

# Proyecto de Software

Conceptos básicos

Cursada 2025

**Antes de empezar, ¿qué pensamos sobre el desarrollo web?**



# ¿Qué abordaremos en esta clase?

- Lineamientos básicos a tener en cuenta
- Conceptos básicos
- Accesibilidad web
- Primera actividad de teoría

# Lineamientos básicos

## Usaremos y desarrollaremos con software libre

El **software libre** es el software donde los usuarios tienen la **libertad** para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software.

**El acceso al código fuente es una condición necesaria.**

La definición oficial pueden encontrarla en el [sitio del proyecto GNU](#)

# Lineamientos básicos

## Utilizaremos git y Gitlab para el desarrollo

**Git** es un **sistemas de control de versiones** distribuido libre diseñado para manejar proyectos con velocidad y eficiencia.

**GitLab** es una aplicación opensource que nos permite administrar repositorios en git mediante una interfaz web.

Este tema se explicará en detalle la clase siguiente.

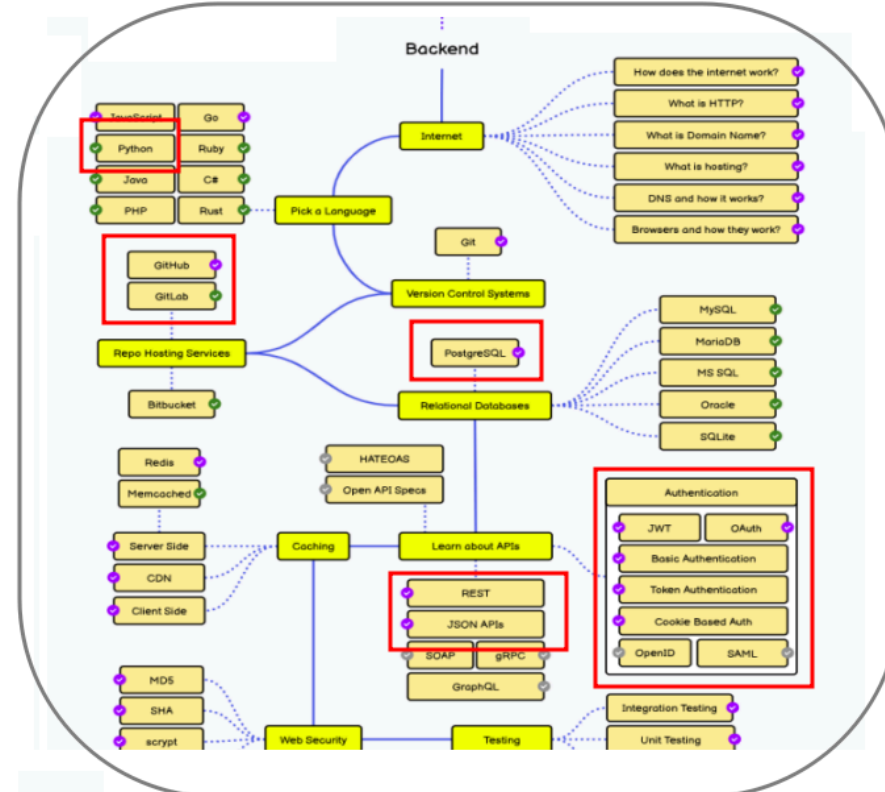
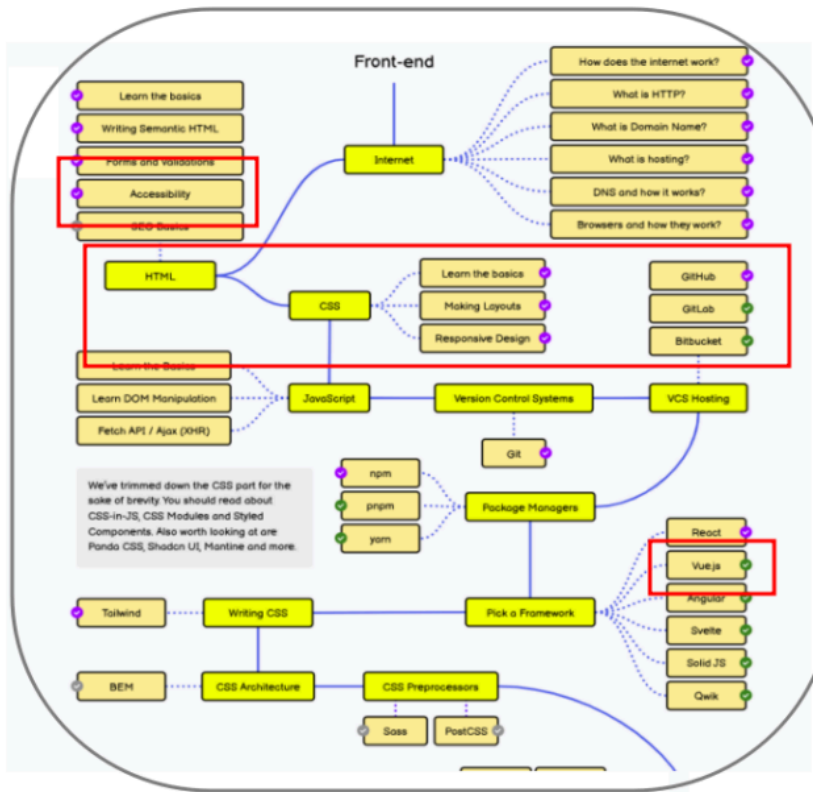
Es requisito en nuestra cursada el **trabajo en grupo**. El uso de git es una ENORME ventaja.

