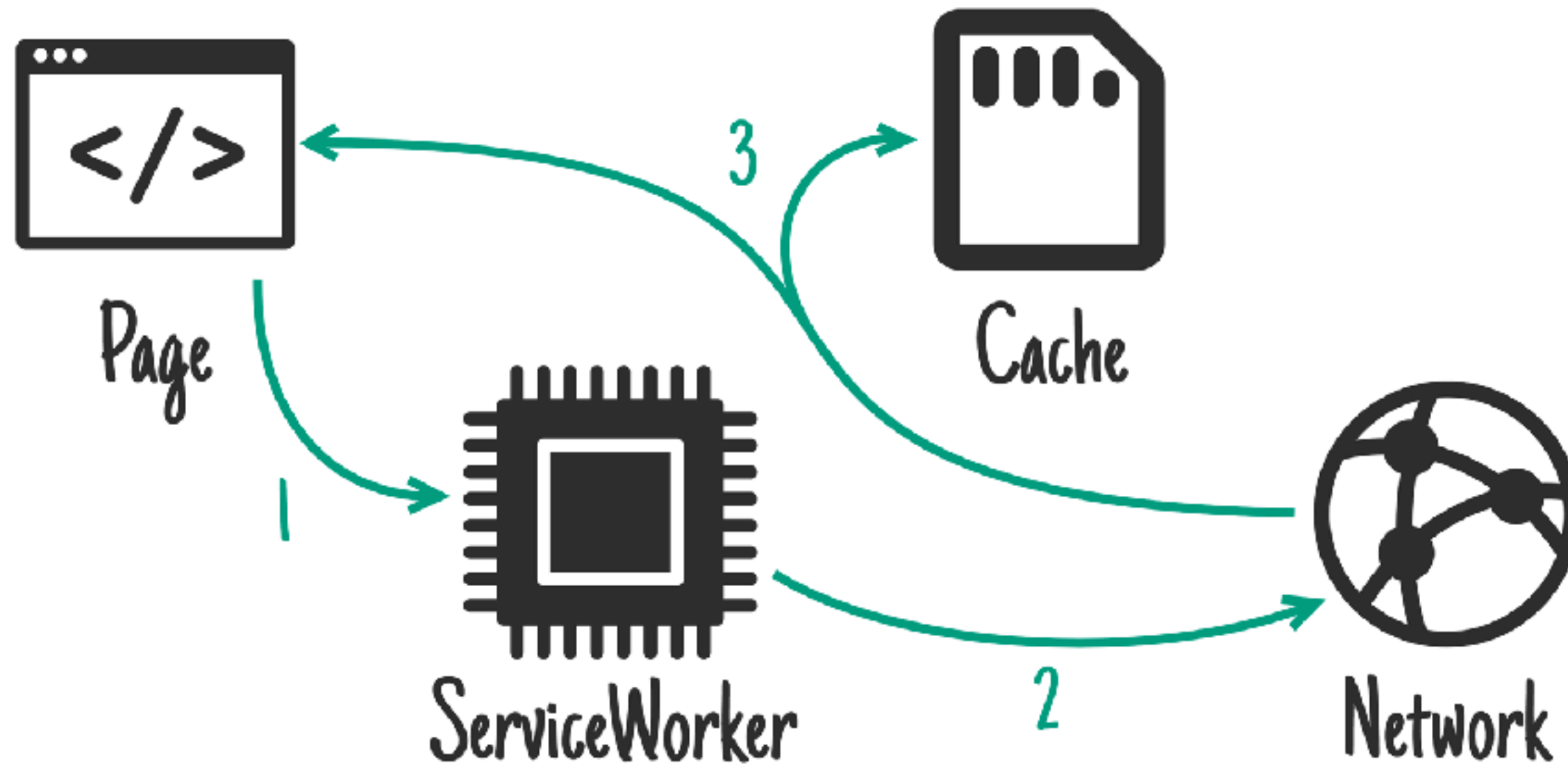
On user interaction

```
document.querySelector('.cache-article').addEventListener('click', function(event) {
    event.preventDefault();

    var id = this.dataset.articleId;
    caches.open('mysite-article-' + id).then(function(cache) {
        fetch('/get-article-urls?id=' + id).then(function(response) {
            // /get-article-urls returns a JSON-encoded array of
            // resource URLs that a given article depends on
            return response.json();
        }).then(function(urls) {
            cache.addAll(urls);
        });
    });
});
```

