

# Frontend Avanzado

Kevin Martínez Javascript Developer

(@kjmesec)

kevinjmartinez/



# Requisitos previos

Conocimientos medios de Javascript

Tener git Instalado y conocimientos básicos. <https://git-scm.com/downloads>



# Repositorios

<https://github.com/kevinccbsg/web-bootcamp-2019-frontend>

<https://gitlab.keepcoding.io/keepcoding-bootcamps/full-stack-web-bootcamp-iv/desarrollo-frontend-con-javascript>

## Link de la aplicación

<https://tvmaze-keepcoding-2019.herokuapp.com/>

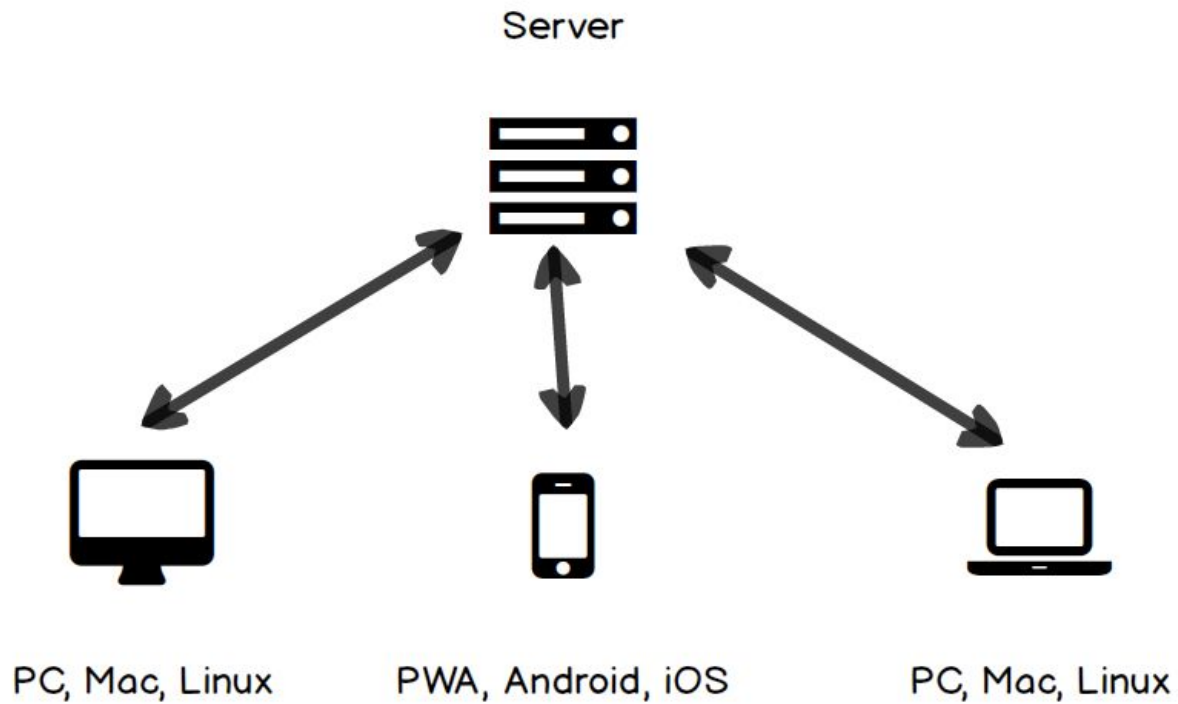