# Objetivo general de la materia

Desarrollar una aplicación web integrando distintas herramientas y tecnologías.

Programaremos en **Python** y **JavaScript**.

# Desarrollo web: ¿qué involucra?

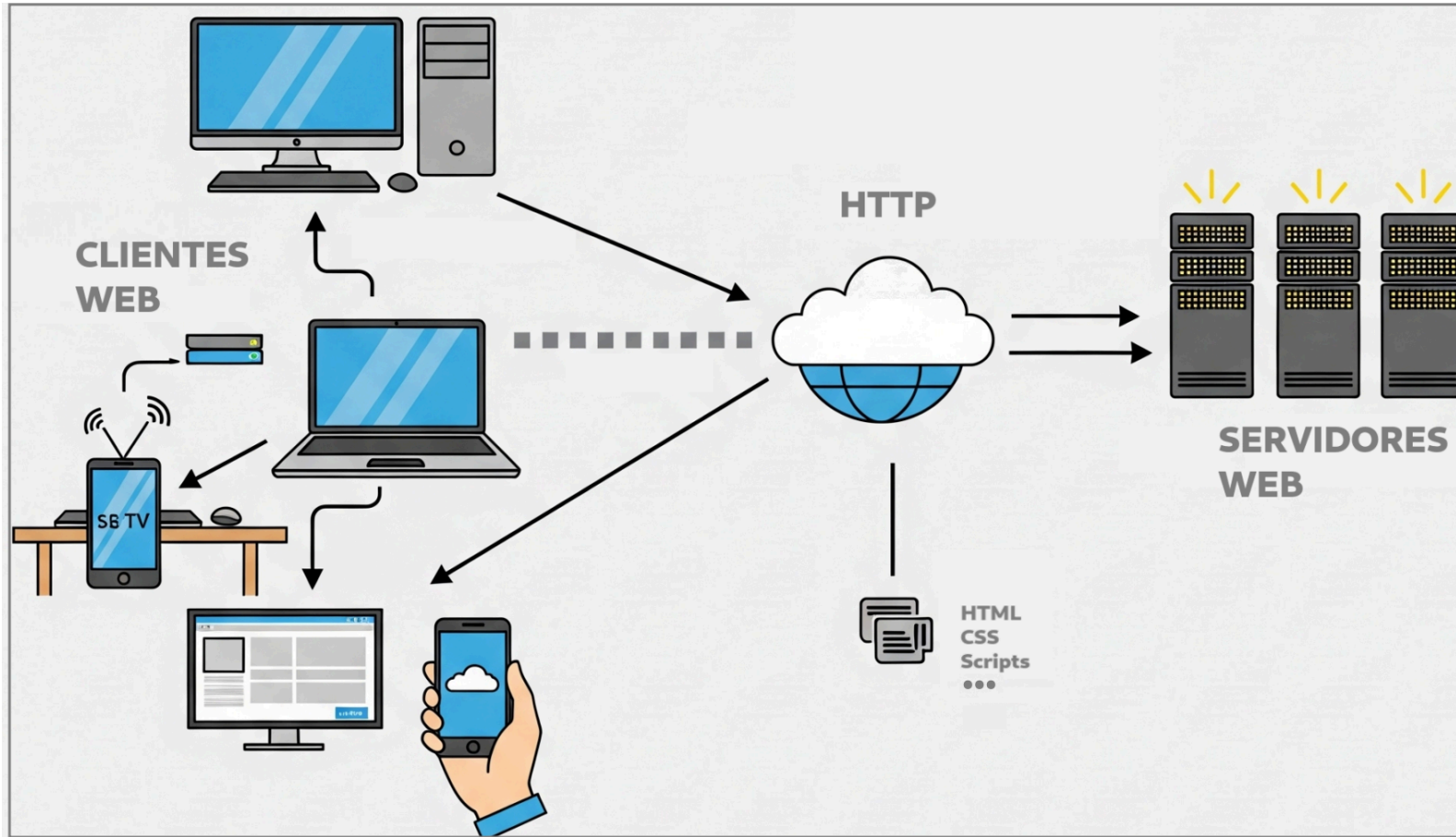


Fuente: <https://roadmap.sh>

**Empecemos por lo más simple...**

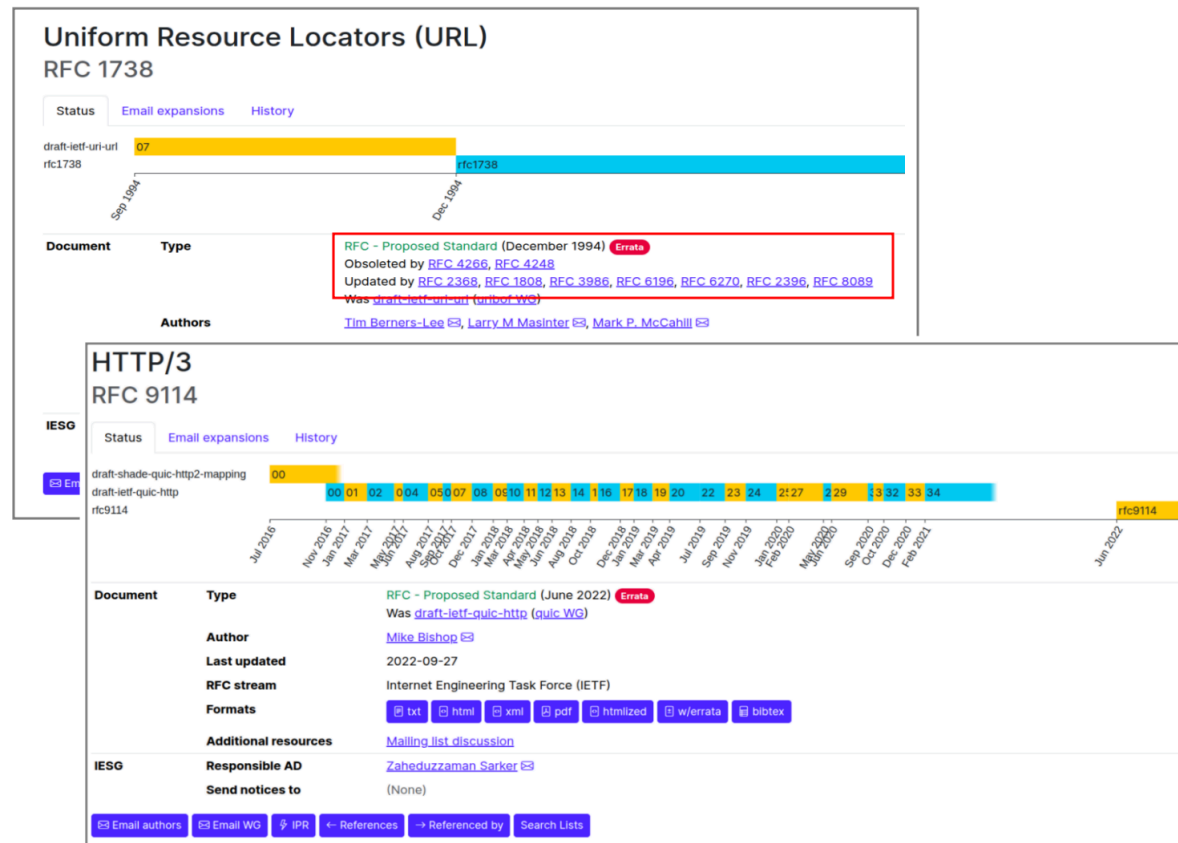


# La arquitectura web básica



# Especificaciones públicas

## Internet Engineering Task Force



RFC: Request For Comments

## W3C: el consorcio de la web

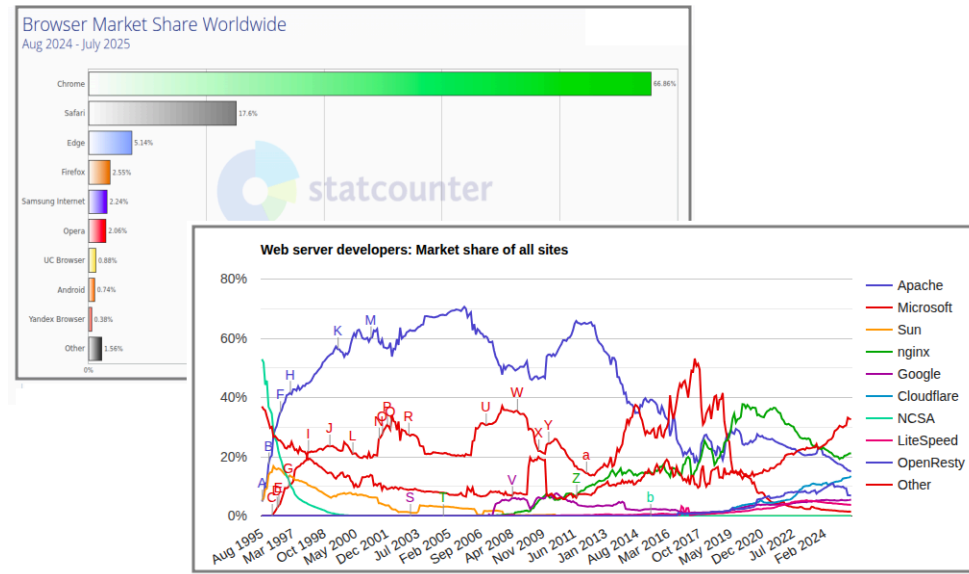
El consorcio de la web se encarga del desarrollo de estándares y guías que aseguren el crecimiento futuro de la web.

# ¡Muchas tecnologías!

Si tenemos que decidir qué tipo de desarrollo proponer o con qué tecnologías trabajar, **¿por dónde empezamos?, ¿qué deberíamos analizar?**

Podemos analizar algunas tendencias y estadísticas.

# Algunas estadísticas



- Sobre servidores
- Sobre navegadores
- Estadísticas de Stackoverflow
- Y otras similares ...

## Para analizar

- Estas estadísticas nos dan un panorama del contexto.
- También nos permiten analizar las tendencias y características de usos.

## Decidir qué tecnologías usar...

- Podemos retomar esto al final de la cursada ...

# La World Wide Web

- Más de 30 años. [A Proposal](#), Tim Berners-Lee, CERN
- En sus inicios:
  - Un mecanismo para identificar todo recurso accesible en la web: **URL/URI**
  - Un protocolo simple: **HTTP**
  - Un lenguaje para definir hipertextos: **el lenguaje HTML**
- Hoy:
  - Involucra numerosas tecnologías.