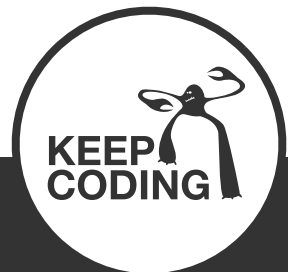


On network response

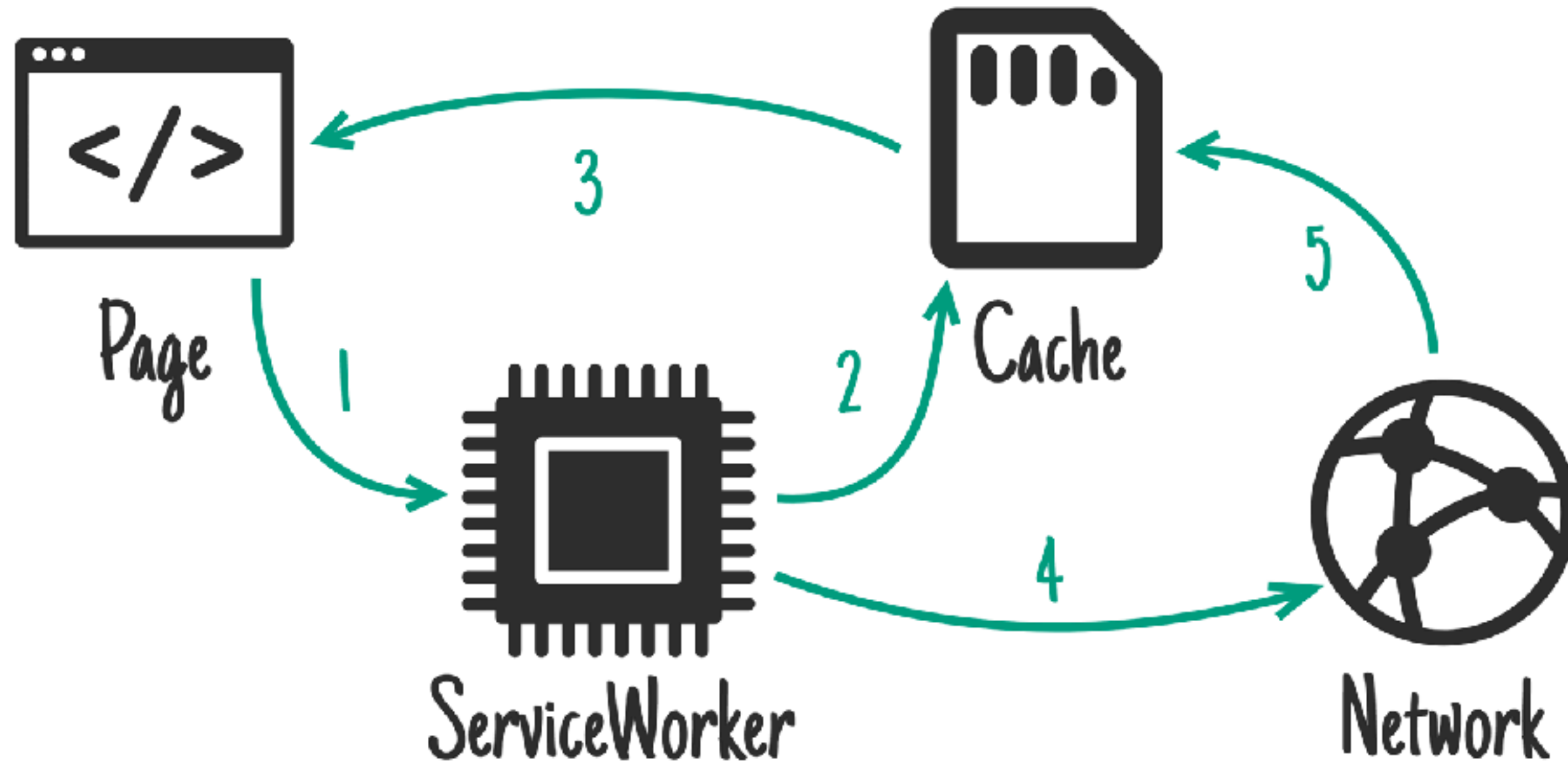


On network response

```
self.addEventListener('fetch', function(event) {
  event.respondWith(
    caches.open('mysite-dynamic').then(function(cache) {
      return cache.match(event.request).then(function (response) {
        return response || fetch(event.request).then(function(response) {
          cache.put(event.request, response.clone());
          return response;
        });
      });
    });
});
```

