

Avancerad webbprogrammering
sedan 1997

ERIK BERGLUND,
LEKTOR

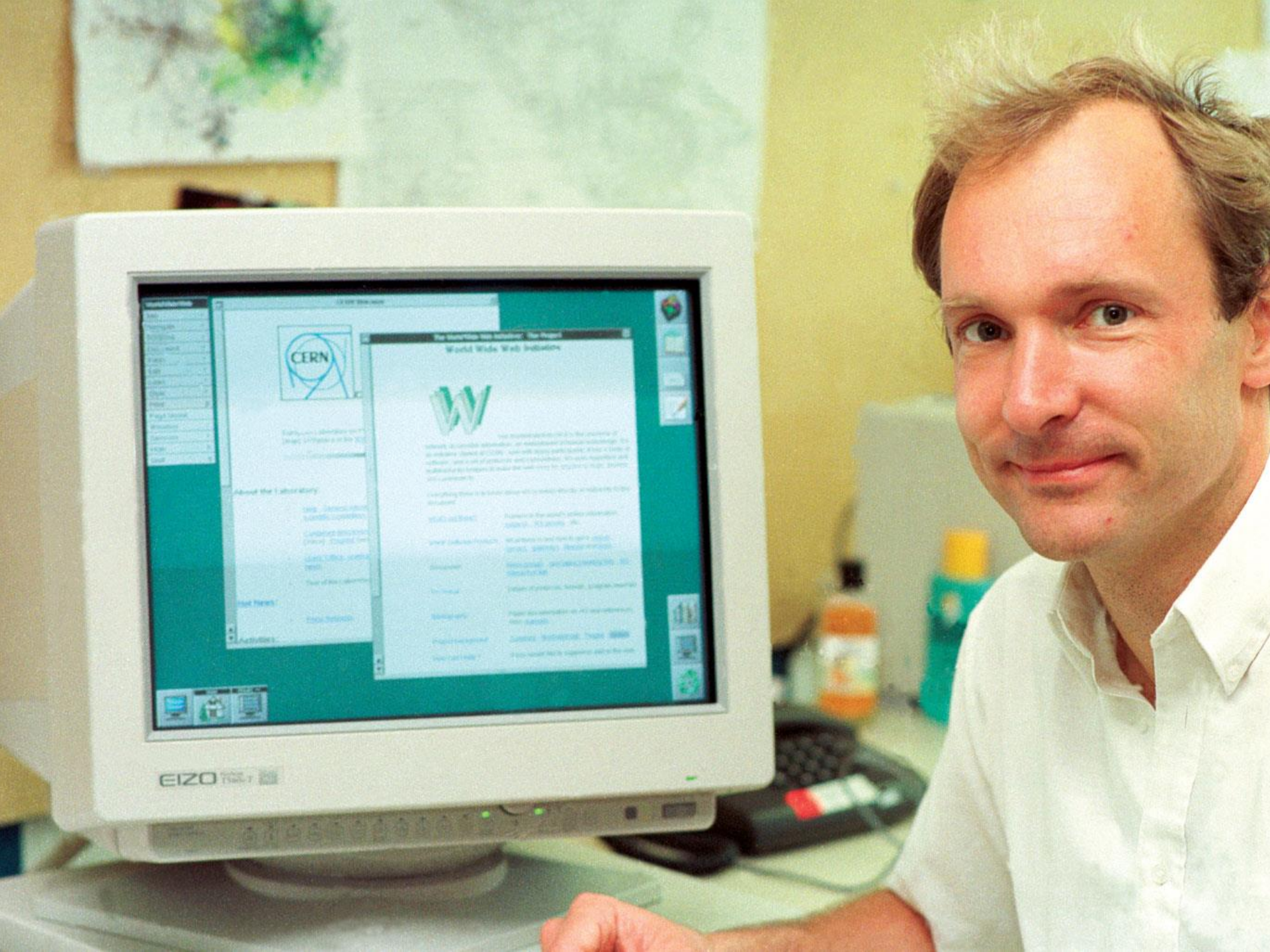


Webbapplikationsutveckling

Om webbutveckling

Rätt tänk om webb och applikationer

Rätt tänk för kursens tekniska plattform



TIOBE Index for January 2025

Jan 2025	Jan 2024	Change	Programming Language	Ratings	Change
1	1		 Python	23.28%	+9.32%
2	3	^	 C++	10.29%	+0.33%
3	4	^	 Java	10.15%	+2.28%
4	2	v	 C	8.86%	-2.59%
5	5		 C#	4.45%	-2.71%
6	6		 JavaScript	4.20%	+1.43%
7	11	^^	 Go	2.61%	+1.24%

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not **about** the *best* programming language or the language in which *most lines of code* have been written.

The operating system



Språk i webben

- **HTML, CSS, JavaScript**
 - JavaScript
 - *Enkeltrådat
 - Web Workers kan göra beräkningar på andra trådar men inte full access
 - Kan anropa server för att hämta ny data
 - Parallella dataanrop och strömmar av data
- Rättighetssystem liknande mobil för åtkomst av fillager, webbkamera, mikrofon, notiser osv...
 - Särskilda Web APler
- WASM, C++ compiled

Related Topics

Screen Capture API

▼ Guides

Using the Screen Capture API

▼ Interfaces

Screen Capture API

The Screen Capture API introduces additions to the existing Media Capture and Streams API to let the user select a screen or portion of a screen (such as a window) to capture as a media stream. This stream can then be recorded or shared with others over the network.

Related Topics

Web Bluetooth API

▼ Interfaces

BluetoothDevice

BluetoothRemoteGATTCharacteristic

BluetoothRemoteGATTServer

Web Bluetooth API



Experimental: This is an **experimental technology**.

Check the [Browser compatibility table](#) carefully before using this in production.

The Web Bluetooth API provides the ability to connect and interact with Bluetooth Low Energy peripherals.

References / Web APIs / Geolocation API

Related Topics

Geolocation API

▼ Guides

Using the Geolocation API

▼ Interfaces

Geolocation

GeolocationCoordinates

GeolocationPosition

GeolocationPositionError

▼ Properties

Geolocation API

Secure context: This feature is available only in [secure contexts](#) (HTTPS), in some or all [supporting browsers](#).

The **Geolocation API** allows the user to provide their location to web applications if they so desire. For privacy reasons, the user is asked for permission to report location information.

WebExtensions that wish to use the `Geolocation` object must add the "geolocation" permission to their manifest. The user's operating system will prompt the user to allow location access the first time it is requested.

Web APler exponerar
datorns egenskaper på
säkert sätt

Web Speech API

The **Web Speech API** enables you to incorporate voice data into web apps. The Web Speech API has two parts: `SpeechSynthesis` (Text-to-Speech), and `SpeechRecognition` (Asynchronous Speech Recognition.)

Progressive Web Apps



Responsive



Secure



Independent of
Connectivity



Bye Bye App Stores



Push Notifications



Fast



Highly Discoverable





ELECTRON

capacitor

Search

Docs Plugins CLI Community Blog Enterprise

A cross-platform native runtime for web apps.

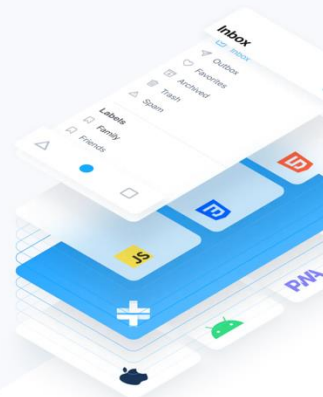
Capacitor is an open source native runtime for building Web Native apps. Create cross-platform iOS, Android, and Progressive Web Apps with JavaScript, HTML, and CSS.

[Install Capacitor →](#)

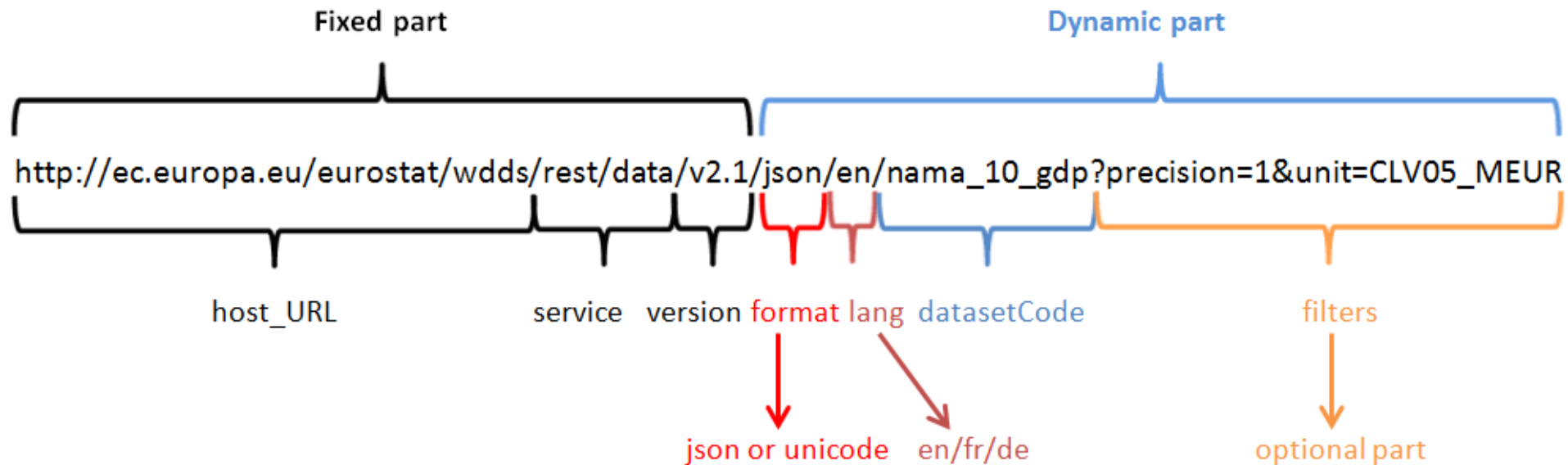
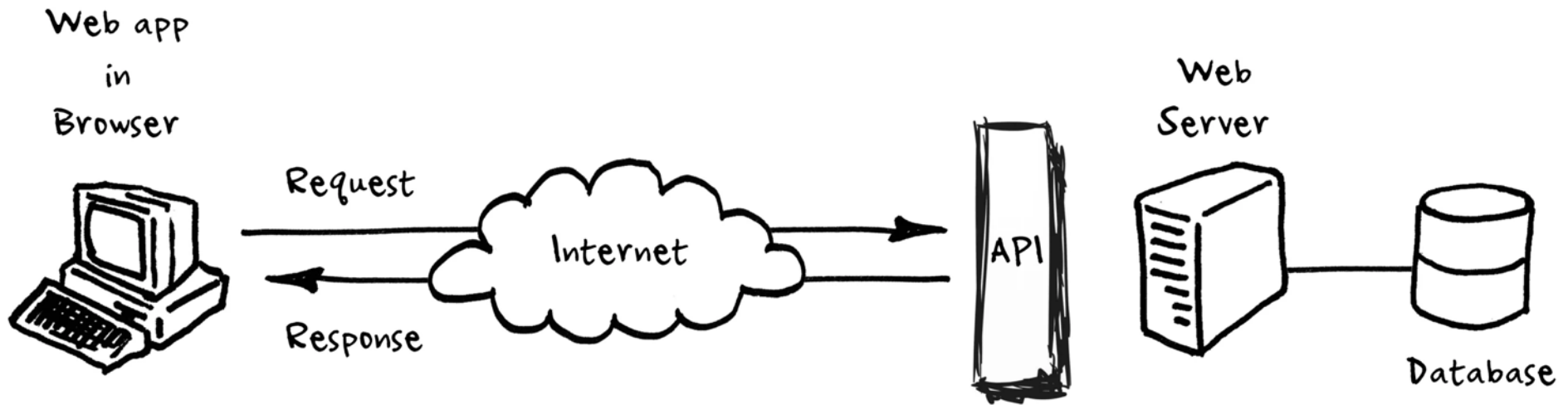
[Explore Plugins](#)

[Migrate from Cordova →](#)