# Uniform Resource Identifier (URI)

Especificaciones: [RFC 3986](#) y [RFC 8820](#)

## Ejemplos típicos

```
http://www.servidor.com.ar/especificacion#parte3  
https://www.taller.com.ar/info.php?id=12&qq=11  
mailto:proyecto@info.unlp.edu.ar  
file:///home/claudia/git/proyecto2014/2020/teorias/clase2/index.html
```

## URI encoding

- Las URIs se transmiten en ASCII.
- Algunos caracteres deben convertirse.

```
../Clase%201/EjemplosClase1/Ejemplo%20con%20enlaces.html
```



# El protocolo HTTP

- HTTP nació como un protocolo muy simple:
  - Cada transacción HTTP consta de 4 pasos: **inicio conexión - solicitud - respuesta - cierre conexión**
- Si bien **evolucionó a lo largo de los años** pero algo no cambió:

HTTP es un **protocolo sin estado**

¿Qué significa esto?

# Solicitudes y respuestas

Las **solicitudes** más comunes:

- **GET**: retorna la información identificada por la URI-solicitada.
- **HEAD**: retorna sólo los encabezados, sin el cuerpo de la respuesta.
- **POST**: se utiliza para el envío de datos al servidor. Estos datos se incluyen en el cuerpo en el mensaje.

Y sus posibles **respuestas**:

- El servidor retorna un código que indica el estado de la solicitud. Por ejemplo, 200 y el recurso, o "el famoso" 404.
- Más info sobre [solicitudes HTTP](#)

**Retomaremos esto al abordar el desarrollo de API REst.**

# El Lenguaje HTML

```
<!DOCTYPE html>
<html lang="es">

<head>
<title>Título</title>
</head>

<body>
  <h1>Proyecto de Software</h1>
  <h2>Reglamento de cursada</h2>
</body>

</html>
```

La última versión en [Web Hypertext Application Technology Working Group](#)

# HTML básico - 1 -

¿Qué indica la cláusula **doctype**?

```
<!DOCTYPE html>
```

Los elementos pueden tener atributos.

```
<html lang="es">
```

# HTML básico - 2 -

- ¿Qué elementos contiene el encabezado? ¿Y el cuerpo?

```
<!DOCTYPE html>
<html lang="es">

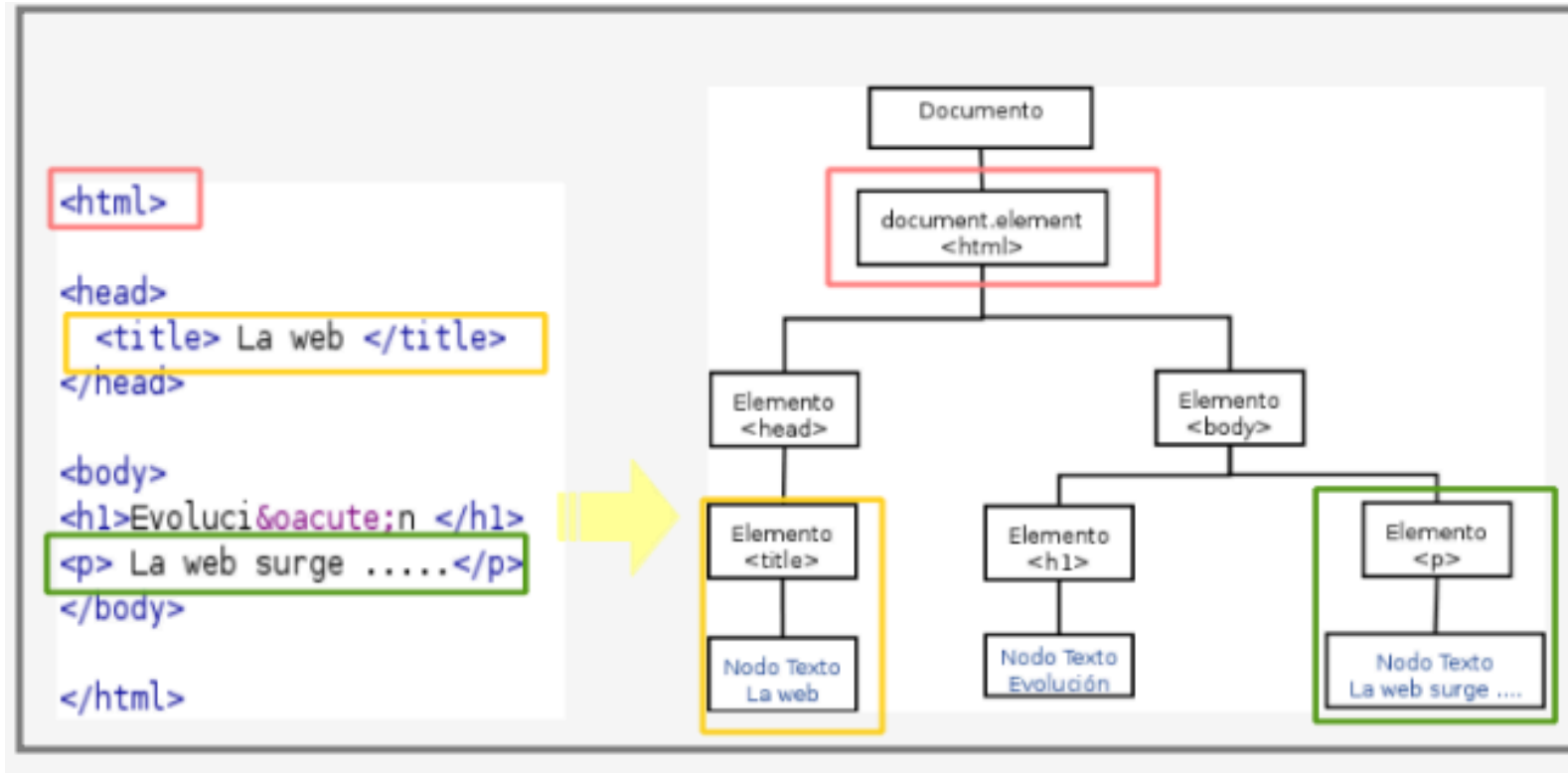
<head>
<title>Título</title>
</head>

<body>
  <h1>Proyecto de Software</h1>
  <h2>Reglamento de cursada</h2>
  Aquí va el texto del reglamento
</body>

</html>
```

- Más info: [Tutorial HTML & CSS Is Hard](#)

# DOM: el modelo de objetos del documento



# DOM: el modelo de objetos del documento

El modelo de objetos del documento es una **API**, que permite acceder a los contenidos de un documento HTML.