# Client-server model



*Client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.*





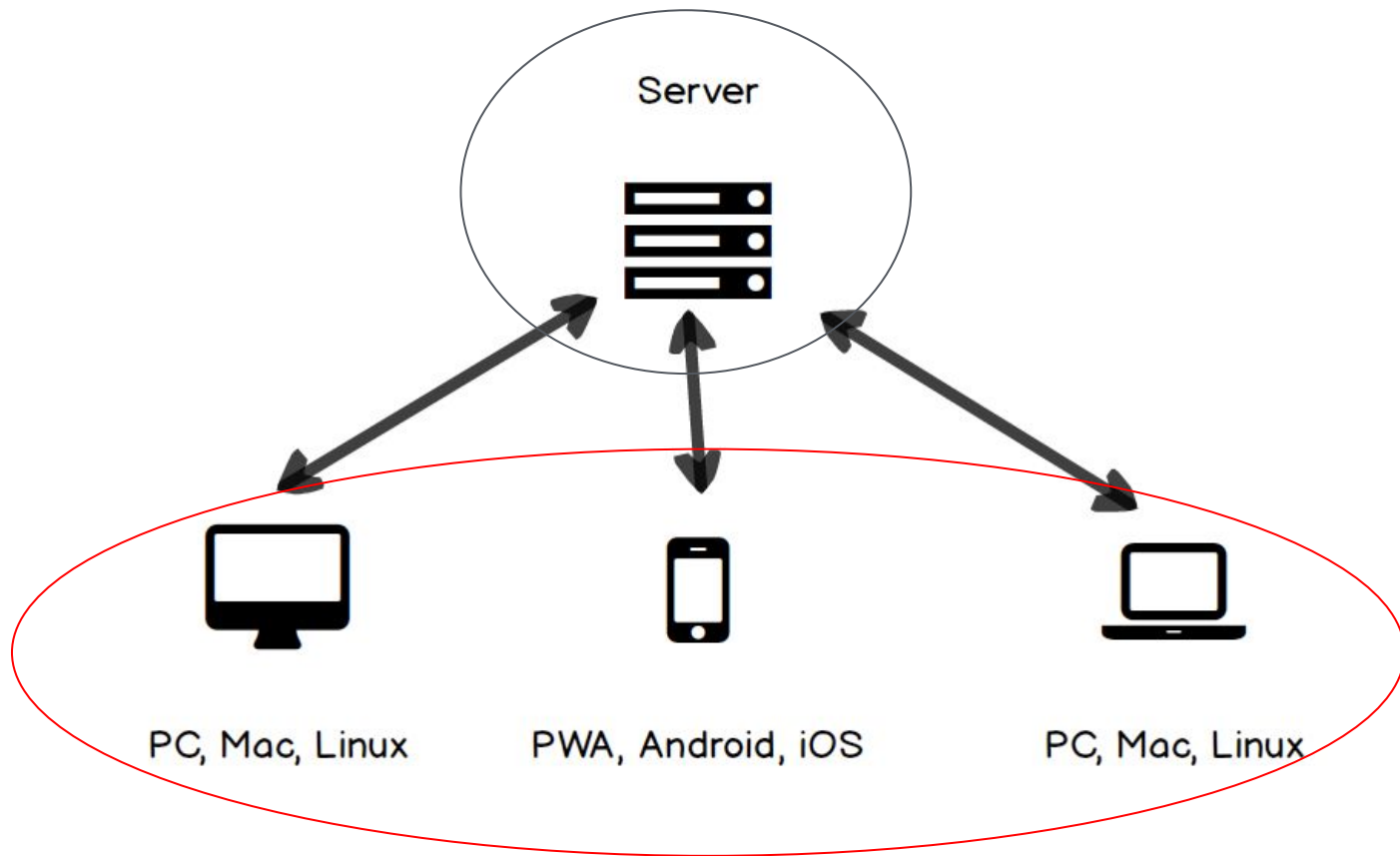
# Frontend Development



*Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly.*









# Javascript



*High-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.*

<https://github.com/tc39/ecma262>

<https://babeljs.io/>



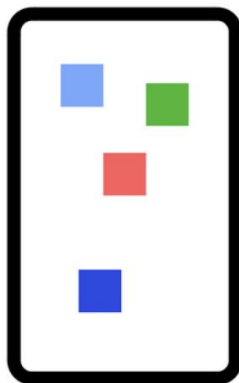
# ¿Como funciona Javascript?

<https://blog.sessionstack.com/how-does-javascript-actually-work-part-1-b0bacc073cf>