🍏 🤖 PWA



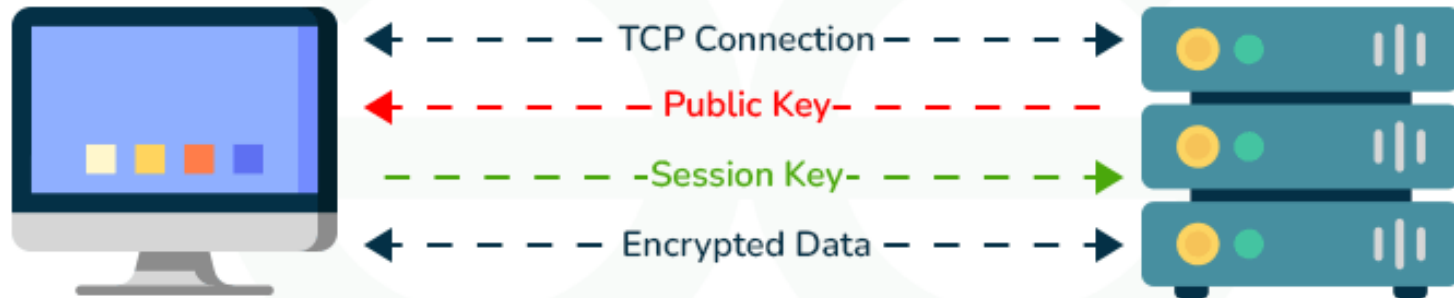
URLen som address, men även del av systemets gränssnitt



URLen är del av gränssnittet

- Nielsen URL rules 1999 (URLs as UI):
 - a **domain name** that is easy to remember and easy to spell
 - **short** URLs
 - **easy-to-type** URLs
 - URLs that visualize the **site structure**
 - URLs that are "**hackable**" to allow users to move to higher levels of the information architecture by hacking off the end of the URL
 - **persistent URLs** that don't change

HTTPS



HTTP/1.1 200 OK

Status Line

Date: Mon, 27 Jul 2009 12:28:53 GMT

Server: Apache/2.2.14 (Win32)

Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT

Content-Length: 88

Content-Type: text/html

Connection: Closed

Headers

<html>

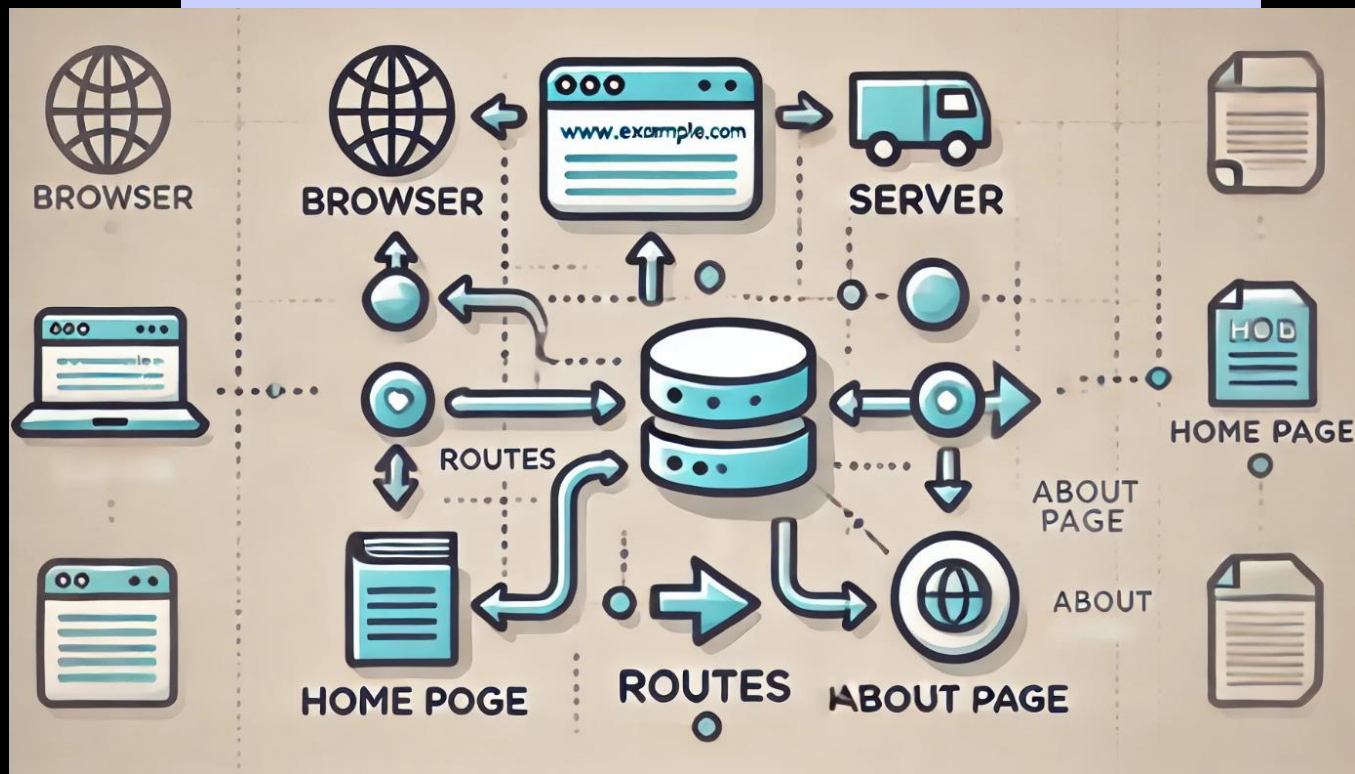
<body>

<h1>Hello, World!</h1>

</body>

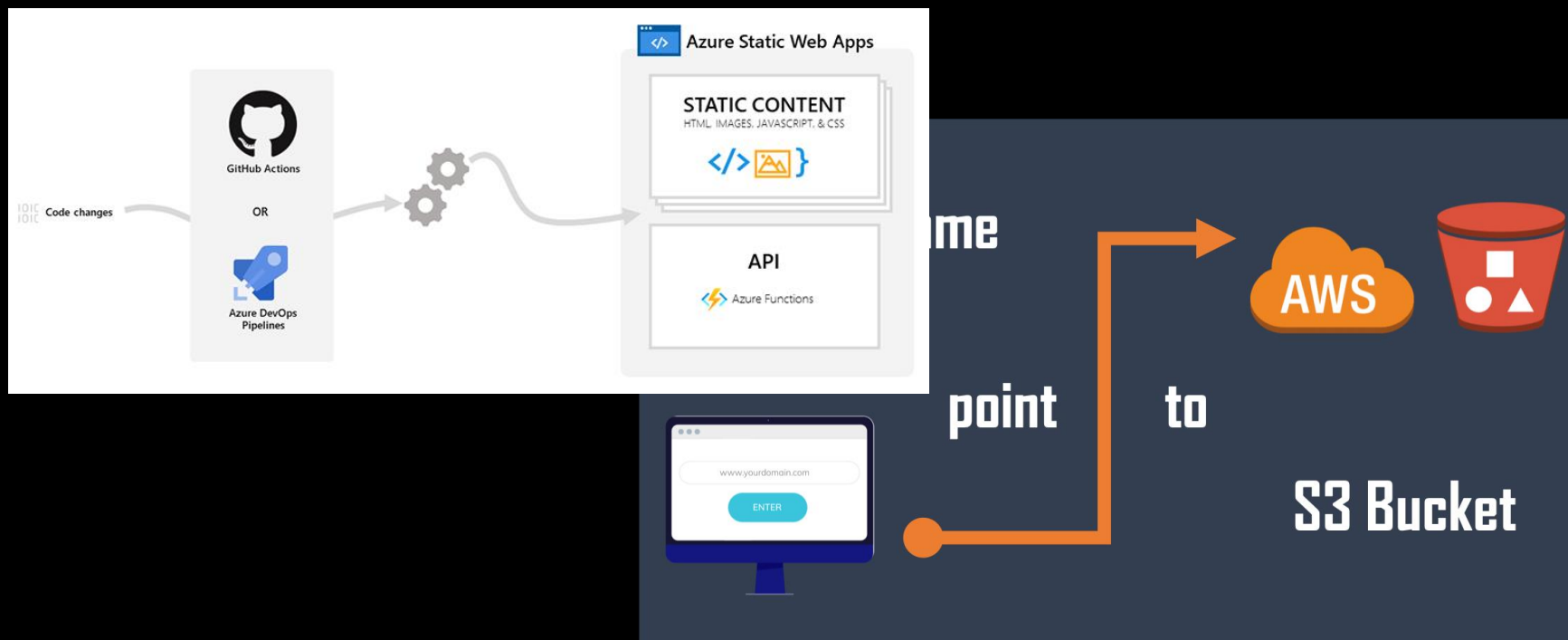
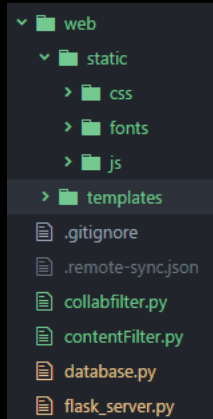
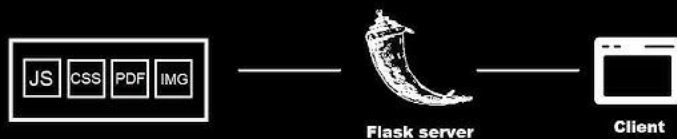
</html>

Body





Serve static files from python Flask server

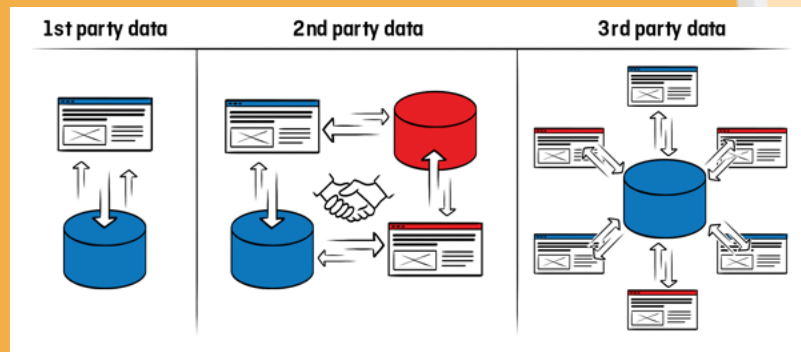
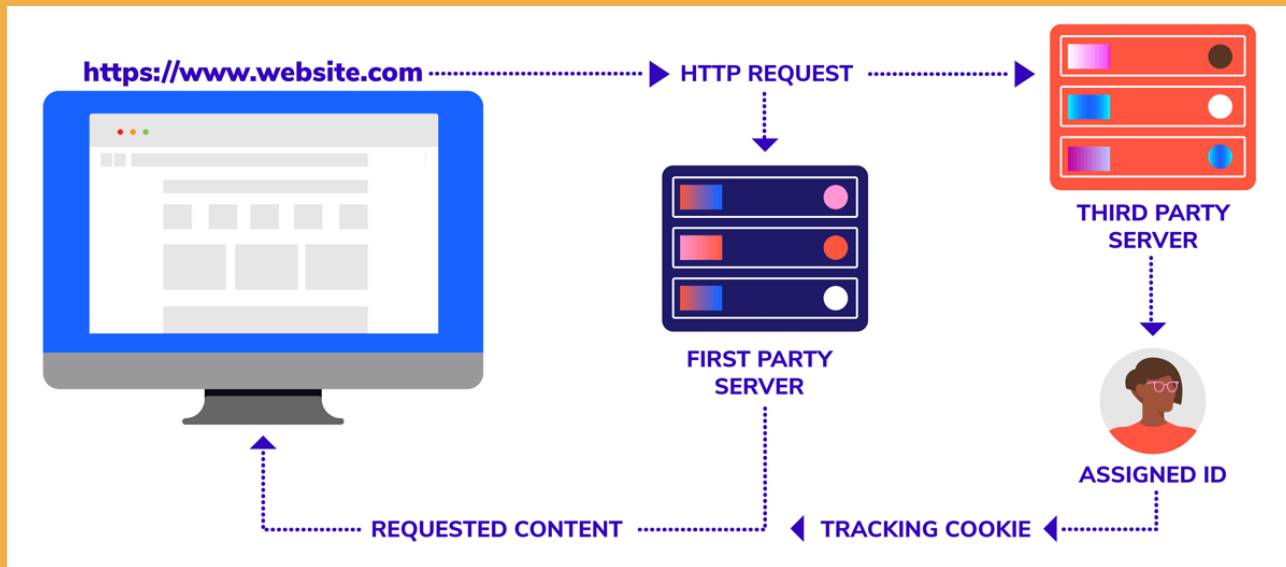