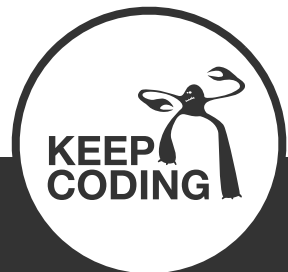


Stale-while-revalidate



Stale-while-revalidate

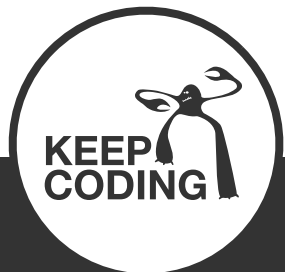
```
self.addEventListener('fetch', function(event) {
  event.respondWith(
    caches.open('mysite-dynamic').then(function(cache) {
      return cache.match(event.request).then(function(response) {
        var fetchPromise = fetch(event.request).then(function(networkResponse) {
          cache.put(event.request, networkResponse.clone());
          return networkResponse;
        })
        return response || fetchPromise;
      })
    })
  );
});
```



Persistence info

```
// From a page:
navigator.storage.requestPersistent().then(function(granted) {
  if (granted) {
    // Hurrah, your data is here to stay!

    navigator.storageQuota.queryInfo("temporary").then(function(info) {
      console.log(info.quota);
      // Result: <quota in bytes>
      console.log(info.usage);
      // Result: <used data in bytes>
    });
  }
});
```



Online / Offline



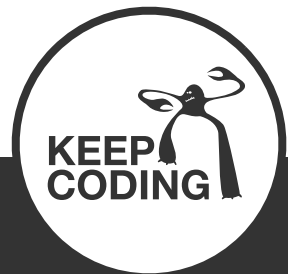
Permite reaccionar ante conexiones/desconexiones



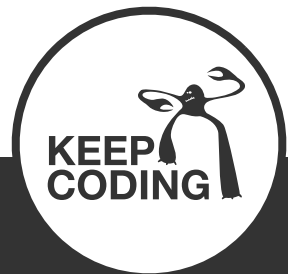
```
document.body.addEventListener("offline", function () {
    updateOnlineStatus("offline")
});
document.body.addEventListener("online", function () {
    updateOnlineStatus("online")
});
```



Drag & Drop

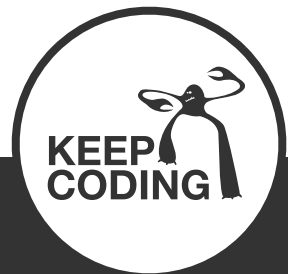


Permite implementar nativamente drag & drop



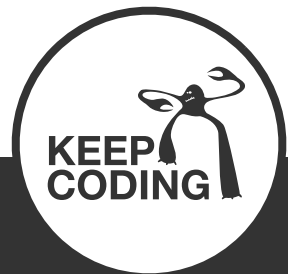
Eventos disponibles

dragstart
drag
dragenter
dragleave
dragover
drop
dragend



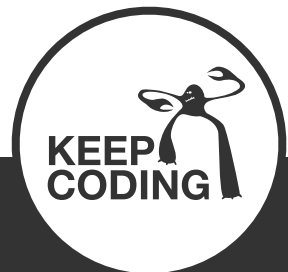
Definir elementos arrastrables

```
<div id="columns">  
  <div class="column" draggable="true"><header>A</header></div>  
  <div class="column" draggable="true"><header>B</header></div>  
  <div class="column" draggable="true"><header>C</header></div>  
</div>
```



Definir elementos arrastrables

```
#columns {  
  font-family: sans-serif;  
  display: flex;  
}  
#columns .column {  
  display: flex;  
  align-items: center;  
  align-content: center;  
  justify-content: center;  
  height: 100px;  
  width: 100px;  
  background-color: #ddd;  
  border-radius: 5px;  
  border: 2px solid black;  
  margin: 3px;  
}
```