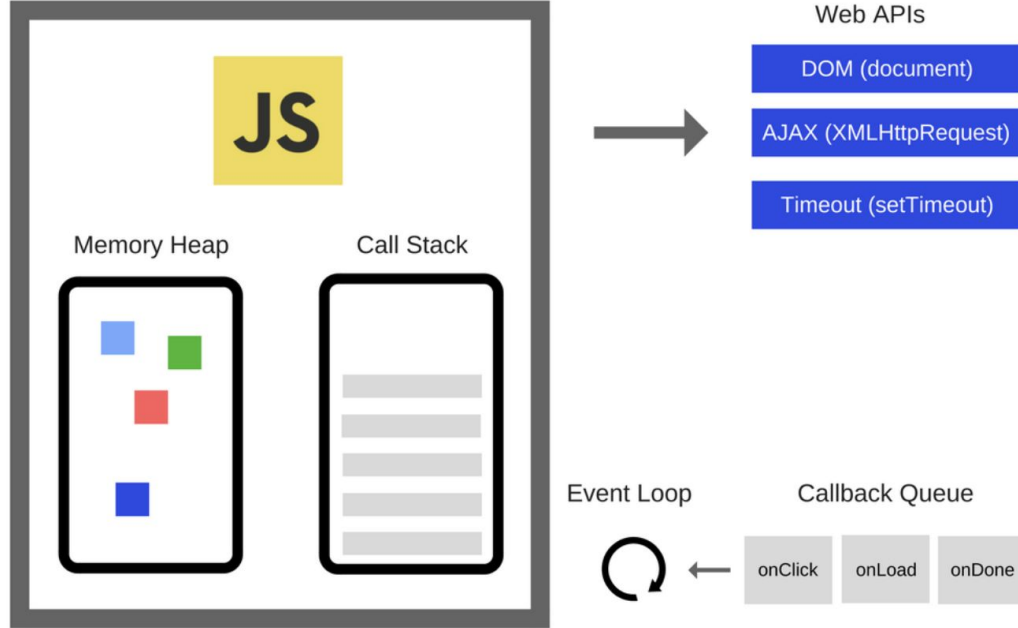


Memory Heap



Call Stack

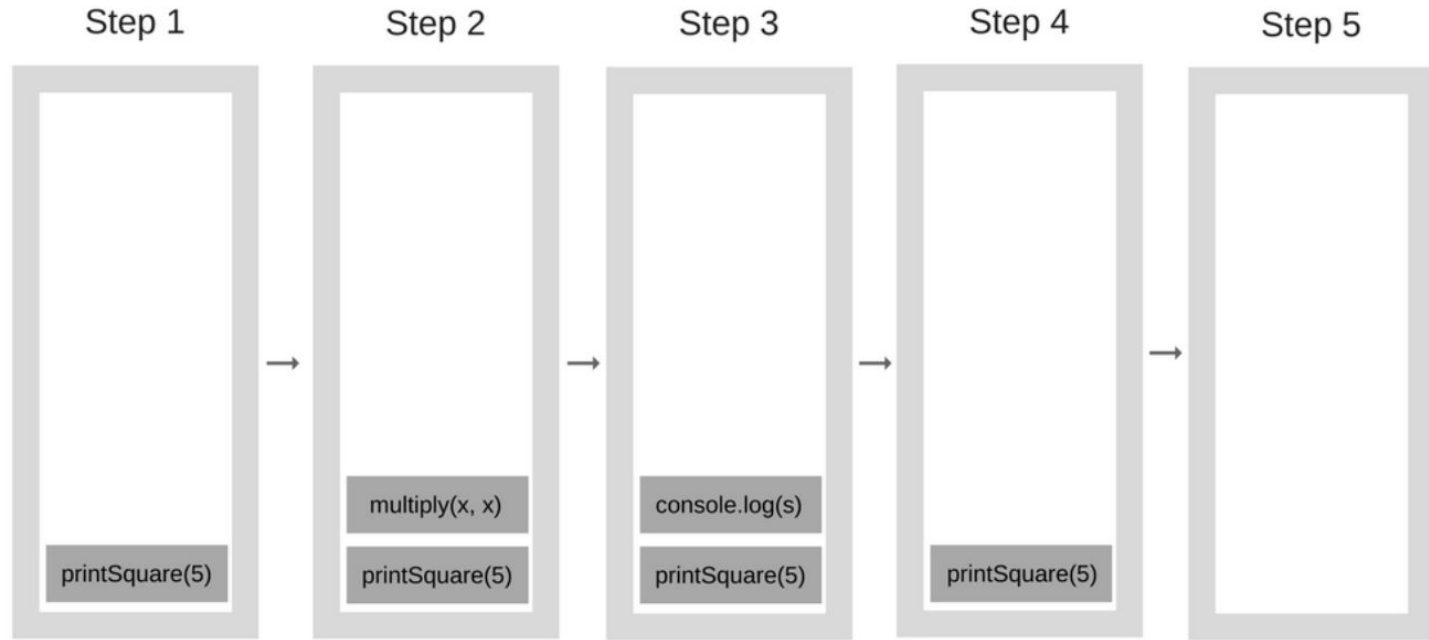




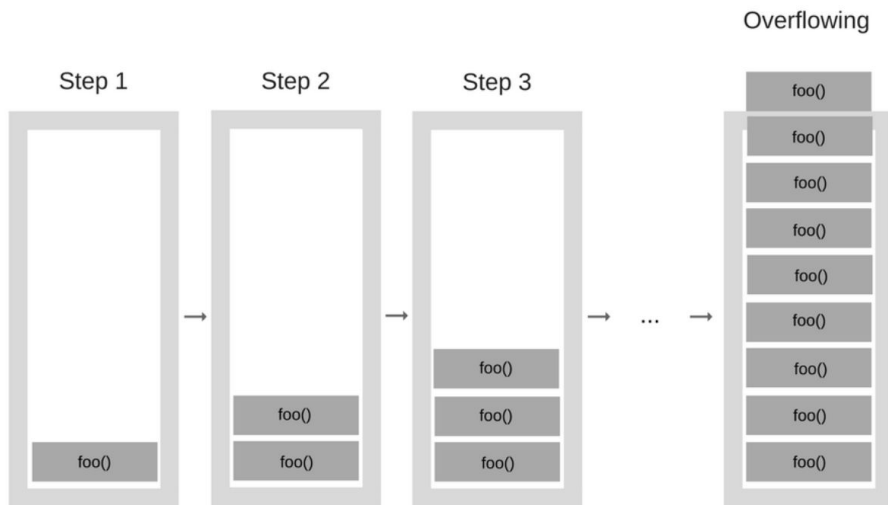
```
function multiply(x, y) {  
    return x * y;  
}  
  
function printSquare(x) {  
    var s = multiply(x, x);  
    console.log(s);  
}  
  
printSquare(5);
```







```
function foo() {  
    foo();  
}  
  
foo();
```



✖ ▶ Uncaught RangeError: Maximum call stack size exceeded



# Conceptos principales

## Memory Heap

Donde la memoria se aloja

## Call Stack

Donde se apila la ejecución de nuestro código.