Local Storage



Session Storage

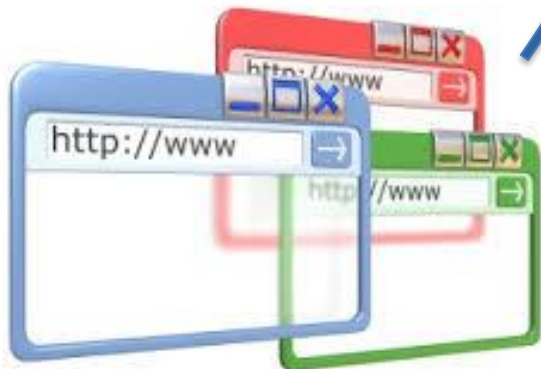




AJAX

Asynchronous Javascript And XML

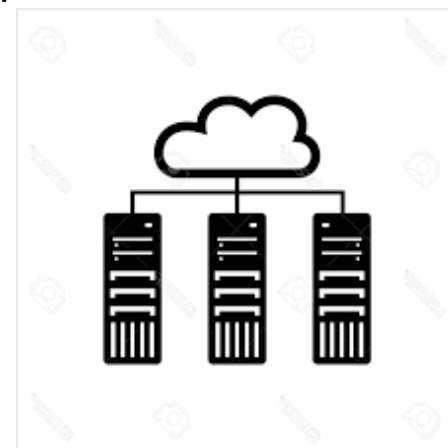
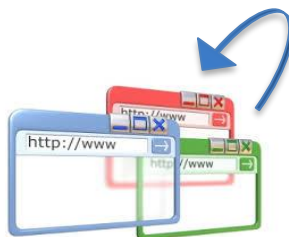
webbapp kör i
webbläsaren



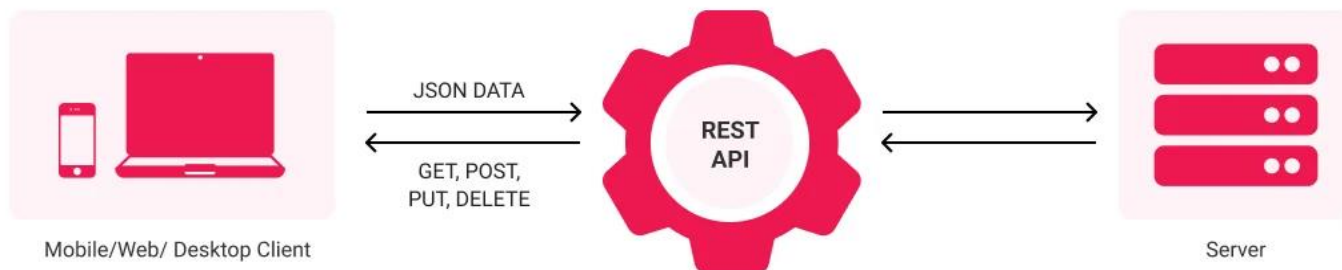
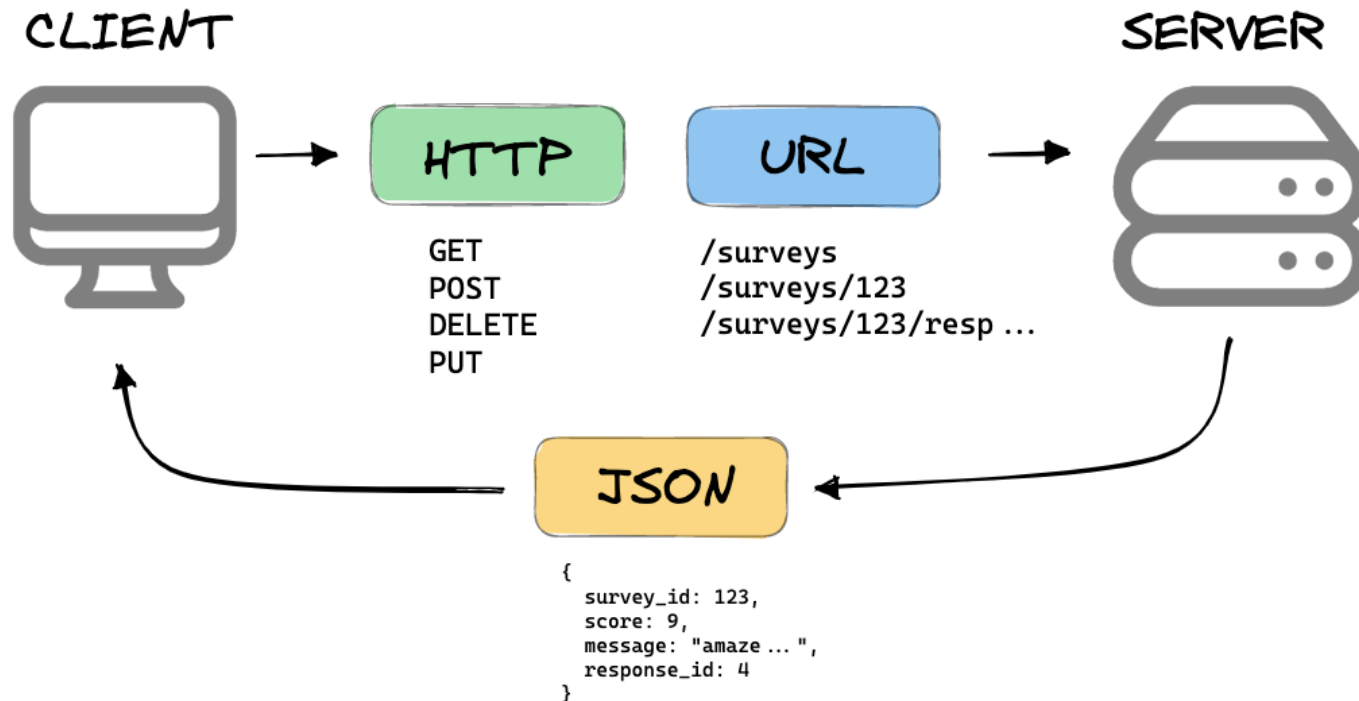
1. Installera/ ladda webbappen
HTML/CSS/JavaScript

2. Hämta/skicka data mot server

JSON



WHAT IS A REST API?



Webbsida [redigera | redigera wikitext]

Ej att förväxla med [Webbplats](#).

Webbsida är en enstaka sida på en [webbplats](#).

Tekniskt är en webbsida en [textfil](#) uppbyggd av till exempel [Hypertext Markup Language](#) (HTML) eller [Extensible Hypertext Markup Language](#) (XHTML) som kan läsas av en [webbläsare](#).

Det är möjligt för en webbsida att integrera olika typer av resurser. Dessa inkluderar stilinformation, som är ansvarig för hur sidan ser ut, [skript](#) som lägger till [interaktion](#) med en webbsida

Nomenklatur [redigera | redigera wikitext]

Ordet *webbsida* används

Ordet *hemsida* ([engelska](#) "webbsida", "ingångssida")

Se även [redigera | redigera wikitext]

- [Hemsida](#)
- [Webbdesign](#)
- [Webbutveckling](#)
- [Webbhotell](#)
- [Webbplats](#)

Källor [redigera | redigera wikitext]

- [^] What is the difference between a website and a web page?
- [^] "hemsida" [↗](#). Svenska Wikipedia.

- "webbsida" [↗](#). Svenska Wikipedia.

Kategori: World Wide Web

Webbapplikation [redigera | redigera wikitext]

(Omdirigerad från [Webbapp](#))



Den här artikeln **behöver [källhänvisningar](#)** för att kunna **verifieras**. (2022-09-01)

Åtgärda genom att lägga till pålitliga källor ([gärna som fotnoter](#)). Uppgifter utan källhänvisning kan raderas när som helst.

En **webbapplikation** är ett samlingsnamn för [programvara](#) som man kommer åt genom att använda en [webbläsare](#).

Se även [redigera | redigera wikitext]

- [AJAX](#)
- [Mobilapplikation](#)
- [Molntjänst](#)
- [Platform as a service](#)
- [Software as a service](#)
- [Tillämpningsprogram](#)
- [Tunn klient](#)
- [Rich Web Applications](#)
- [Webbtjänst](#)



Ingen religion, men ...

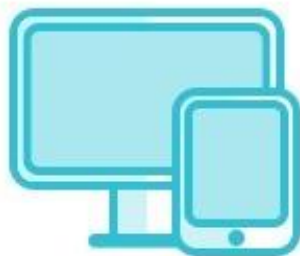




Users

Collect Data

Display Results



What the User Sees
& Interacts with
HTML, CSS, JavaScript

Frontend

Request

Response



Contains App Logic

PHP, JavaScript, Python, Java

Web Server



File System

HTML, CSS, Images



Database

MySQL, PostgreSQL,
MariaDB

Backend

Web Application Architecture



stripe

{ REST:API }

JSON
JavaScript Object Notation

{ REST:API }

JSON
JavaScript Object Notation

jQuery



JS

{ REST:API }

JSON
JavaScript Object Notation

Flask
web development,
one drop at a time



PYTHON



SQLite

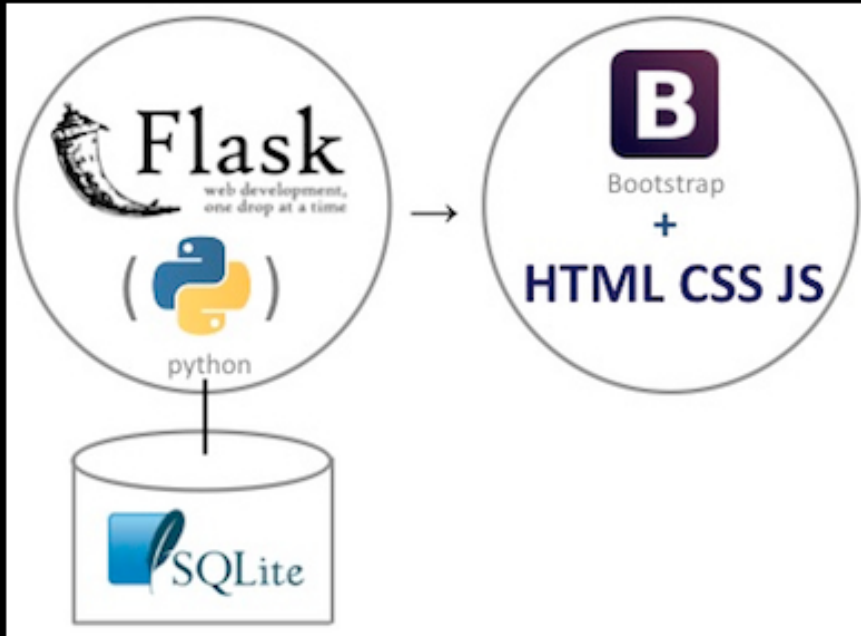
vs



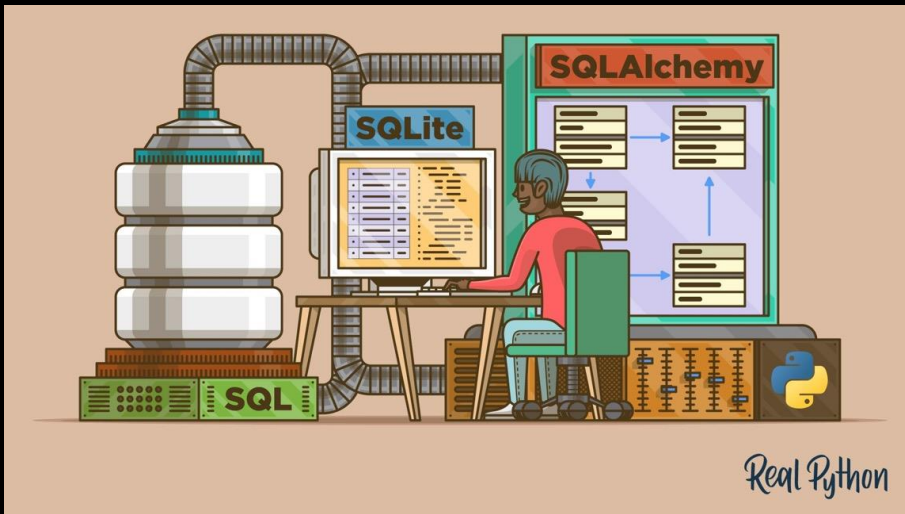
PostgreSQL

vs

MySQL



.gitignore sqlite-
db in project



Webbaserad deployment

- Uppdatera system ofta
 - varje gång användare byter sida
 - egentligen hela tiden!
- Ingen annan plattform tillåter så frekvent uppdatering
 - Bra för iterativ utveckling och ständig anpassning
 - jämför mobilen där en uppdatering kan blockas i en vecka av Google och Apples hantering

- När sidan drivs från klienten
 - Minimal kostnad
 - Maximal responsivitet i webbläsaren
 - JavaScript enkeltrådat
 - Maximalt antal samtidiga användare
- När sidan drivs från backend
 - Maximal upptid första gången
 - Svaga och gamla klient-maskiner fungerar bäst
 - dölja affärshemligheter

Webbens alla syntaxer



```
<!DOCTYPE>
```

```
<html>
```

```
<head>
```

```
<title> Demo </title>
```

```
</head>
```

```
<body>
```

```
<p> this is first demo of html structure </p>
```

```
</body>
```