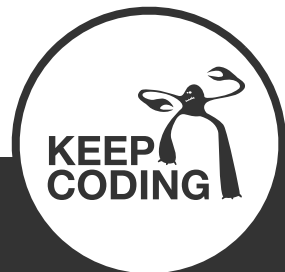
Comienzo/fin de arrastre

```
function handleDragStart(e) {  
  // this|e.target is the source node.  
  this.style.opacity = '0.4';  
}  
  
function handleDragEnd(e) {  
  // this|e.target is the source node.  
  this.style.opacity = '1';  
  [].forEach.call(cols, function (col) {  
    col.classList.remove('over');  
  });  
}  
  
var cols = document.querySelectorAll('#columns .column');  
[].forEach.call(cols, function(col) {  
  col.addEventListener('dragstart', handleDragStart, false);  
  col.addEventListener('dragend', handleDragEnd, false);  
});
```



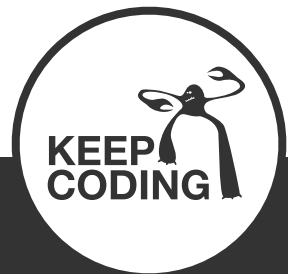
Arrastre sobre otros elementos (I)

```
function handleDragEnter(e) {  
    // this|e.target is the current hover target.  
    this.classList.add('over');  
}  
  
function handleDragLeave(e) {  
    // this|e.target is previous target element.  
    this.classList.remove('over');  
}  
  
var cols = document.querySelectorAll('#columns .column');  
[].forEach.call(cols, function(col) {  
    col.addEventListener('dragstart', handleDragStart, false);  
    col.addEventListener('dragenter', handleDragEnter, false);  
    col.addEventListener('dragleave', handleDragLeave, false);  
    col.addEventListener('dragend', handleDragEnd, false);  
});
```



Arrastre sobre otros elementos (II)

```
#columns .column.over {  
    border-style: dashed;  
}
```



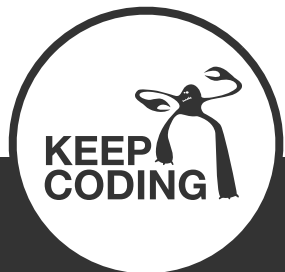
Soltar elemento arrastrado (I)

```
var dragSrcEl = null;

function handleDragStart(e) {
  // Target (this) element is the source node.
  this.style.opacity = '0.4';

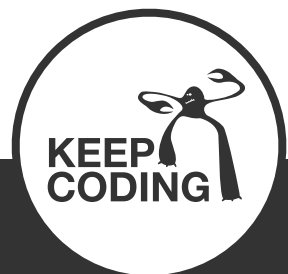
  dragSrcEl = this;

  e.dataTransfer.effectAllowed = 'move';
  e.dataTransfer.setData('text/html', this.innerHTML);
}
```



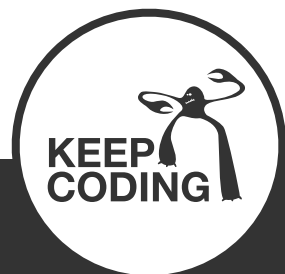
Soltar elemento arrastrado (II)

```
function handleDrop(e) {  
    // this/e.target is current target element.  
  
    if (e.stopPropagation) {  
        e.stopPropagation(); // Stops some browsers from redirecting.  
    }  
  
    // Don't do anything if dropping the same column we're dragging.  
    if (dragSrcEl !== this) {  
        // Set the source column's HTML to the HTML of the column we dropped on.  
        dragSrcEl.innerHTML = this.innerHTML;  
        this.innerHTML = e.dataTransfer.getData('text/html');  
    }  
  
    return false;  
}
```



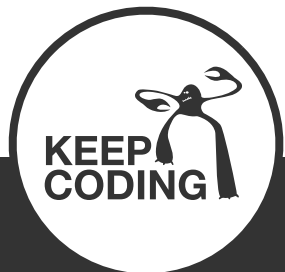
Soltar elemento arrastrado (III)

```
function handleDragOver(e) {  
  if (e.preventDefault) {  
    e.preventDefault(); // Necessary. Allows us to drop.  
  }  
  
  e.dataTransfer.dropEffect = 'move';  
  return false;  
}  
  
var cols = document.querySelectorAll('#columns .column');  
[].forEach.call(cols, function(col) {  
  col.addEventListener('dragstart', handleDragStart, false);  
  col.addEventListener('dragenter', handleDragEnter, false);  
  col.addEventListener('dragover', handleDragOver, false);  
  col.addEventListener('dragleave', handleDragLeave, false);  
  col.addEventListener('drop', handleDrop, false);  
  col.addEventListener('dragend', handleDragEnd, false);  
});
```