## Web APIS

No vienen dadas por el motor sino que las da el DOM.

## Event Loop

Monitoriza el call stack y el callback queue. Si el primero está vacío toma el primer evento del callback queue y lo manda al call stack

## Callback Queue

Donde se almacenan las ejecuciones de las Web APIs asíncronas



# Creando proyecto



# Conceptos del día día



# Arrays, Object mutability, Destructuring, Arrow functions, Rest operator



# Preguntas de entrevistas



# Variables





# Context



# Arrow function Context



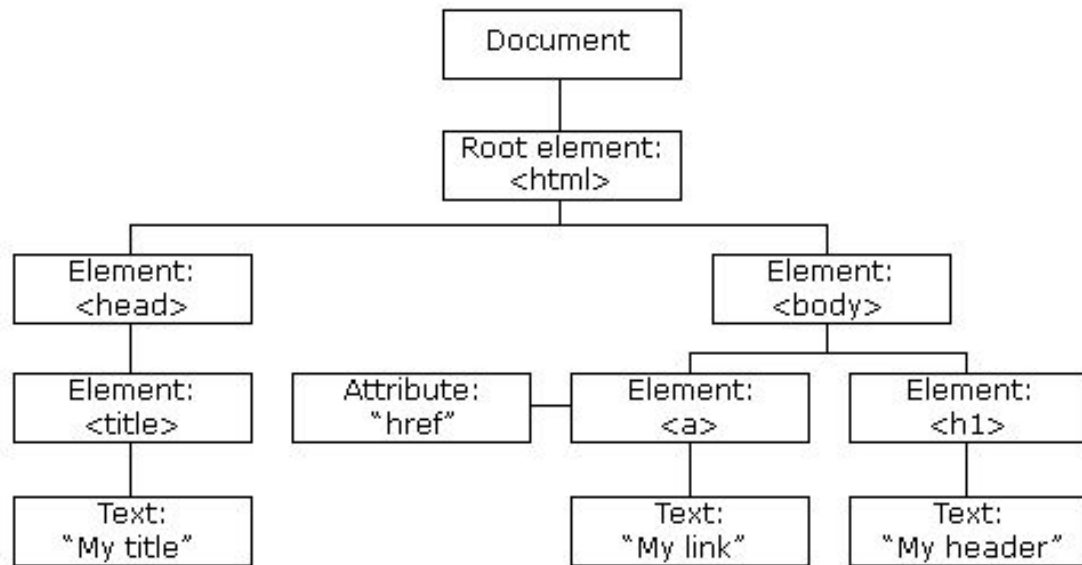
# Que es el DOM

(Document Object Model)



*The Document Object Model is a cross-platform and language-independent interface that treats an XML or HTML document as a tree structure wherein each node is an object representing a part of the document*