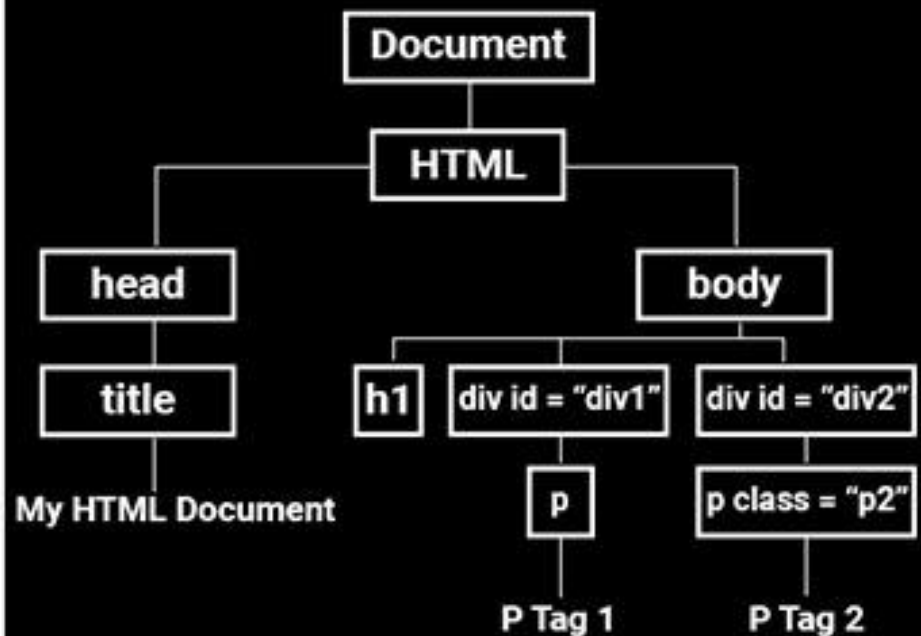
```
</html>
```

What is Document Object Model ?

HTML Document

```
index.html x
1  <html>
2    <head>
3      <title>My HTML Document</title>
4    </head>
5
6    <body>
7      <h1>Heading</h1>
8      <div id="div1">
9        <p>P Tag 1</p>
10     </div>
11     <div id="div2">
12       <p class="p2">P Tag 2</p>
13     </div>
14   </body>
15 </html>
```

Document Object Model (DOM)



Identifiera grupp för stil och annat

Identifiera specifikt object

```
▼ <div class="row header-logo hidden-xs">
  ::before
  ▶ <div class="col-sm-6 col-xs-12">...</div>
  ▼ <div class="col-sm-6 col-xs-12 text-right" id="search">
    <div class="spacer-25"></div>
    ▼ <form action="/suchergebnis.asp" method="GET" name="mmssearch">
      ▼ <div class="search-group">
        <input class="search-input" type="text" value placeholder="Name, WKN, ISIN" name="_"
        <input id="form-submit" class="search-submit" type="submit" value> = $0
      </div>
    </form>
    <div class="suggBox" id="suggBox0" style="z-index: 5002001; top: 203px; width: 700px; l
  ▶ <div class="suggBox" id="preSuggBox0" style="z-index: 5002000; top: 203px; width: 700px;
```

```
// Select all the <div> elements on the current page
const allDivs = document.getElementsByTagName("div")
```

onds on 28.07.2017 20:18:28 from NT -->

Document.getElementById()

Document.getElementsByClassName()

```

<html>
<head>
<style>
div.img {
    margin: 5px;
    padding: 5px;
    border: 1px solid #0000ff;
    height: auto;
    width: auto;
    float: left;
    text-align: center;
}

```

Example

Set the font size for different elements:

```

div.a {
    font-size: 15px;
}

div.b {
    font-size: large;
}

div.c {
    font-size: 150%;
}

```

Try it Yourself »

Relativ
Absolut
Fixed

...

normal
percent
smaller, larger, smaller-
auto

	Recommended	Occasional use	Not recommended
Screen	em, px, %	ex	pt, cm, mm, in, pc
Print	em, cm, mm, in, pt, pc, %	px, ex	

```
<head>
<style>
hr {color: red;}
p {margin-left: 20px;}
body {background-image: url("images/background.gif");}
</style>
</head>
```

Margin box

Border box

Padding box

Element box



Example

Change the background color of the <body> element to "lightblue" when the browser window is 600px wide or less:

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

Även tal i CSS är text.
DVS vi kan inte direkt
räkna med positioner

```
2  
3 function changeBackground(i) {  
4   var colors = ["red", "green", "blue", "purple", "orange"];  
5   if (i == colors.length) {  
6     i = 0;  
7   }  
8   color = colors[i];  
9   document.body.style.backgroundColor = color;  
10  console.log(i);  
11  return i;  
12 }  
13  
14 function Increment() {  
    changeBackground(i);  
}
```

OBS!

```
<style>  
  #d1 {  
    display: none !important;  
    color: red;  
  }  
</style>  
<script>  
  let showElement = () => {  
    let ele = document.getElementById('d1');  
    ele.setAttribute('style', 'display: block !important;');  
  }  
</script>
```

Overriding this property

<https://www.encodedna.com>

Now, this is important.

```
<html>
  <head>
    <div>
      <div>
        <form method="post" action="#" id="formvalue" onkeyup="
          drawChart()" />

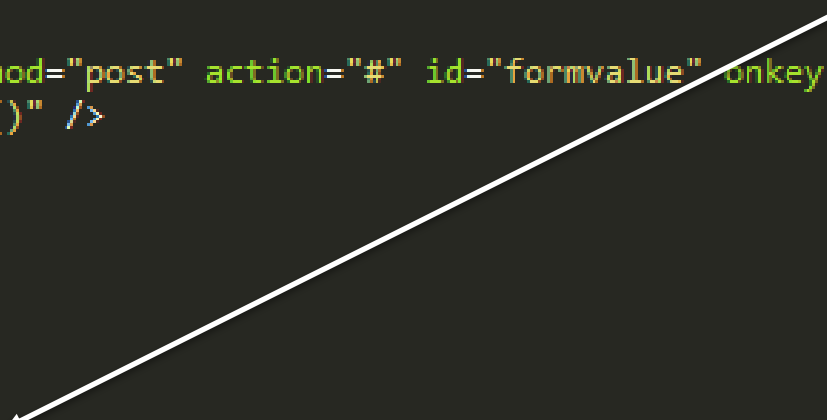
        </form>
      </div>
    </div>

    <script type="text/javascript" src="https://www.google.com/jsapi"></
      script>
    <script type="text/javascript">

      var bid = 43;
      var ask = 21;

      google.load("visualization", "1", {packages:["corechart"]});
      google.setOnLoadCallback(drawChart);
      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          ['Price', 'Quantity'],
          ['Value #1', bid],
          ['Value #2', ask],
        ]);
```

ordning
spelar roll



Om JavaScript

JAVA *is to*
JAVASCRIPT
as HAM *is to*
HAMSTER



ILLUSTRATED BY SEGUE TECHNOLOGIES

```

40
41 function parse (i, body) {
42     var obj = {};
43     body.split('\n')
44     .forEach(v=>v.replace(/\s*(.*)\s*:\s*(.*)\s*/,
45         (s, key, val)=>{obj[key]=isNaN(val)||val.length<1?val||undefined:Number(val);
46             })));
47
48     var objArr = Object.values(obj);
49
50     var res = [];
51     res[0] = i
52     res.push(objArr)
53
54     return res
55 }
56

```

Objects as first-class citizens

Functions as First-class citizens(supports functional programming)

Dynamic Typing

var dynamicType = "a string"
dynamicType = 5

```
1 let x = 1;
2
3 if (x === 1) {
4   let x = 2;
5
6   console.log(x);
7   // Expected output: 2
8 }
9
10 console.log(x);
11 // Expected output: 1
12
```

keyword	const	let	var
global scope	NO	NO	YES
function scope	YES	YES	YES
block scope	YES	YES	NO
can be reassigned	NO	YES	YES


```
const personPrototype = {  
  greet() {  
    console.log("hello!");  
  },  
};  
  
const carl = Object.create(personPrototype);  
carl.greet(); // hello!
```



JavaScript's Prototype



JAVASCRIPT

```
1 let wheels = { wheels: 4 }  
2 let color = { color: "black"}  
3 const Car = () => {  
4   return Object.assign({}, wheels, color);  
5 }  
6 let car = Car();  
7 console.log(car);
```

1. `false`, `0`, empty strings (`""`), `NaN`, `null`, and `undefined` all become `false`.
2. All other values become `true`.

Use `<script async>` when the script is independent(has no dependency). When the script depends, use `<script defer>`.

this

A function's `this` keyword behaves a little differently in JavaScript compared to other languages. It also has some differences between [strict mode](#) and non-strict mode.

In most cases, the value of `this` is determined by how a function is called (runtime binding). It can't be set by assignment during execution, and it may be different each time the function is called. The [bind\(\)](#) method can [set the value of a function's this regardless of how it's called](#), and [arrow functions](#) don't provide their own `this` binding (it retains the `this` value of the enclosing lexical context).

Callback Hell

A guide to writing asynchronous JavaScript programs

What is "callback hell"?

Asynchronous JavaScript, or JavaScript that uses callbacks, is hard to get right intuitively. A lot of code ends up looking like this:

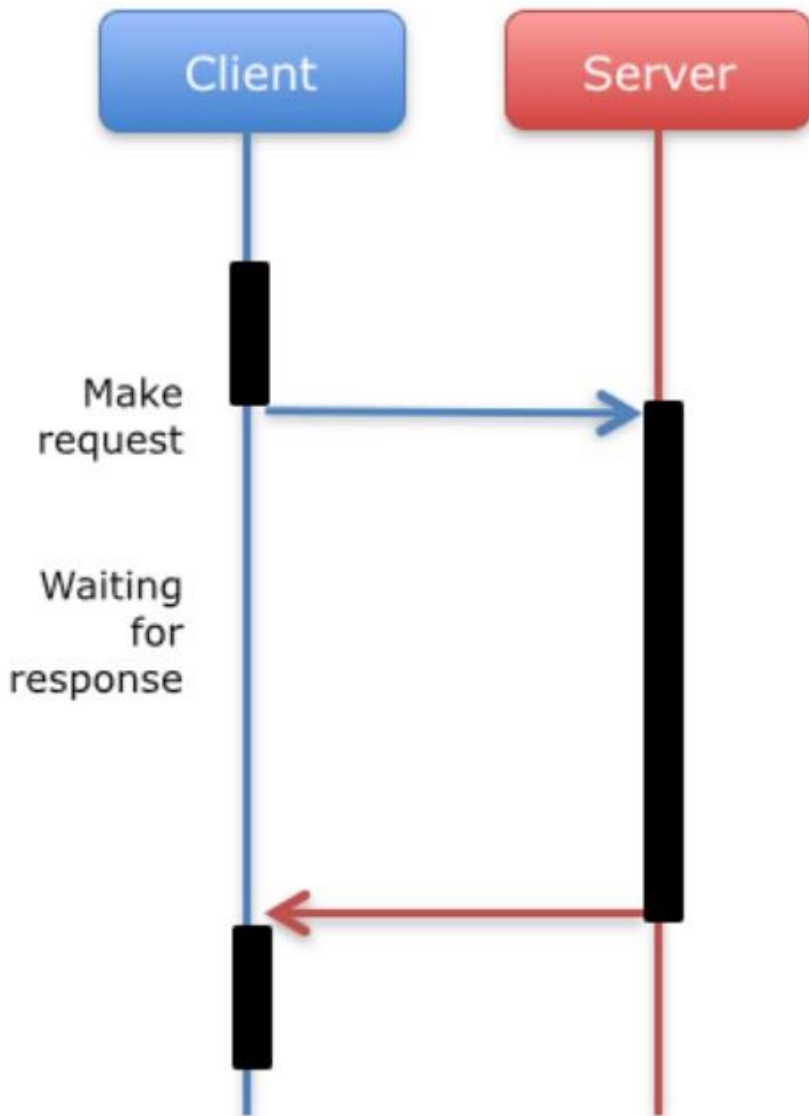
```
fs.readdir(source, function (err, files) {
  if (err) {
    console.log('Error finding files: ' + err)
  } else {
    files.forEach(function (filename, fileIndex) {
      console.log(filename)
      gm(source + filename).size(function (err, values) {
        if (err) {
          console.log('Error identifying file size: ' + err)
        } else {
          console.log(filename + ' : ' + values)
          aspect = (values.width / values.height)
          widths.forEach(function (width, widthIndex) {
            height = Math.round(width / aspect)
            console.log('resizing ' + filename + 'to ' + height + 'x' + height)
            this.resize(width, height).write(dest + 'w' + width + '_' + filename, function(err) {
              if (err) console.log('Error writing file: ' + err)
            })
          }).bind(this)
        }
      })
    })
  }
})
```

See the pyramid shape and all the `}}` at the end? Eek! This is affectionately known as **callback hell**.