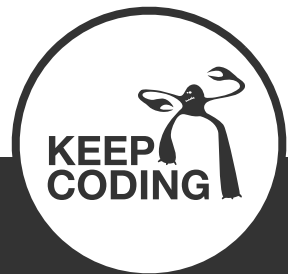


Arrastrar ficheros al navegador

```
function handleDrop(e) {  
    e.stopPropagation(); // Stops some browsers from redirecting.  
    e.preventDefault();  
  
    var files = e.dataTransfer.files;  
    for (var i = 0, f; f = files[i]; i++) {  
        // Read the File objects in this FileList.  
    }  
}
```



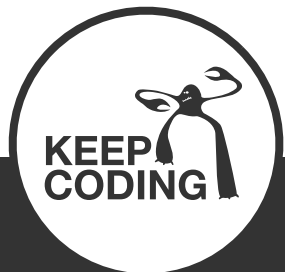
File API



Permite leer contenidos de ficheros del disco del usuario



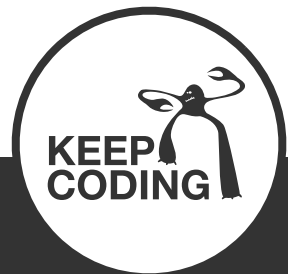
```
fileInput.addEventListener('change', function(e) {
    var display = document.getElementById('display');
    var file = document.getElementById('fileInput').files[0];
    var textType = /text.*/;
    if (file.type.match(textType)) {
        var reader = new FileReader();
        reader.onload = function(e) {
            display.innerText = reader.result;
        }
        reader.readAsText(file); // como texto
        reader.readAsDataURL(file); // devuelve datos en base64
        reader.readAsBinaryString(file); // datos binarios
    } else {
        display.innerText = "Archivo no soportado!"
    }
});
```



FormData



Permite simular un formulario para enviarlo por AJAX.
De esta manera podemos subir archivos...¡por AJAX!



```
var formData = new FormData();

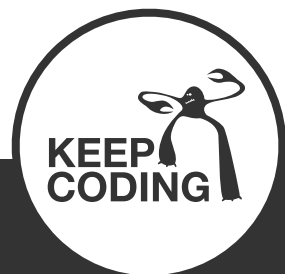
formData.append("username", "Groucho");
formData.append("accountnum", 123456); // number 123456 is immediately converted to string "123456"

// HTML file input user's choice...
formData.append("userfile", fileInputElement.files[0]);

// JavaScript file-like object...
var content = '<a id="a"><b id="b">hey!</b></a>'; // the body of the new file...
var blob = new Blob([content], { type: "text/xml" });

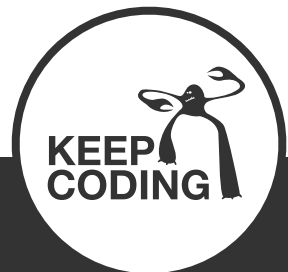
formData.append("webmasterfile", blob);

var request = new XMLHttpRequest();
request.open("POST", "http://foo.com/submitform.php");
request.send(formData);
```

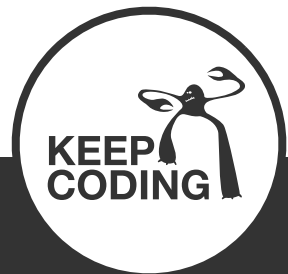


```
$.ajax({  
  url: "stash.php",  
  type: "POST",  
  data: formData,  
  processData: false, // tell jQuery not to process the data  
  contentType: false // tell jQuery not to set contentType  
});
```

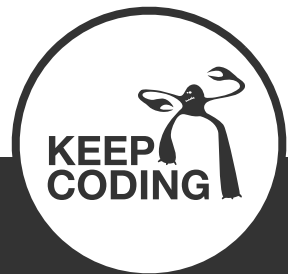
Si queremos usar jQuery, hay que indicarle que no procese los datos
ni que asigne ContentType