- Proporciona una **representación** estructurada, **orientada a objetos**, de los elementos individuales y el contenido del documento, con métodos para recuperar y fijar las propiedades de dichos objetos.
- Proporciona métodos para agregar y eliminar objetos al documento.
- También proporciona una **interfaz estándar** para trabajar con **eventos**.
- La especificación en: <https://dom.spec.whatwg.org/>

# DOM: el modelo de objetos del documento

Entre otras cosas, define una forma de acceder a un elemento a través del atributo **id**.

```
<h1>Proyecto de Software</h1>  
<h1 id="reglamento">Reglamento de cursada</h1>
```

```
document.getElementById( 'reglamento' )
```

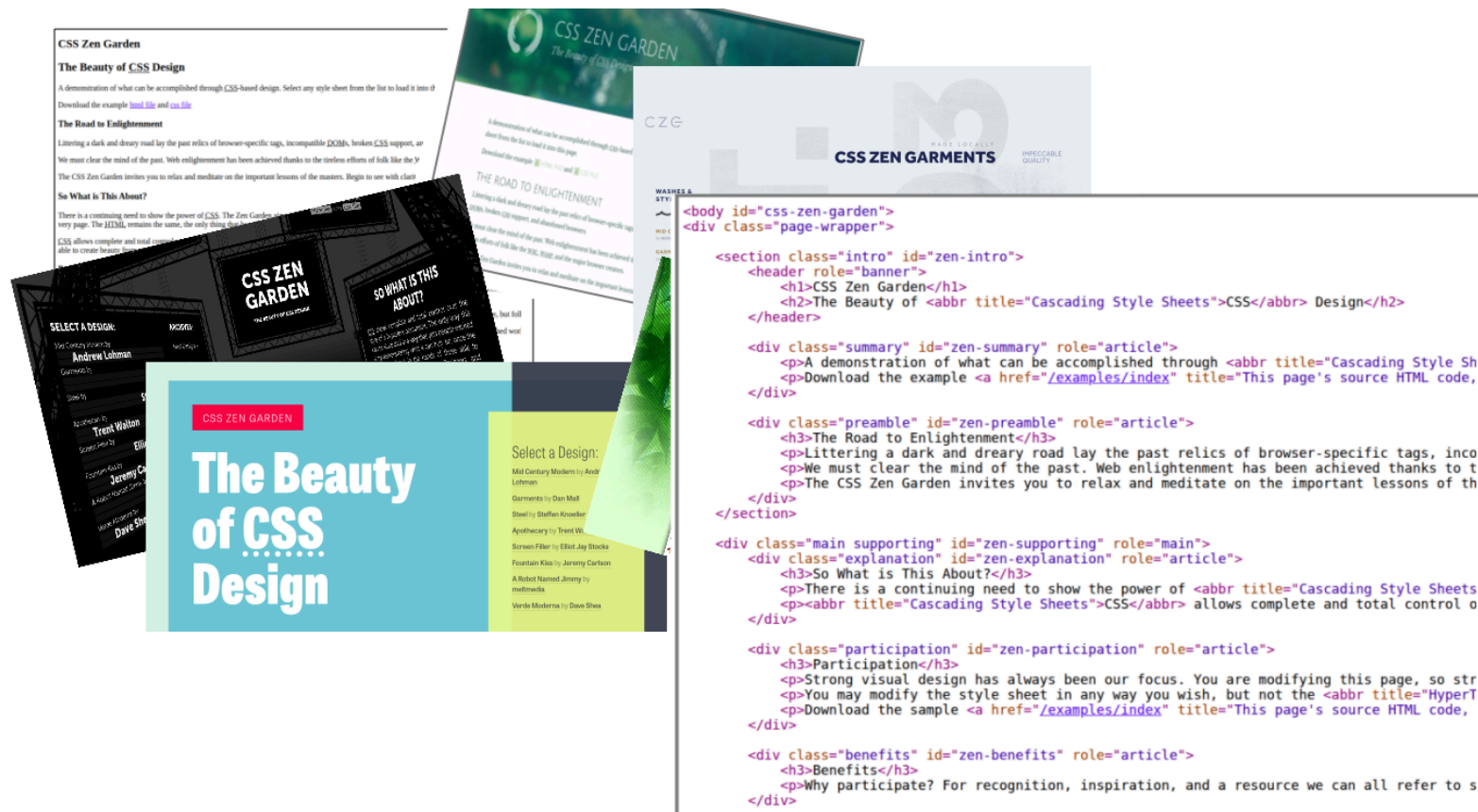
Retomaremos esto con mayor profundidad.