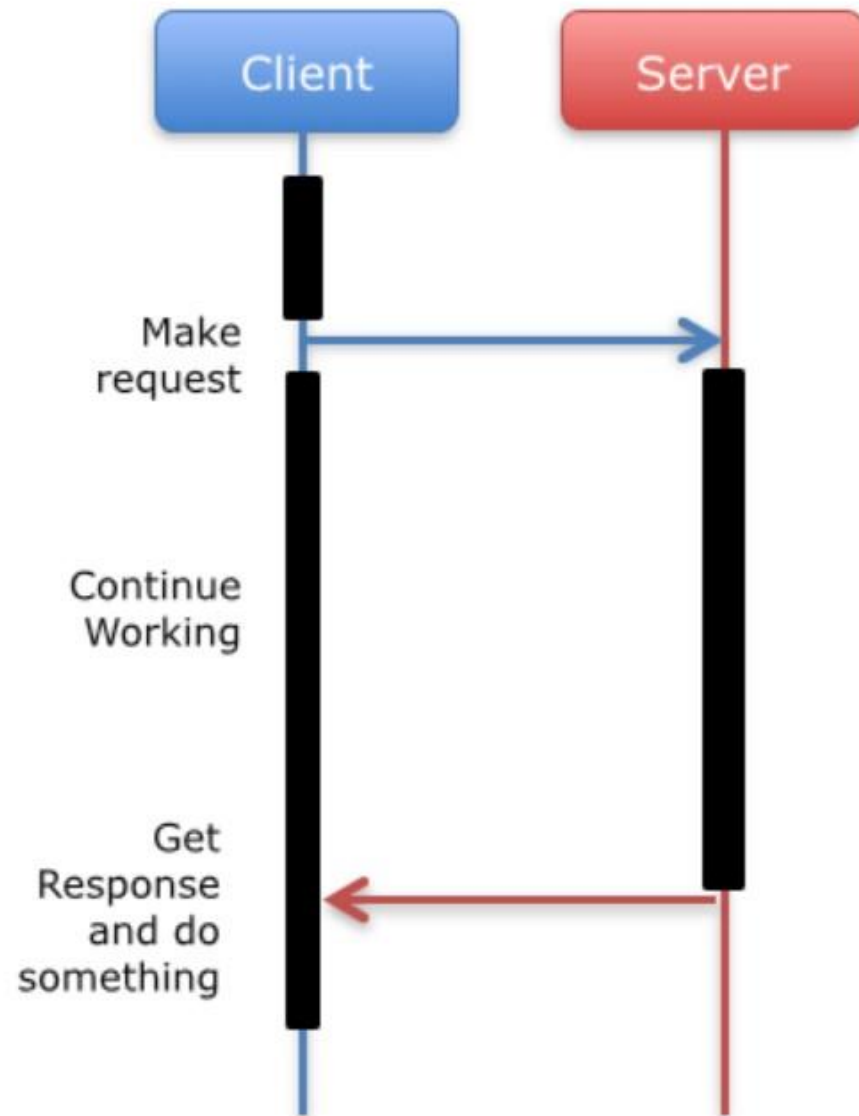
The cause of callback hell is when people try to write JavaScript in a way where execution happens visually from top to bottom. Lots of people make this mistake! In other languages like C, Ruby or Python there is the expectation that whatever happens on line 1 will finish before the code on line 2 starts running and so on down the file. As you will learn, JavaScript is different.

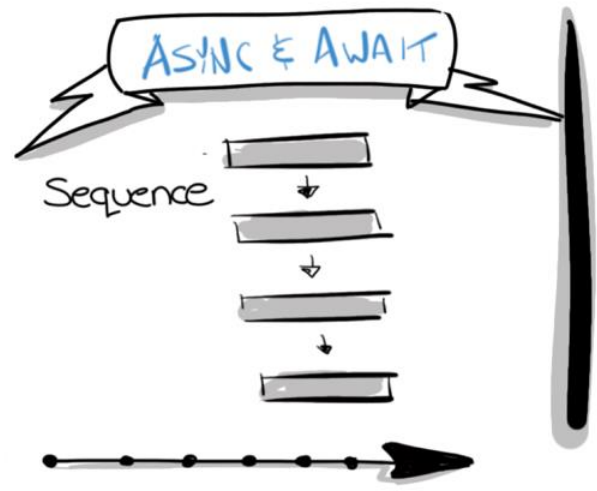
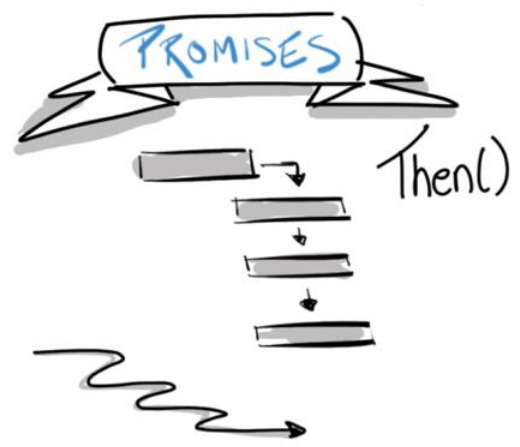
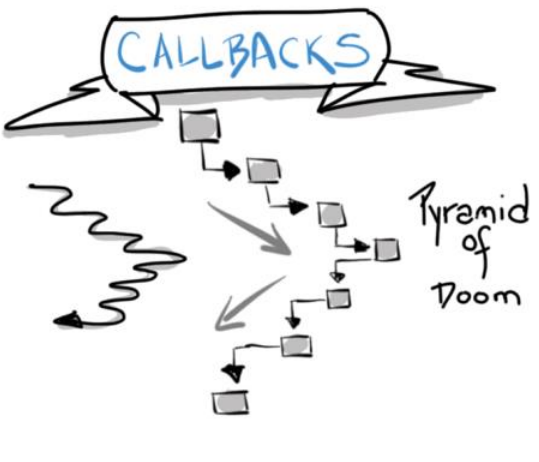
What are callbacks?

Synchronous



Asynchronous





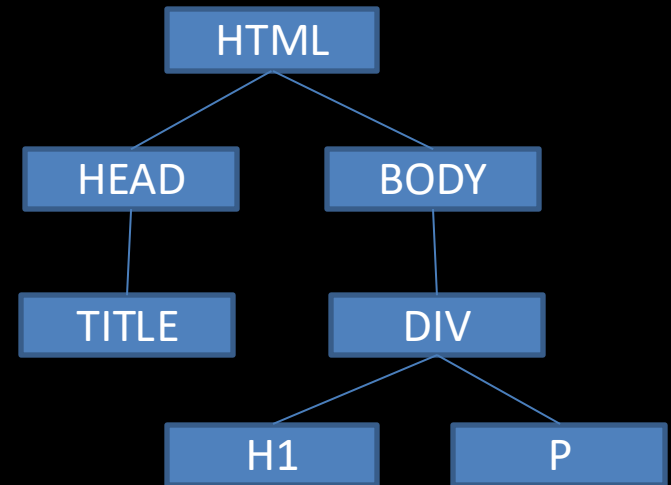
HTML & DOM

- Document Object Model

```
<html>
  <head>
    <title>my title</title>
  </head>
  <body>
    <div>
      <h1>Headline</h1>
      <p>Hello world!</p>
    </div>
  </body>
</html>
```

HTML File
Code in file

Browser



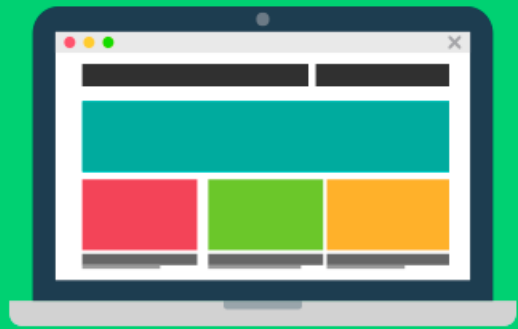
DOM Tree
Data structure in memory
(Note: Each node of this tree can be access as a JavaScript Object)

JSONasJs

```
{menu: {  
  id: "file",  
  value: "File",  
  popup: {  
    menuitem: [  
      {value: "New", onclick: "CreateNewDoc()"},  
      {value: "Open", onclick: "OpenDoc()"},  
      {value: "Close", onclick: "CloseDoc()"}  
    ]  
  }  
}}
```

Show plain JSON

```
{  
  status : 200 (number)  
  error : [ »  
    0 : 0 (number)  
    1 : No errors (string)  
  ]  
  emptyObject : { *empty* }  
  emptyArray : [ *empty* ]  
  data : { »  
    company_id : 268355140124  
    country_id : 39 (string)  
    timezone_id : 1 (string)
```

FRONTEND



BACKEND



Build fast, responsive sites with Bootstrap

Quickly design and customize responsive mobile-first sites with Bootstrap, the world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.

[Get started](#)

[Download](#)

Currently **v5.0.0-beta1** · [Previous releases](#)



Infuse analytics everywhere with the AI-powered embedded analytics platform. Start your free trial.

ads via Carbon

V4.6 (jQuery first)
V5 – beta (JavaScript first)

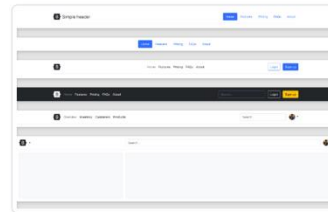
RESPONSIVE

WEB DESIGN



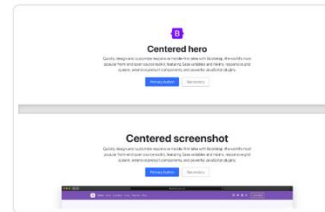
Snippets

Common patterns for building sites and apps that build on existing components and utilities with custom CSS and more.



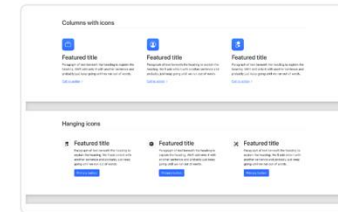
Headers

Display your branding, navigation, search, and more with these header components



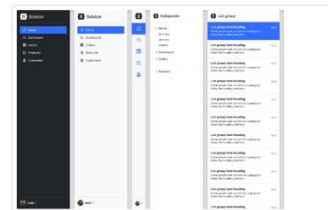
Heroes

Set the stage on your homepage with heroes that feature clear calls to action.



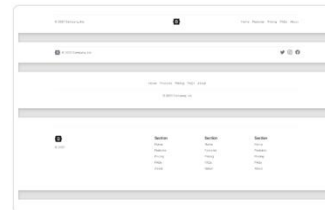
Features

Explain the features, benefits, or other details in your marketing content.



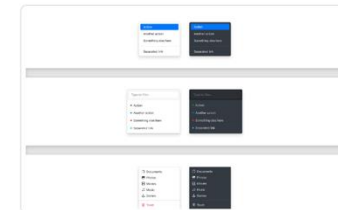
Sidebars

Common navigation patterns ideal for offcanvas or multi-column layouts.



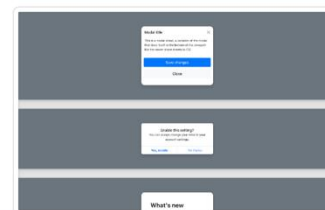
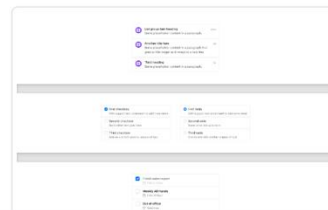
Footers

Finish every page strong with an awesome footer, big or small.



Dropdowns

Enhance your dropdowns with filters, icons, custom styles, and more.



Eriks TDOS | TrelloGetting started · BootstrapPricing & Plans - FirebaseSauce Labs: FeaturesjQuery

← → ↺jquery.com

☆🔍📶48JB☰

🔍

PluginsContributeEventsSupportjQuery Foundation



write less, do more.

Save time, save money.

Test in 300+ browsers, 8+ OSes, and 50+ mobile devices.

Trusted by the jQuery team.

BrowserStack

Try it for free

DownloadAPI DocumentationBlogPluginsBrowser Support

Search



Lightweight Footprint

Only 32kB minified and gzipped. Can also be included as an AMD module.



CSS3 Compliant

Supports CSS3 selectors to find elements as well as in style property manipulation.