Notificaciones de escritorio



Permite enviar notificaciones de escritorio




```
document.getElementById('notify').addEventListener('click', function(){
    if(! ('Notification' in window) ){
        alert('Notificaciones no soportadas');
        return;
    }
    Notification.requestPermission(function(permission){
        var notification = new Notification(
            'Hello, hello',
            {
                body: 'Hola!',
                icon: 'vertigo.png'
            }
        );
    });
});
```

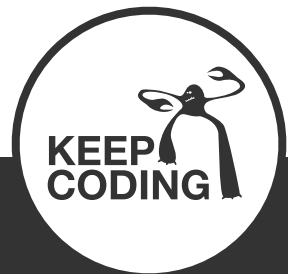


IndexDB



IndexDB:

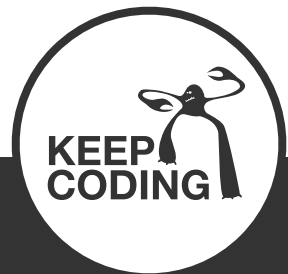
- Base de datos embebida en la web
- Almacenamiento en local
- Información estructurada basada en contenedores|colecciones|objetos
- Múltiples bases de datos



Crear la base de datos

```
var indexedDB = window.indexedDB || window.mozIndexedDB ||  
window.webkitIndexedDB || window.msIndexedDB;
```

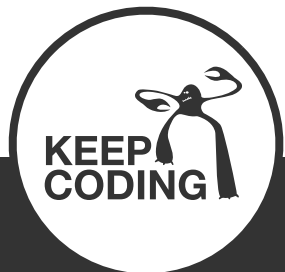
```
var request = indexedDB.open("object", 1);  
var db;
```



Capturar el resultado de la creación

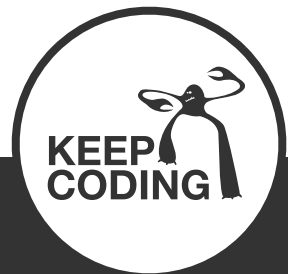
```
request.onsuccess = function (e) {  
    console.log('Base de datos cargada correctamente');  
    db = request.result;  
};
```

```
request.onerror = function (e) {  
    console.log('Error cargando la base de datos');  
};
```



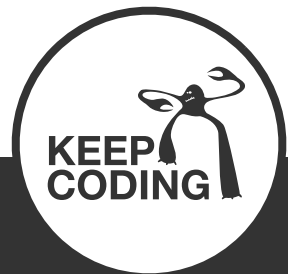
Inicializar la base de datos

```
request.onupgradeneeded = function (e) {  
    var db = event.target.result;  
  
    var objectStore = db.createObjectStore("people", {  
        keyPath : 'id',  
        autoIncrement : true  
    });  
    objectStore.createIndex('by_name', 'name', { unique : false });  
    objectStore.createIndex('by_dni', 'dni', { unique : true });  
};
```



Añadir elementos a una colección (I)

```
request.onsuccess = function (e) {  
  console.log('Base de datos cargada correctamente');  
  db = request.result;  
  
  fetch('https://jsonplaceholder.typicode.com/users').then(function (response) {  
    return response.json();  
  }).then(saveElements);  
};
```



Añadir elementos a una colección (II)

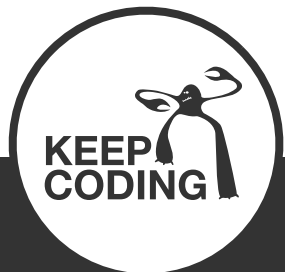
```
function saveElements(json) {
  people = json;

  var transaction = db.transaction("people", "readwrite");

  transaction.oncomplete = function (event) {
    alert("All done!");
  };

  transaction.onerror = function (event) {
    // handle errors!
  };

  var peopleObjectStore = transaction.objectStore("people");
  for (var i in people) {
    var addResponse = peopleObjectStore.add(people[i]);
    addResponse.onsuccess = (function (i) {
      return function () {
        console.log('added', i);
      }
    })(i);
  }
}
```



Obtener elementos a una colección

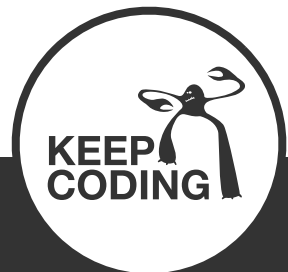
```
var transaction = db.transaction('people');
var objectStore = transaction.objectStore("people");
var request = objectStore.get(1);
request.onerror = function (event) {
    // Handle errors!
};
request.onsuccess = function (event) {
    // Do something with the request.result!
    console.log("Name for SSN 444-44-4444 is " + request.result.name);
};
```