# Hojas de Estilo: CSS (Cascading Style Sheets)

- El documento tiene una estructura y una forma de visualización
  - **Estructura:** usando las etiquetas HTML más apropiadas.
  - **Visualización:** usando hojas de estilo
- Ver sitio [CSS Zen Garden](#)

# CSS para la visualización



# CSS básico - 1 -

## Reglas: selectores y propiedades

```
p {  
    color: red;  
    font-size: 14px;  
}  
ol, ul {  
    color: blue;  
    font-size: 18px;  
}  
ol li {  
    color: blue;  
}
```

- ¿A qué elementos estamos especificando estilos?
- ¿Cuáles son los selectores y cuáles las propiedades?

# CSS básico - 2 -

## Selectores

selector	Descripción
a	Un elemento
a, b, c	Varios elementos
a b	Elementos descendientes
a > b	Elemento hijo directo
a + b	Elementos adyacentes
a:last-child	Último elemento a
...	...

# CSS básico - 3 -

## ¿Dónde se ubican las reglas?

```
<!DOCTYPE html>
<html lang="es">

  <head>
    <title>Formularios web</title>
    <link rel="stylesheet" href="estilo.css">
    <style>
      h1 {color:red}
    </style>
  </head>
  <body>
    <h1 style="color:red"> Mi color favorito</h1>
  </body>
</html>
```

# CSS básico - 4 -

## ¿Cómo se aplican las reglas?

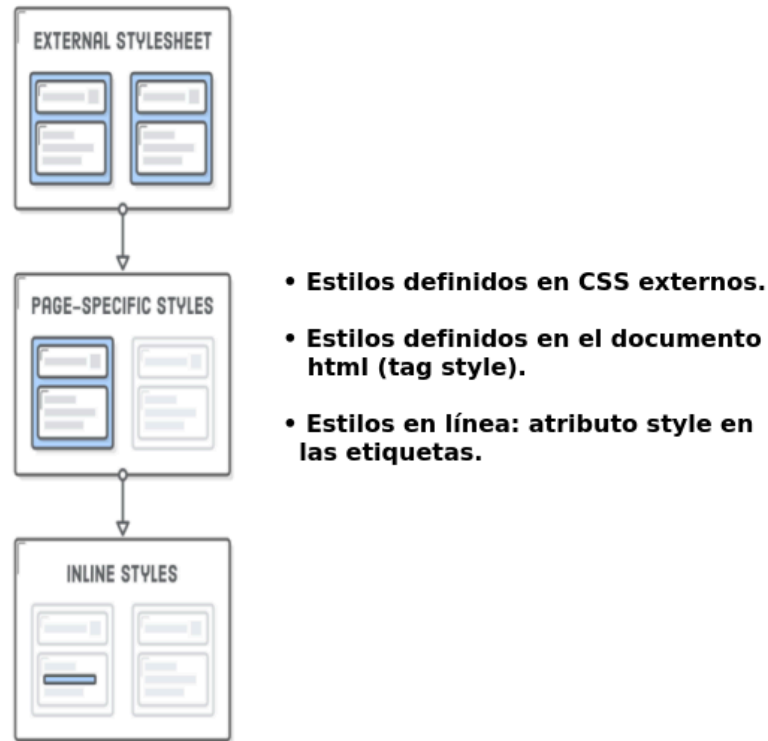


Imagen sacada de: <https://internetingishard.com>

# Tarea: ¿de qué color se muestra el texto "Mi color favorito"?

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Jugando con estilos</title>
  <style>
    h1 {color:blue}
  </style>
  <link rel="stylesheet" href="estilo.css">
</head>
<body>
  <h1 style="color:red"> Mi color favorito</h1>
</body>
</html>
```

Donde en estilo.css, hay una regla para h1 asociando el color verde.

# Diseño responsivo - 1 -

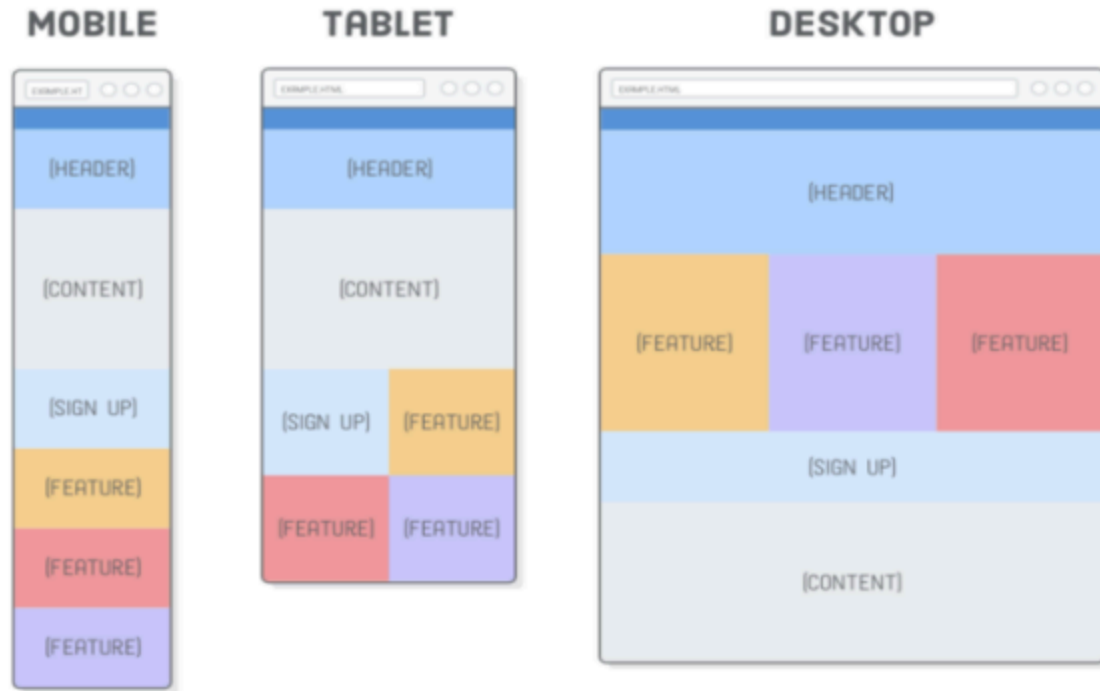


Imagen obtenida de: <https://www.internetingishard.com/html-and-css/responsive-design/>



# Diseño responsivo - 2 -

## ¿Qué es el viewport?

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```



El asentamiento de Petra se localiza en un



El asentamiento de Petra se localiza en un valle angosto, al este del valle de la Aravá que se extiende desde el mar Muerto hasta el Golfo de Aqaba. Los restos más célebres de Petra son sin duda sus construcciones labradas en la misma roca del valle (hemispeos), en particular, los edificios conocidos como el Khazneh (el Tesoro) y el Deir (el Monasterio). Fundada en la antigüedad hacia el final de siglo VIII a. C. por los edomitas, fue ocupada en el siglo VI a. C. por los nabateos que la hicieron prosperar gracias a su situación en la ruta de las caravanas que llevaban el incienso, las especias y otros productos de lujo entre

# Diseño responsivo - 3 -

## Media queries

Permiten definir estilos específicos dependiendo del medio donde se mostrará la página.

- Veamos [ejemplo\\_media\\_queries.html](#)
- Es posible utilizar operadores para hacer más específica la regla:

```
@media (min-width: 700px) and (orientation: landscape) { ... }
```

- Se pueden hacer cosas interesantes como: media types: **braille**, print, projection, screen, **speech**, tv, etc.
- Más info en <https://developers.google.com>

# Hay muchas herramientas

- [Bootstrap](#)
- [Materialize](#)
- [Bulma](#)
- Y muchos otros ...

¿Qué son? ¿Usaron?

**Hagamos unas pruebas ...**

**¿Saben qué son las WCAG?**

¿Qué pasa cuando accedemos a la web con la pantalla apagada?

¿Qué pasa si no disponemos de un mouse? ¿O un teclado?

Probemos algo...

- [Sitio educ.ar](#)
- [Sitio Diario El Día](#)

# Algunas reflexiones

- OMS: discapacidad y salud - Datos marzo-2023
  - Según la OMS, **1300 millones de personas** (es decir, 1 de cada 6 personas en todo el mundo) sufren una discapacidad importante.
- OMS: ceguera y discapacidad visual - Datos agosto-2023
  - En el mundo hay al menos **2200 millones de personas** con deterioro de la visión cercana o lejana.

# Accesibilidad web

"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect".

Tim Berners-Lee, W3C

# Algunos videos

Tarea: veamos el siguiente [video introductorio a aspectos de accesibilidad web](#)

## Perspectivas





# ¿Qué involucra pensar en la accesibilidad?

- Hay personas que tienen dificultad para leer o comprender textos.
- Hay personas que no pueden ver, escuchar o moverse.
- No todos utilizan mouse y teclados.
- Conexión a Internet lenta.
- Se trabaja con tecnologías muy diversas.
- Pantallas de sólo texto, pequeñas o de distintas resoluciones.

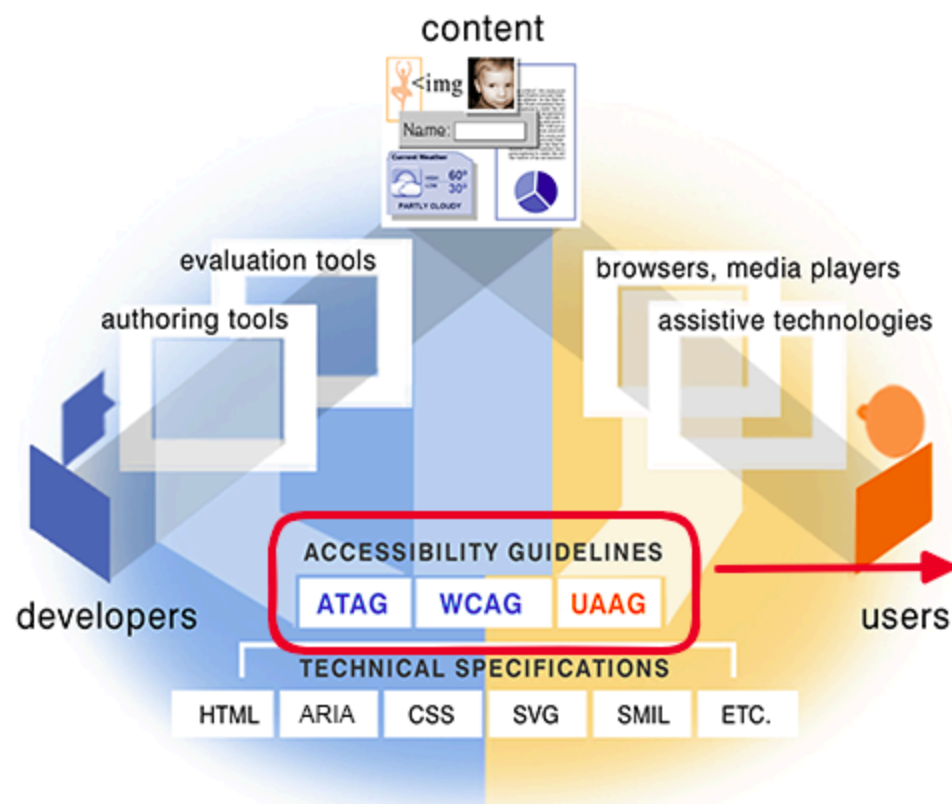
# WAI: Web Accessibility Initiative

## Objetivos

Desarrollar estrategias, pautas, recursos para hacer la Web accesible a **personas con discapacidad**. Pero también será accesible en **otros entornos y aplicaciones**, como navegador de voz, teléfono móvil, PC de automóvil; y ante limitaciones bajo las que opere, como entornos ruidosos, habitaciones infra o supra iluminadas, entorno de manos libres.

URL: <https://www.w3.org/WAI/>

# WAI: ¿qué involucra?



ATAG: pautas de accesibilidad para las herramientas de creación de contenido

UAAG: pautas de accesibilidad para el agente de usuario

WCAG: pautas de Accesibilidad para el Contenido Web

Sacado del [sitio de la WAI de la W3C](#)

# WAI: ¿qué involucra?

- Para aplicaciones de usuario: [WAI-User Agent Accessibility Guidelines \(UAAG\)](#)
- Para herramientas de autor: [WAI-Authoring Tool Accessibility Guidelines \(ATAG\)](#)
- Para contenido web: [Web Content Accessibility Guidelines \(WCAG\)](#)
  - WCAG 1.0
  - WCAG 2.0
  - WCAG 2.1
  - **WCAG 2.2 (2023)**

# WCAG: guías para contenido web accesible

- Definen **principios de diseño web**.
- Cada principio tiene **pautas**.
- Cada pauta tiene **criterios testeables**.
- Cada criterio tiene **nivel de conformidad: A, AA y AAA**.

# Principios de diseño

- **1- Perceptible:** la información y los componentes de la interfaz de usuario deben ser mostrados a los usuarios en formas que ellos puedan entender.
- **2- Operable:** los componentes de la interfaz de usuario y la navegación debe ser manejable.
- **3- Comprensible:** la información y las operaciones de usuarios deben ser comprensibles.
- **4- Robustez:** el contenido deber ser suficientemente robusto para que pueda ser bien interpretado por una gran variedad de agentes de usuario, incluyendo tecnologías de asistencia.