Cross-Browser

IE, Firefox, Safari, Opera, Chrome, and more.

Download jQuery

v1.11.3 or v2.1.4

[View Source on GitHub →](#)
[How jQuery Works →](#)

What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Corporate Members



Support from our corporate members makes it possible for the jQuery Foundation to continue our work on our JavaScript libraries and pushing the open web forward with events and participation in the standards process. View our [members page](#) for a full list of corporate and individual members.

Other jQuery Foundation Projects





fo1.keyapp-demo.mp4teknik.mkvapp-demo.mp4 Removedsc_o.m2tssc_u2.m2tsThesisPlanReactNativ....pdf

Show All

DOM Traversal and Manipulation

Get the `<button>` element with the class 'continue' and change its HTML to 'Next Step...'

```
1 | $( "button.continue" ).html( "Next Step..." )
```

Event Handling

Show the `#banner-message` element that is hidden with `display:none` in its CSS when any button in `#button-container` is clicked.

```
1 | var hiddenBox = $( "#banner-message" );  
2 | $( "#button-container button" ).on( "click", function( event ) {  
3 |     hiddenBox.show();  
4 | });
```

Ajax

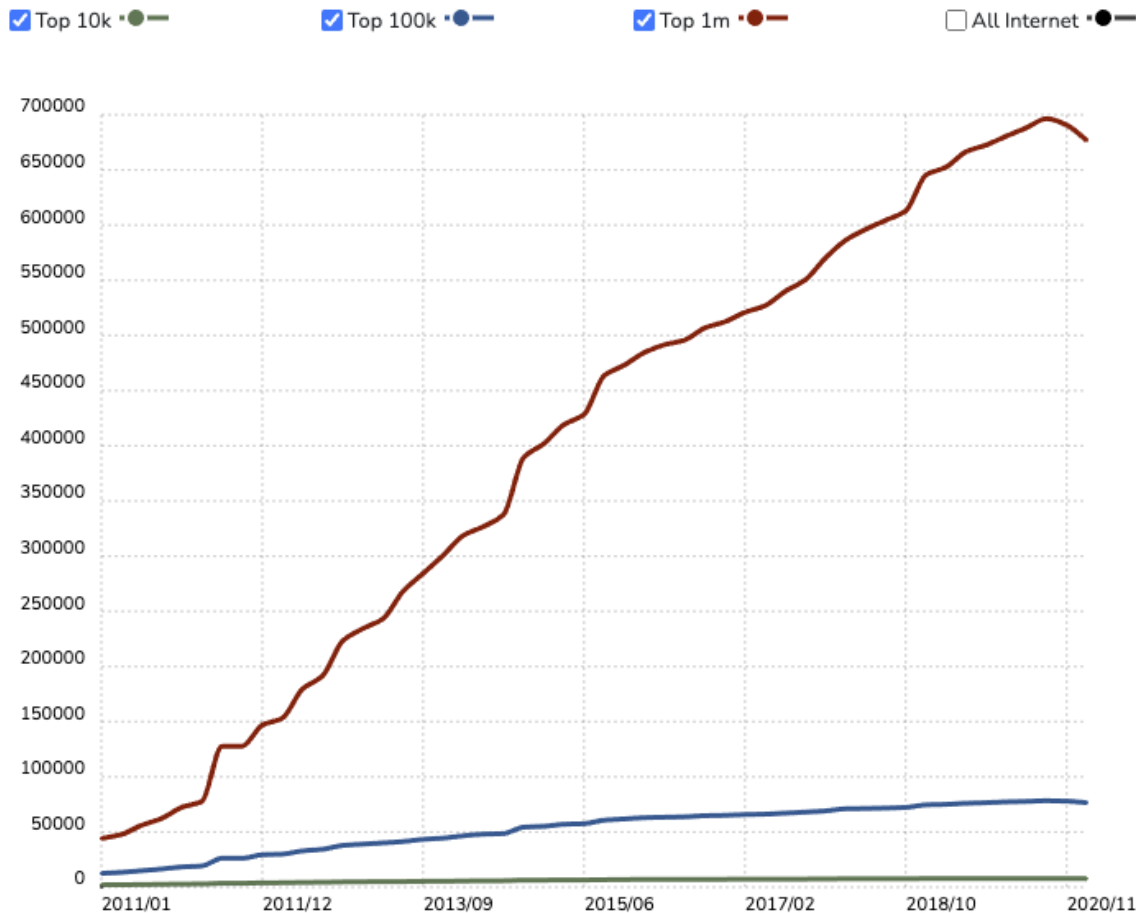
Call a local script on the server `/api/getWeather` with the query parameter `zipcode=97201` and replace the element `#weather-temp`'s html with the returned text.

```
1 | $.ajax({  
2 |     url: "/api/getWeather",  
3 |     data: {  
4 |         zipcode: 97201  
5 |     },  
6 |     success: function( result ) {  
7 |         $( "#weather-temp" ).html( "<strong>" + result + "</strong> degrees" );  
8 |     }  
9 | });
```

```
$( "#p1" ).css( "color", "red" ).slideUp( 2000 ).slideDown( 2000 );
```

jQuery is a **JavaScript library** designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax.

jQuery Usage Statistics



[Download Lead List](#)

We know of at least **71,803,422** live websites using jQuery.

Site Totals

Total Live **71,803,422**

4,992,521 additional website redirects?

Swedish Live Sites **267,314**
Estimated

Top 1m **73.83%**
738,305

Top 100k **80.34%**
80,339

Top 10k **79.81%**
7,981

United States **41,862,215***

United Kingdom **3,133,706***

Russia **2,877,676***

Germany **2,173,899***

China **2,156,359***

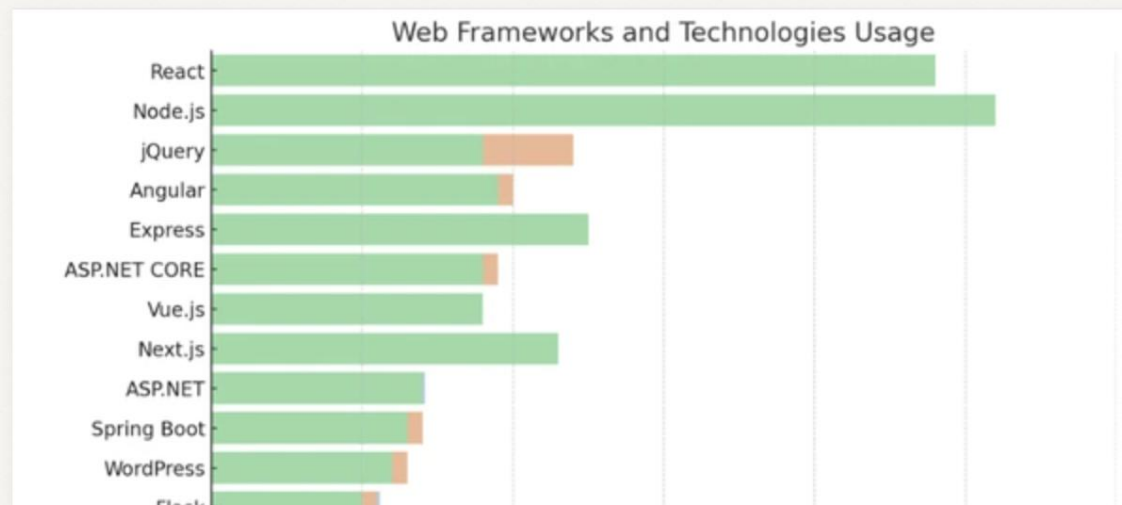
Japan **1,554,851***

What Is JQuery & Why's It The Top JS Library In 2024?

The idea that JQuery is a legacy framework on its way to obsolescence needs to be clarified—so let's state the facts and clear up the ambiguity.

According to Stack Overflow's survey, [22.87%](#) of developers still use JQuery, it's the third most popular web framework and one of the most starred repositories on GitHub.

Furthermore, it is used by more professional devs than all the other frameworks included in the survey, as can be seen in the graph below.



Stripe, Inc. [US]

https://stripe.com/se

Stripe

Features Pricing More

Documentation Help & Support Sign In

Payments infrastructure for the internet

Whether you're building a marketplace, mobile app, online storefront, or subscription service, Stripe has the features you need.

Explore the Stripe stack Sign up

SQUARESPACE

Create your own SPACE

GET STARTED

Dashboard for iPhone. Download the app to track and manage your payments on the go. Learn more

Feature-packed payments

Card storage, subscriptions, and more.

See all features

Simple pricing

Leave complex fee structures behind.

Learn more

Natively web & mobile

JavaScript & native libraries available.

View documentation

KICKSTARTER

Instacart

shopify

Pinterest

lyft

22-jQuery2.pptx

Developing Ajax App....pptx

Part1_Slides.pptx

L12-jQueryIntro.pptx

Module 1 - 1 - Introd....pptx

fo1.key

Show All

Erika TDCGetting stPricing & FjQueryTDP027Google CaljQuery LeSignin Temview-sourSQL ExpreCSS3 MiniFacebookFörsta använd...

←→Ghttps://developers.facebook.com

facebook for developersProductsDocsTools & SupportNews

SearchMy Apps

Connect on a global scale.

Build, grow and monetize your apps with Facebook



Facebook Login
Account Creation in two taps



Sharing on Facebook
Promote your app or website organically



Facebook Social Plugins
Easily make your app or website social



Facebook Analytics for Apps
Understand how people use your app



App Monetization
Monetize with ads and publisher tools

View All Products



News
FbStart Helps Mobile Apps go big
FbStart is a new program from Facebook designed to help early...

22-jQuery2.pptxDeveloping Ajax App.....pptxPart1_Slides.pptxL12-jQueryIntro.pptxModule 1 - 1 - Introd....pptxfo1.key

Show All



FRONTEND



BACKEND



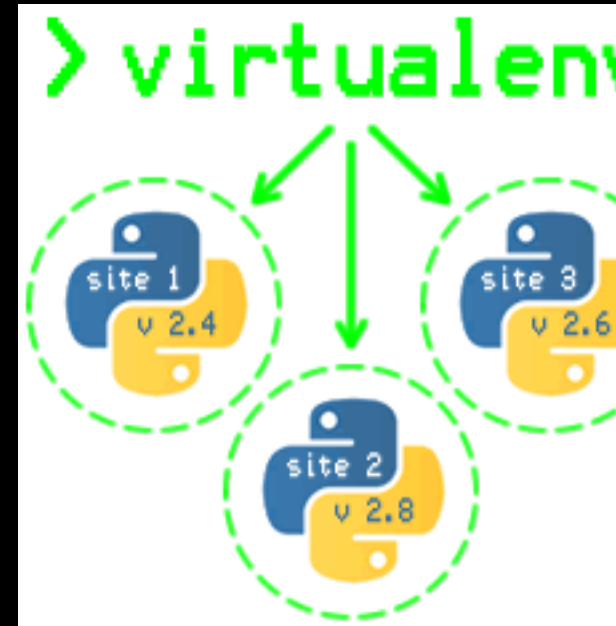
Flask

web development,
one drop at a time



Python

- Dynamisk typing “duck typing”
- TAB istället för {}
- Inget ; vid slut av uttryck
- Effektivt, enkelt, kortfattat
- Väldigt avancerad listhantering på språklig/grundläggande nivå



Micro FRAMEWORKS

Sinatra

README
DOCUMENTATION
CONTRIBUTE
CODE
CREW
ABOUT

Put this in
your pipe

```
require 'sinatra'

get '/hi' do
  "Hello World!"
end
```

and smoke it

```
$ gem install sinatra
$ ruby hi.rb
```



Scalatra

Slim

PHP Micro Framework



Flask

web development,
one drop at a time

HOME WHY? README DOWNLOAD SOURCE CONTACT



Express JS

microjs


```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```

```
from markupsafe import escape

@app.route('/user/<username>')
def show_user_profile(username):
    # show the user profile for that user
    return 'User %s' % escape(username)

@app.route('/post/<int:post_id>')
def show_post(post_id):
    # show the post with the given id, the id is an integer
    return 'Post %d' % post_id

@app.route('/path/<path:subpath>')
def show_subpath(subpath):
    # show the subpath after /path/
    return 'Subpath %s' % escape(subpath)
```



Flask kodstil är även vanlig för
serverless function backends

som AWS Lambda, Azure
Functions och Google Cloud

Amazon Lambda vs Azure Functions:

Serverless Computing

Run with a Production Server

When running publicly rather than in development, you should not use the built-in development server (`flask run`). The development server is provided by Werkzeug for convenience, but is not designed to be particularly efficient, stable, or secure.

Instead, use a production WSGI server. For example, to use [Waitress](#), first install it in the virtual environment:

```
$ pip install waitress
```

You need to tell Waitress about your application, but it doesn't use `FLASK_APP` like `flask run` does. You need to tell it to import and call the application factory to get an application object.