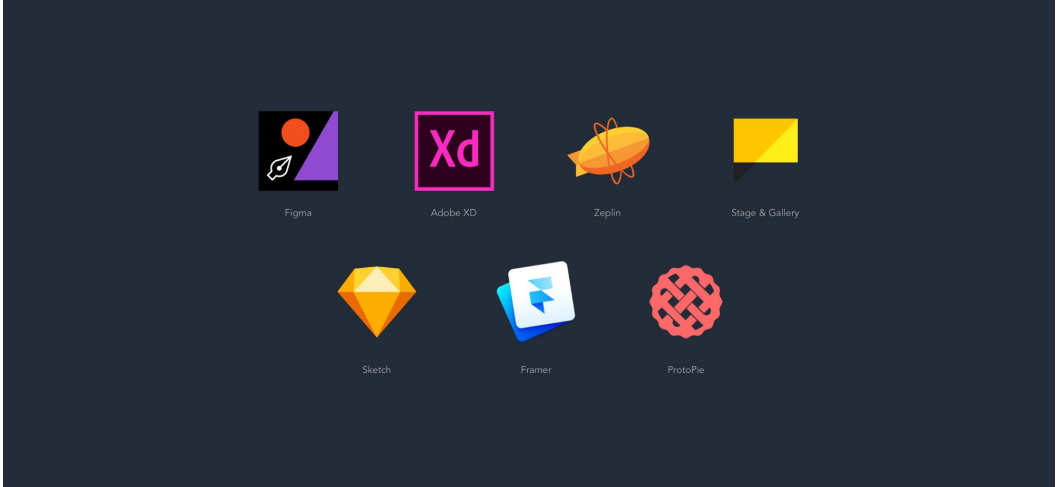


# Primeros pasos de TVMaze



# Developers vs Designers







# Mobile First y Alcance!!!!



# Consejos de CSS





# Variables

# Sprites

# Animaciones

# Selectores para navegadores



# Links útiles

<https://caniuse.com/>

<http://pleeease.io/play/>

<https://cssgridgarden.com/>

<https://flexboxfroggy.com/>

<https://coolors.co/>

<https://colorhunt.co/>

<https://material.io/resources/color/#!/?view.left=0&view.right=0>

<https://cantunsee.space/>

<https://www.cssdesignawards.com/>

<https://css-tricks.com/guide-svg-animations-smil/>

<https://css-tricks.com/using-svg/>

<https://modernizr.com/download>



# Imágenes Responsive



# Por fin, Javascript para el Front



# DOM con Javascript



**document.<Selector>**

**getElementById**

**querySelector**

**getElementsByClassName**

**querySelectorAll**

**getElementsByTagName**





# Eventos en Javascript



# Manejo de CSS con Javascript



# Event bubbling



# CB/Then/Async/Await



# Maquetando nuestra web



# Request a una API



```
<!DOCTYPE html>
<html>
<body>
<h1>The XMLHttpRequest Object</h1>

<button type="button" onclick="loadDoc()">Request data</button>

<p id="demo"></p>

<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.responseText;
    }
  };
  xhttp.open("POST", "demo_post2.asp", true);
  xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  xhttp.send("fname=Henry&lname=Ford");
}
</script>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $.ajax({url: "demo_test.txt", success: function(result){
            $("#div1").html(result);
        }});
    });
});
</script>
</head>
<body>

<div id="div1"><h2>Let jQuery AJAX Change This Text</h2></div>

<button>Get External Content</button>

</body>
</html>
```





# Axios

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



# Fetch

<https://caniuse.com/#feat=fetch>



# Fetch

```
fetch('./api/some.json')
  .then(
    function(response) {
      if (response.status !== 200) {
        console.log('Looks like there was a problem. Status Code: ' +
          response.status);
        return;
      }

      // Examine the text in the response
      response.json().then(function(data) {
        console.log(data);
      });
    }
  )
  .catch(function(err) {
    console.log('Fetch Error :-S', err);
  });
```

Nativo

Promised based

Doble procesado

Multiples respuestas

Mejor que axios?