Consultas a una colección (I)

```
var index = objectStore.index("by_name");
index.get("Ervin Howell").onsuccess = function(event) {
    alert("Donna's email is " + event.target.result.email);
};

// Using a normal cursor to grab whole people record objects
index.openCursor().onsuccess = function(event) {
    var cursor = event.target.result;
    if (cursor) {
        alert("Name: " + cursor.key +
            ", SSN: " + cursor.value.ssn + ", email: " + cursor.value.email);
        cursor.continue();
    }
};
```



Consultas a una colección (II)

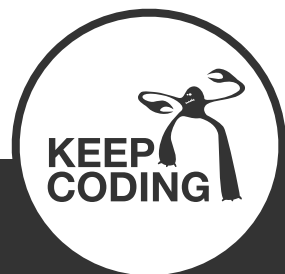
```
// Only match "Donna"
var singleKeyRange = IDBKeyRange.only("Donna");

// Match anything past "Bill", including "Bill"
var lowerBoundKeyRange = IDBKeyRange.lowerBound("Bill");

// Match anything past "Bill", but don't include "Bill"
var lowerBoundOpenKeyRange = IDBKeyRange.lowerBound("Bill", true);

// Match anything up to, but not including, "Donna"
var upperBoundOpenKeyRange = IDBKeyRange.upperBound("Donna", true);

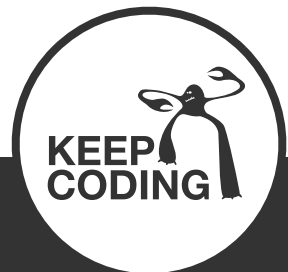
// Match anything between "Bill" and "Donna", but not including "Donna"
var boundKeyRange = IDBKeyRange.bound("Bill", "Donna", false, true);
```



Consultas a una colección (III)

```
objectStore.openCursor(boundKeyRange, "prev").onsuccess = function(event) {  
    var cursor = event.target.result;  
    if (cursor) {  
        // Do something with the entries.  
        cursor.continue();  
    }  
};
```

```
index.openKeyCursor(null, "nextunique").onsuccess = function(event) {  
    var cursor = event.target.result;  
    if (cursor) {  
        // Do something with the entries.  
        cursor.continue();  
    }  
};
```

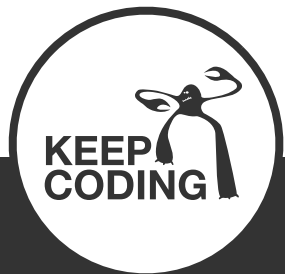


Actualizar elementos de una colección

```
var objectStore = db.transaction(["people"], "readwrite").objectStore("people");
var request = objectStore.get(1);
request.onerror = function(event) {
    // Handle errors!
};
request.onsuccess = function(event) {
    // Get the old value that we want to update
    var data = request.result;

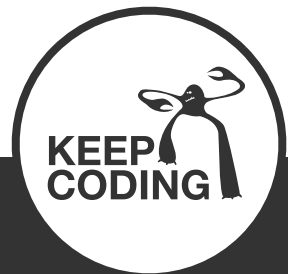
    // update the value(s) in the object that you want to change
    data.age = 42;

    // Put this updated object back into the database.
    var requestUpdate = objectStore.put(data);
    requestUpdate.onerror = function(event) {
        // Do something with the error
    };
    requestUpdate.onsuccess = function(event) {
        // Success - the data is updated!
    };
};
```



Eliminar elementos a una colección

```
var transaction = db.transaction(["people"], "readwrite");  
  
var request = transaction.delete(1);  
request.onsuccess = function(event) {  
    // It's gone!  
};
```

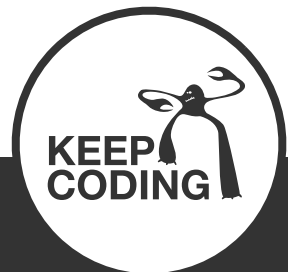


Librerías basadas en IndexedDB

<http://dexie.org/>

<https://github.com/erikolson186/zangodb>

<https://github.com/localForage/localForage>





GRACIAS
www.keepcoding.io