```
$ waitress-serve --call 'flaskr:create_app'
```

```
Serving on http://0.0.0.0:8080
```

See [Deployment Options](#) for a list of many different ways to host your application. Waitress is just an example, chosen for the tutorial because it supports both Windows and Linux. There are many more WSGI servers and deployment options that you may choose for your project.

Continue to [Keep Developing!](#).

```
from flask import Flask, jsonify
```

```
app = Flask(__name__)
```

```
@app.route('/api/get-json')
```

```
def hello():
```

```
    return jsonify(hello='world') # Returns HTTP Respo
```



FLASK TEMPLATES

IN THIS ARTICLE, WE WILL LEARN ABOUT
TEMPLATES IN THE FLASK WEB FRAMEWORK
AND HOW TO USE THEM.

Laddingstider

Home

Open DevTools

CSS

Console

Network

Storage

Issues

Command Menu

Mobile Simulation

DOM

JavaScript

Performance

Accessibility

Remote Debugging

Memory

Media

WebAuthn

Workspaces

Progressive Web Apps

Security

Keyboard Shortcuts

Resources

Customize

Extend DevTools

It's a wrap for **Chrome Dev Summit 2020**! Watch all the sessions at goo.gle/cds20-sessions now!

Home > Products > Chrome DevTools > Tools

Rate and review

Chrome DevTools

Chrome DevTools is a set of web developer tools built directly into the [Google Chrome](#) browser. DevTools can help you edit pages on-the-fly and diagnose problems quickly, which ultimately helps you build better websites, faster.

Check out the video for live demonstrations of core DevTools workflows, including debugging CSS, prototyping CSS, debugging JavaScript, and analyzing load performance.



Open DevTools

There are many ways to open DevTools, because different users want quick access to different parts of the DevTools UI.

- When you want to work with the DOM or CSS, right-click an element on the page and select **Inspect** to jump into the **Elements** panel. Or press **Command+Option+C** (Mac) or **Control+Shift+C** (Windows, Linux, Chrome OS).
- When you want to see logged messages or run JavaScript, press **Command+Option+J** (Mac) or

Table of contents

Open DevTools

Get started

Discover DevTools

Device Mode

Elements panel

Console panel

Sources panel

Network panel

Performance panel

Memory panel

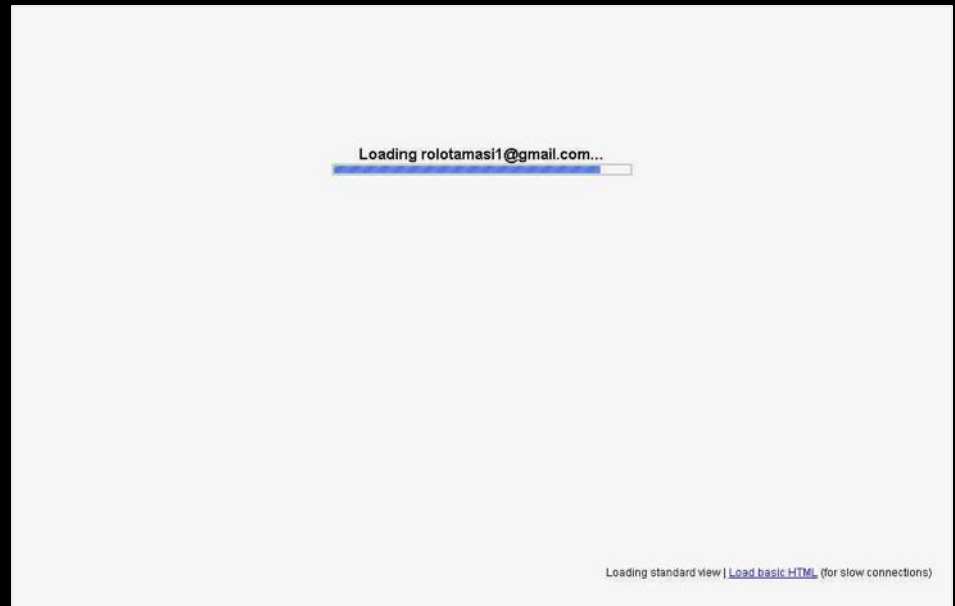
Application panel

Security panel

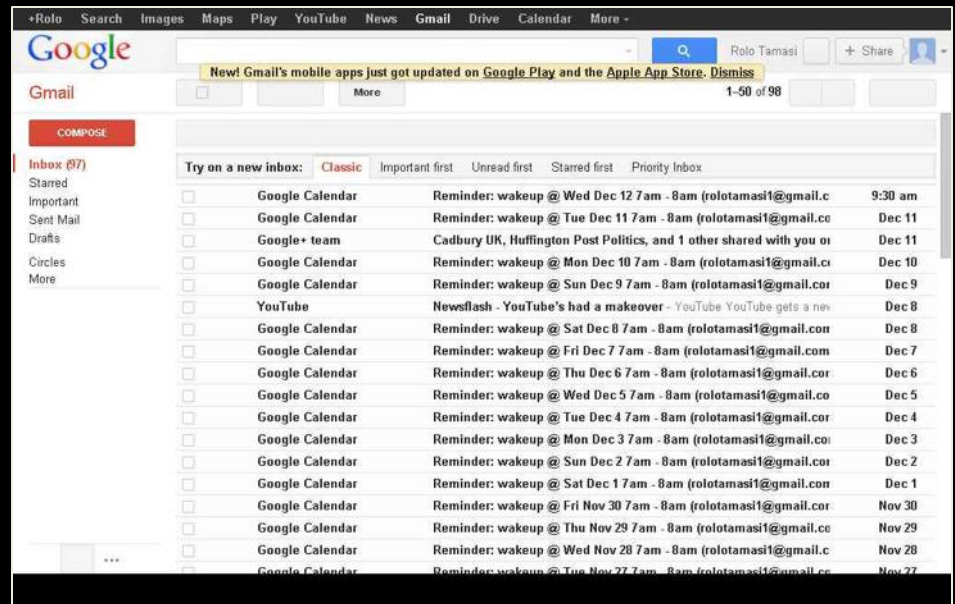
Community

Feedback

3.3 seconds
window.onload



4.8 seconds
90% rendered



window.onload

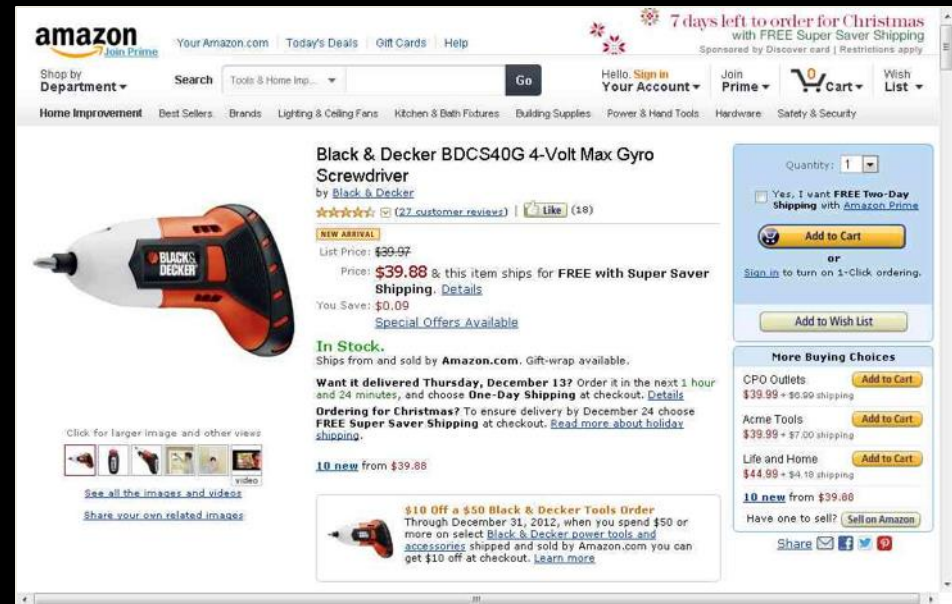
≠

user perception

2.0 seconds
88% rendered



5.2 seconds
window.onload



KYLE RUSH

[HOME](#)[ABOUT](#)[CONTACT](#)

Meet the Obama campaign's \$250 million fundraising platform

Nov 27, 2012
We made the new platform 60% faster and this resulted in a 14% increase in donation conversions.

- 6 month life span
- \$250 million dollars, 4,276,463 donations
- 81,548,259 pageviews, 17,807,917 unique visitors
- 60% faster time to paint than previous platform
- 240 a/b tests, 49% increase in conversions

kylerush.net/blog/meet-the-obama-campaigns-250-million-fundraising-platform/

General Web Terms

Ej denna kurs, de som vill läser sådant
i TDDD27 vt2 (om ett år?)

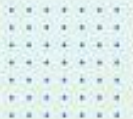
TypeScript vs. JavaScript: The Difference You Should Know



TypeScript



JavaScript



www.radixweb.com

Ej denna kurs, de som vill läser sådant
i TDDD27 vt2 (om ett år?)



React

VS



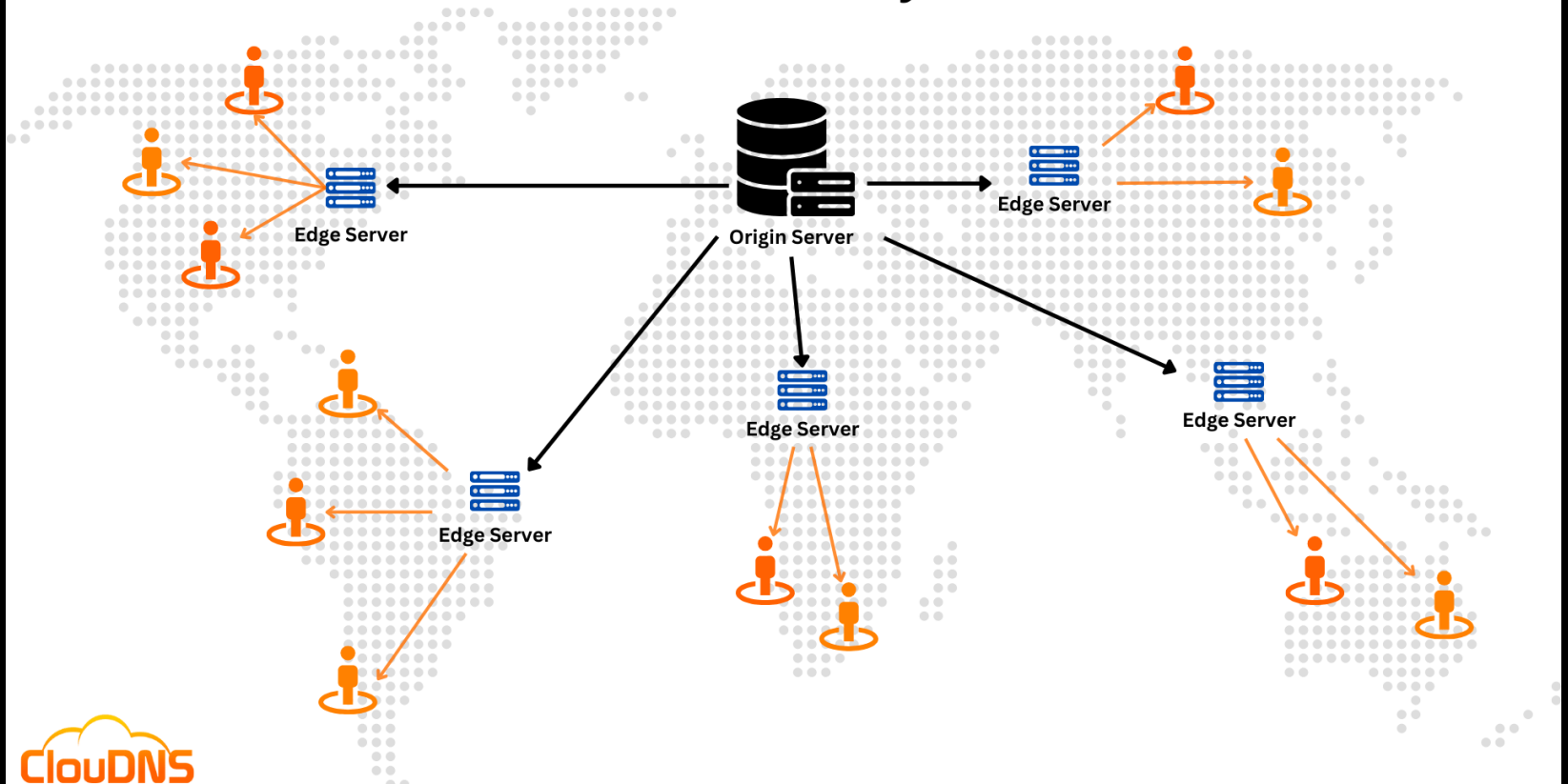
Vue.js

VS



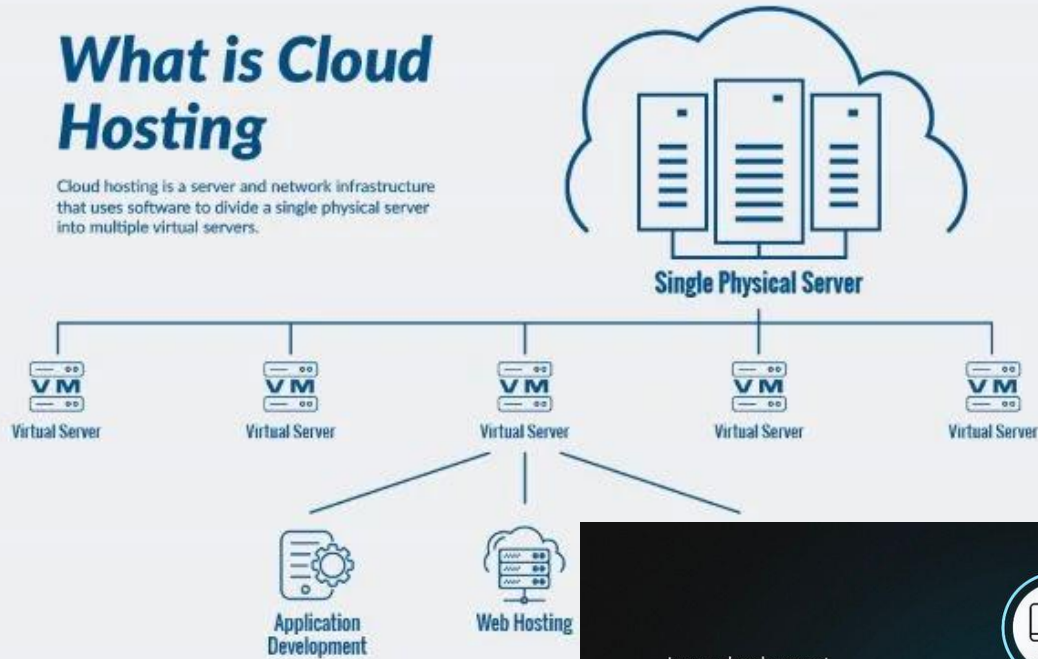
ANGULAR

CDN (Content Delivery Network)



What is Cloud Hosting

Cloud hosting is a server and network infrastructure that uses software to divide a single physical server into multiple virtual servers.



Lower development expenses

Proven backend tools

Advanced security and compliance

Backend as a Service benefits

Focus on the front-end

Scalability and great performance

Accelerated app development



WHAT IS AN EDGE SERVER?

Edge Computing Embedded Systems



#Knowledge

blog post

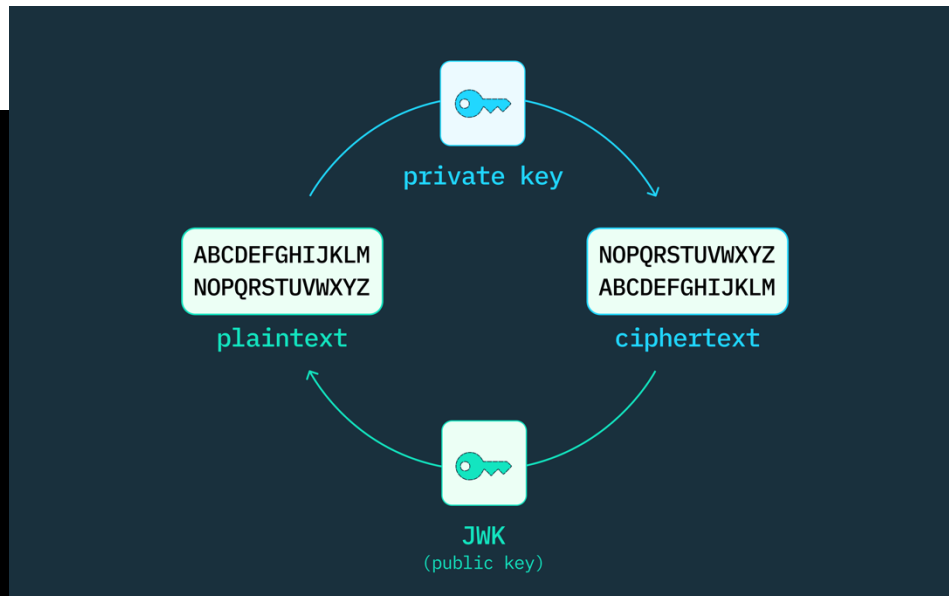
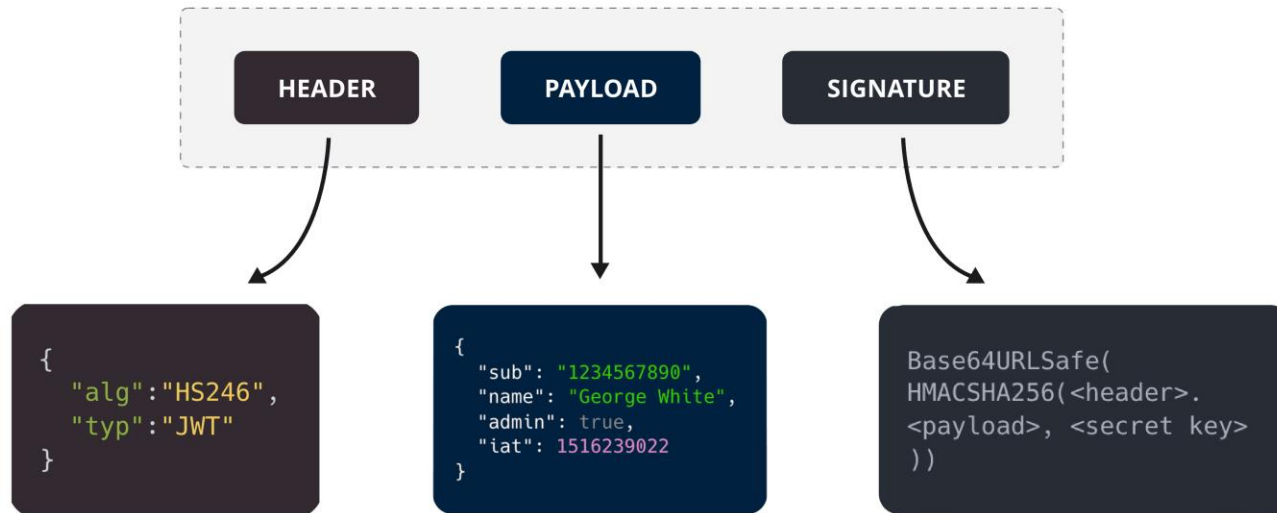
The Hidden Gold of Web Accessibility: Everything About ARIA Labels

read it on the blog →

SUBMIT