**¿Que es un polyfill?**



# Configurando un Linter



# Intersection Observer



# Un rato en nuestras Devtools



# Despliegue con Heroku





# Service workers

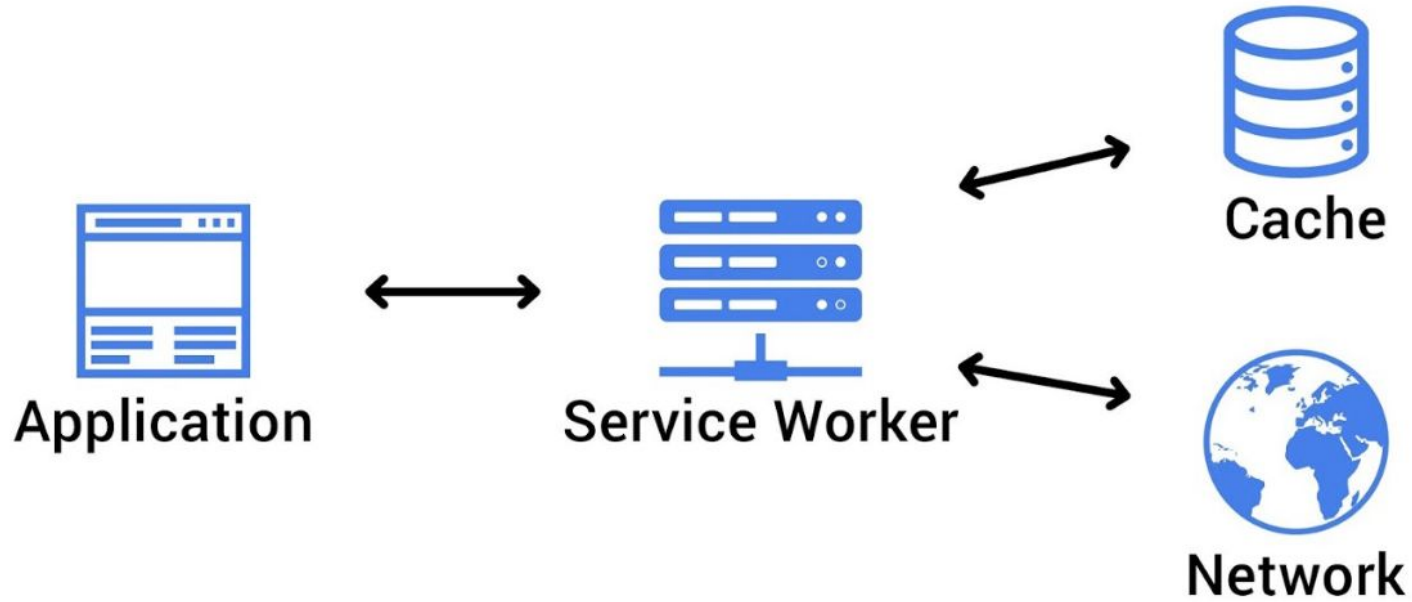


# Service workers



*A service worker is a script that your browser runs in the background, separate from a web page, opening the door to features that don't need a web page or user interaction. Today, they already include features like push notifications and background sync.*





```
if('serviceWorker' in navigator) {  
  navigator.serviceWorker.register('/sw.js')  
    .then(() => {  
      console.log("Service Worker Registered");  
    })  
}
```



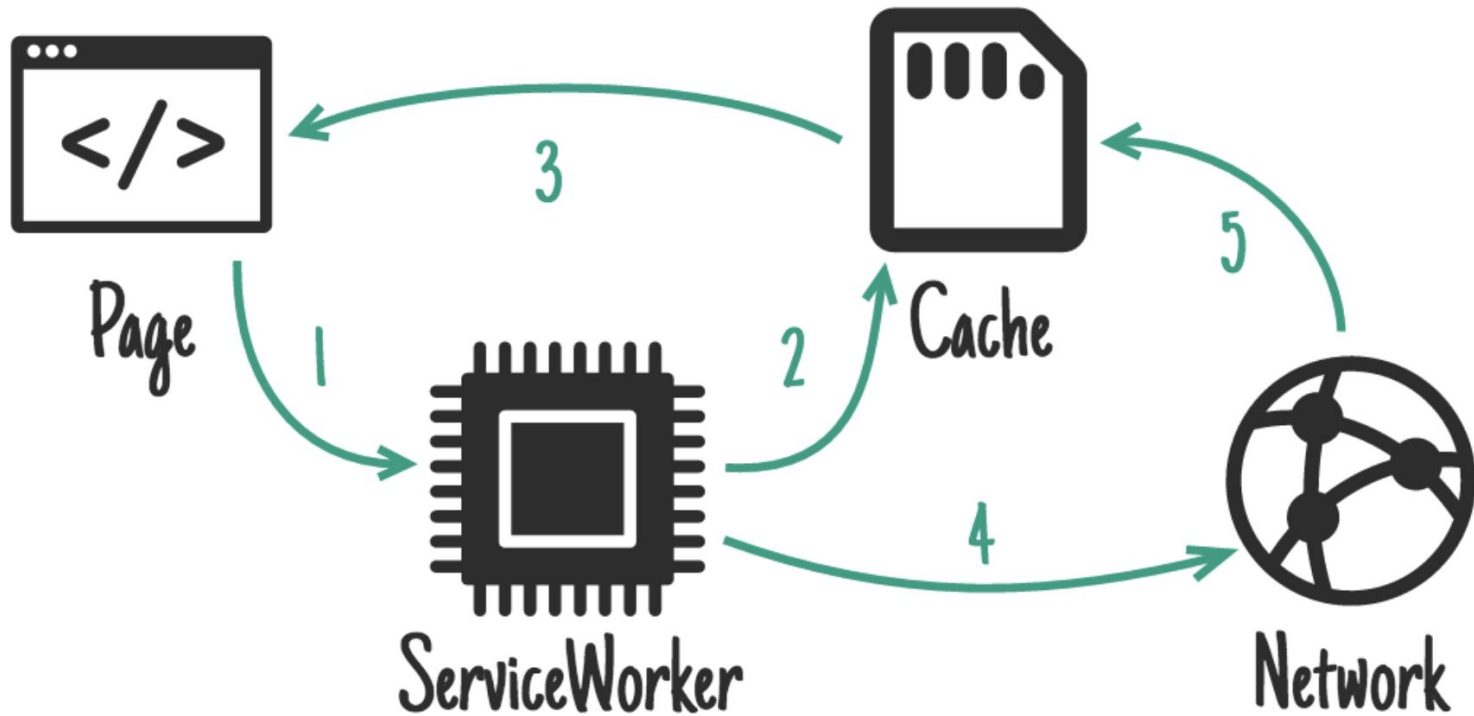
# Workbox



*Workbox is a library that bakes in a set of best practices and removes the boilerplate every developer writes when working with service workers.*