```
<button>  
aria-label="press enter to submit"  
</button>
```

Structure of a JSON Web Token (JWT)



JS Best Practise and Common Errors

JavaScript Best Practices

[< Previous](#)[Next >](#)

Avoid global variables, avoid `new`, avoid `==`, avoid `eval()`

Avoid Global Variables

Minimize the use of global variables.

This includes all data types, objects, and functions.

Global variables and functions can be overwritten by other scripts.

Use local variables instead, and learn how to use [closures](#).

Always Declare Local Variables

All variables used in a function should be declared as **local** variables.

Local variables **must** be declared with the `var`, the `let`, or the `const` keyword, otherwise they will become global variables.

Initialize Variables

It is a good coding practice to initialize variables when you declare them.

This will:

- Give cleaner code
- Provide a single place to initialize variables
- Avoid undefined values

```
// Declare and initiate at the beginning
let firstName = "";
let lastName = "";
let price = 0;
let discount = 0;
let fullPrice = 0,
const myArray = [];
const myObject = {};
```


Declare Objects with **const**

Declaring objects with `const` will prevent any accidental change of type:

Example

```
let car = {type:"Fiat", model:"500", color:"white"};  
car = "Fiat";      // Changes object to string
```

```
const car = {type:"Fiat", model:"500", color:"white"};  
car = "Fiat";      // Not possible
```

Declare Arrays with **const**

Declaring arrays with `const` will prevent any accidental change of

Example

```
let cars = ["Saab", "Volvo", "BMW"];  
cars = 3;    // Changes array to number
```

```
const cars = ["Saab", "Volvo", "BMW"];  
cars = 3;    // Not possible
```

Don't Use new Object()

- Use `""` instead of `new String()`
- Use `0` instead of `new Number()`
- Use `false` instead of `new Boolean()`
- Use `{}` instead of `new Object()`
- Use `[]` instead of `new Array()`
- Use `/()/` instead of `new RegExp()`
- Use `function (){}` instead of `new Function()`

Use === Comparison

The `==` comparison operator always converts (to matching types) before comparison.

The `===` operator forces comparison of values and type:

Example

```
0 == "";           // true
1 == "1";          // true
1 == true;         // true

0 === "";          // false
1 === "1";         // false
1 === true;        // false
```

Try it Yourself »

Use Parameter Defaults

If a function is called with a missing argument, the value of the missing argument is set to `undefined`.

Undefined values can break your code. It is a good habit to assign default values to arguments.

Example

```
function myFunction(x, y) {  
  if (y === undefined) {  
    y = 0;  
  }  
}
```

Try it Yourself »

ECMAScript 2015 allows default parameters in the function definition:

```
function (a=1, b=1) { /*function code*/ }
```

Avoid Number, String, and Boolean as Objects

Always treat numbers, strings, or booleans as primitive values. Not as objects.

Declaring these types as objects, slows down execution speed, and produces nasty side effects:

Example

```
let x = "John";  
let y = new String("John");  
(x === y) // is false because x is a string and y is an object.
```

[Try it Yourself »](#)

Or even worse:

Example

```
let x = new String("John");  
let y = new String("John");  
(x == y) // is false because you cannot compare objects.
```

[Try it Yourself »](#)

JavaScript Common Mistakes

[< Previous](#)[Next >](#)

This chapter points out some common JavaScript mistakes.

Accidentally Using the Assignment Operator

This `if` statement returns `true` (maybe not as expected), because `10` is true:

```
let x = 0;  
if (x = 10)
```

[Try it Yourself »](#)

This `if` statement returns `false` (maybe not as expected), because `0` is false:

```
let x = 0;  
if (x = 0)
```

[Try it Yourself »](#)

Expecting Loose Comparison

In regular comparison, data type does not matter. This `if` statement returns true:

```
let x = 10;  
let y = "10";  
if (x == y)
```

[Try it Yourself »](#)

In strict comparison, data type does matter. This `if` statement returns false:

```
let x = 10;  
let y = "10";  
if (x === y)
```

[Try it Yourself »](#)

Misunderstanding Floats

All numbers in JavaScript are stored as 64-bits **Floating point numbers** (Floats).

All programming languages, including JavaScript, have difficulties with precise floating point values:

```
let x = 0.1;
let y = 0.2;
let z = x + y           // the result in z will not be 0.3
```

Try it Yourself »

To solve the problem above, it helps to multiply and divide:

Example

```
let z = (x * 10 + y * 10) / 10;    // z will be 0.3
```

Try it Yourself »

All programming languages, including JavaScript, have difficulties with precise floating point values:

0.30000000000000004

Misplacing Semicolon

Because of a misplaced semicolon, this code block will execute regardless of the value of x:

```
if (x == 19);  
{  
    // code block  
}
```

Try it Yourself »

Breaking a Return Statement

It is a default JavaScript behavior to close a statement automatically at the end of a line.

But, what will happen if you break the return statement in two lines like this:

Example 4

```
function myFunction(a) {  
  let  
  power = 10;  
  return  
  a * power;  
}
```

Try it Yourself »

The function will return `undefined` !

Why? Because JavaScript thought you meant:

Accessing Arrays with Named Indexes

Example:

```
const person = [];  
person["firstName"] = "John";  
person["lastName"] = "Doe";  
person["age"] = 46;  
person.length;           // person.length will return 0  
person[0];               // person[0] will return undefined
```

Undefined is Not Null

JavaScript objects, variables, properties, and methods can be `undefined`.

In addition, empty JavaScript objects can have the value `null`.

Correct:

```
if (typeof myObj !== "undefined" && myObj !== null)
```

Try it Yourself »