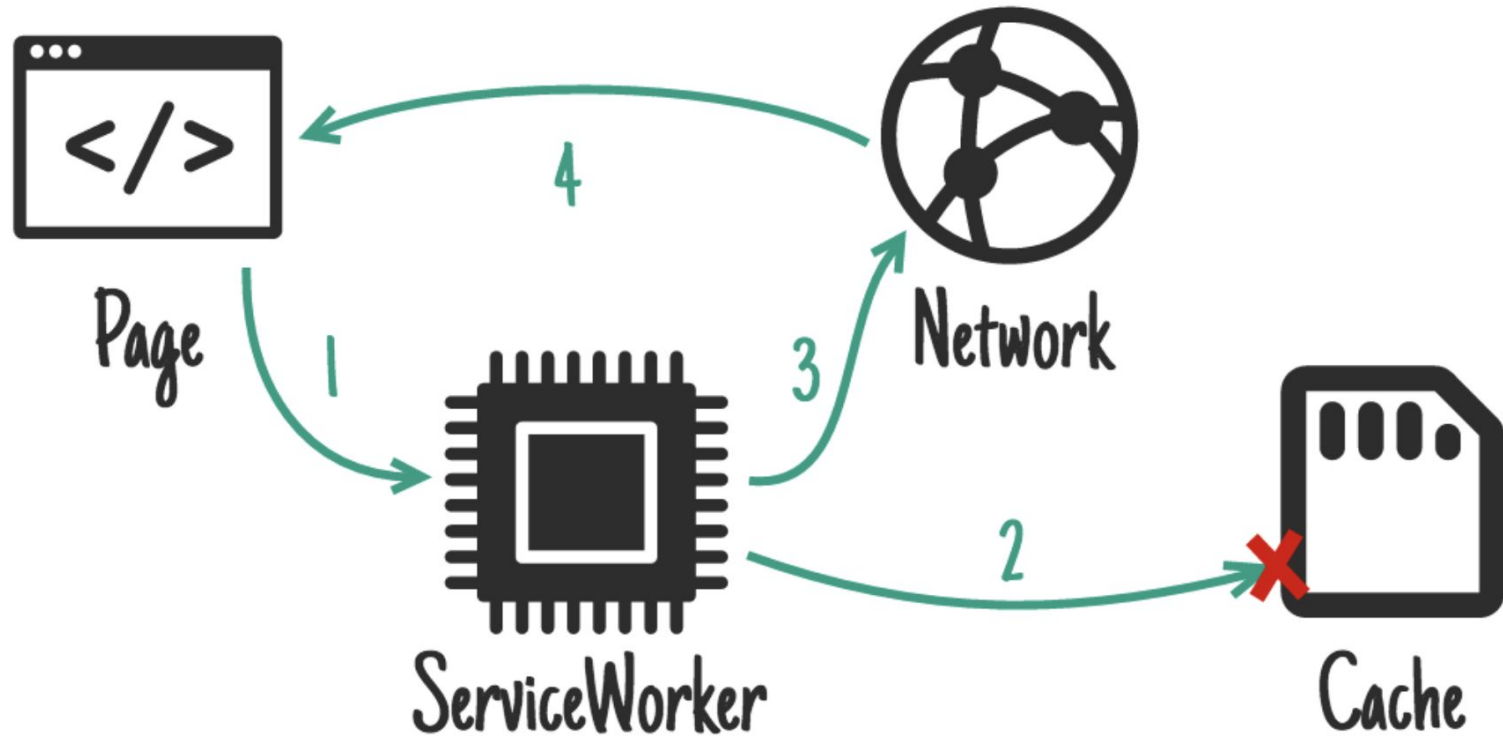


## Stale-While-Revalidate

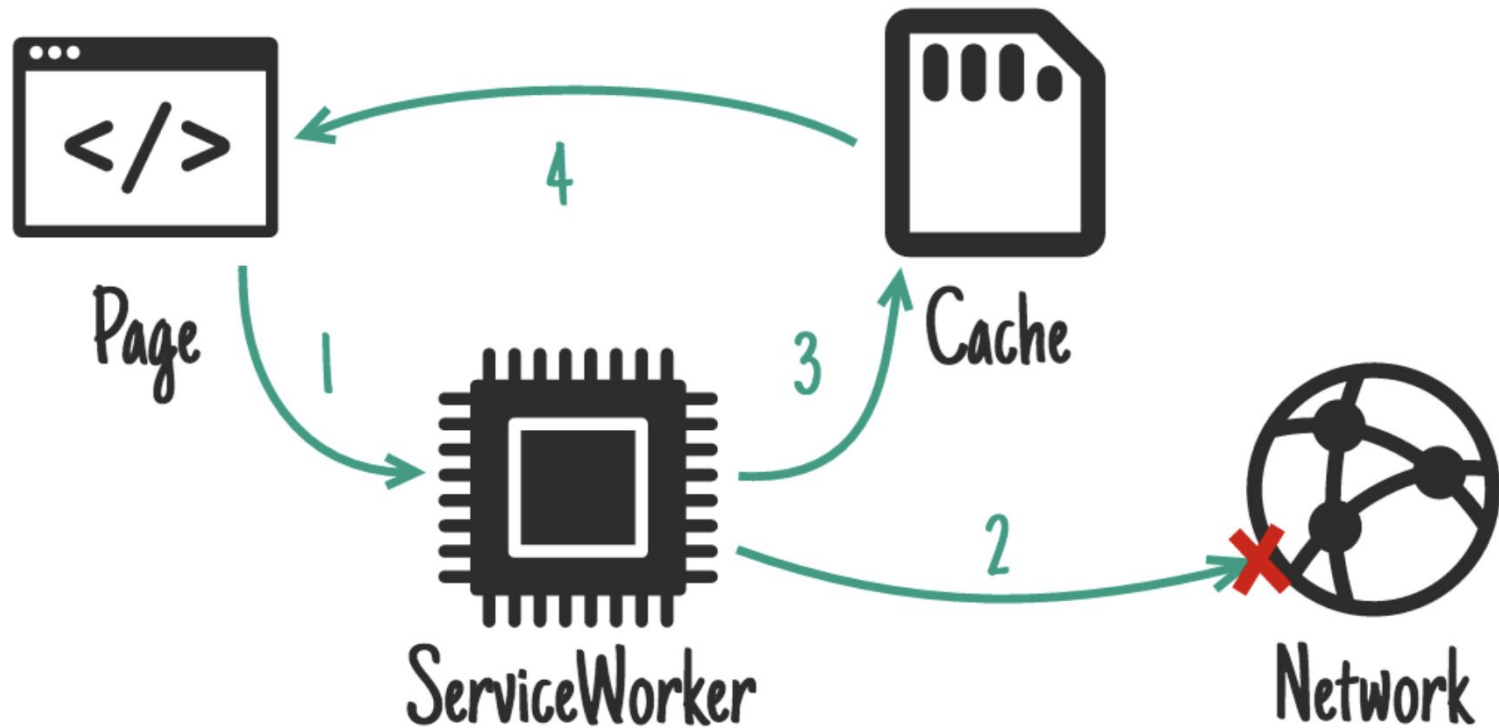




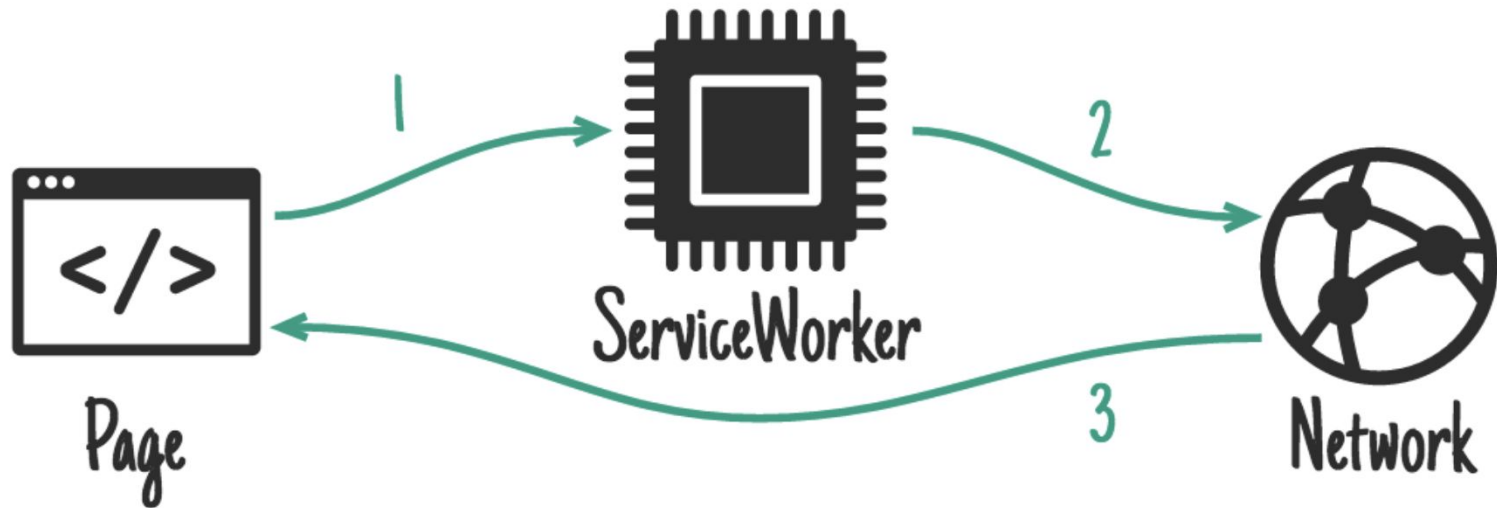
## Cache First (Cache Falling Back to Network)



## Network First (Network Falling Back to Cache)



Network Only



Kevin Martínez Javascript Developer

kevinccbsg@gmail.com

(@kjmesec)

(@KeepCoding\_)