# Cumplir pautas principio 1 - Perceptible

- Proporcionar **alternativas textuales** para contenido no textual.
- Proporcionar **subtítulos y otras alternativas** para multimedia.
- Crear contenido que se pueda **presentar de diferentes formas**, incluyendo a las tecnologías de apoyo, sin perder información.
- Facilitar que los usuarios puedan **ver y oír el contenido**.

# Cumplir pautas principio 2 - Operable

- Proporcionar acceso a todas las funcionalidades **mediante el teclado**.
- Conceder a los usuarios **tiempo suficiente** para leer y usar el contenido.
- No usar contenido que pudiera causar **convulsiones** o reacciones físicas.
- Ayudar a los usuarios a **navegar y encontrar el contenido**.
- Facilitar **métodos de entrada diferentes al teclado**.



# Cumplir pautas principio 3 - Comprensible

- Proporcionar texto **legible y comprensible**.
- Proporcionar contenido que sea **predecible** en apariencia y operación.
- Ayudar a los usuarios a **evitar y corregir errores**.

## Cumplir pautas principio 4 - Robustez

- Maximizar la **compatibilidad** con herramientas de usuario actuales y futuras.

**Y si hasta acá no vimos los beneficios de proporcionar una interfaz accesible ....**

# Marco legal

En distintos países **existen normativas y leyes** que obligan a que sitios web y apps nativas sean accesibles.

# En Argentina: Ley Nacional de Accesibilidad

La [ley N° 26.653](#) de "Accesibilidad Web" busca promover la accesibilidad de la información, facilitando especialmente el acceso a todas las personas con discapacidad con el objeto de **garantizarles la igualdad real de oportunidades y trato**, evitando así todo tipo de discriminación.

Sacado de: <https://www.argentina.gob.ar/jefatura/innovacion-publica/onti/accesibilidad-web>

# La ley comprende:

- El Estado nacional, y sus organismos descentralizados o autárquicos.
- Entes públicos no estatales.
- Empresas del Estado.
- Empresas privadas concesionarias de servicios públicos.
- Empresas prestadoras o contratistas de bienes y servicios.
- Instituciones u organizaciones de la sociedad civil que sean beneficiarias o reciban subsidios, donaciones o condonaciones, por parte del Estado o celebren con el mismo contrataciones de servicios.

## Estándares de Accesibilidad



Material elaborado por la Prof. Ivana Harari, en su presentación "Introducción a la accesibilidad digital".

# ¿Cómo validamos?: chequeo inicial

La W3C brinda un [check list](#) para revisar rápidamente la accesibilidad.

Título de la página

Textos alternativos

[Imágenes y videos](#)

Textos

[Encabezados, contrastes de color y redimensiones](#)

Interacción

[Acceso por teclado, foco visual, formularios, errores, etc.](#)

General

[Contenido en movimiento o parpadeante, alternativas multimedia \(audio y video\), Verificación de estructura básica.](#)



# ¿Cómo validamos?: evaluación manual

## Evaluar el sitio en forma manual

- El sitio debe ser evaluado teniendo en cuenta distintos escenarios de interacción, comprobando su estabilidad y acceso.
- En diferentes circunstancias, el contenido y sus servicios deben poder accederse :

Con teclado solamente

Con lectores de pantalla

Con o sin imágenes

Con o sin color

Con y sin tablas

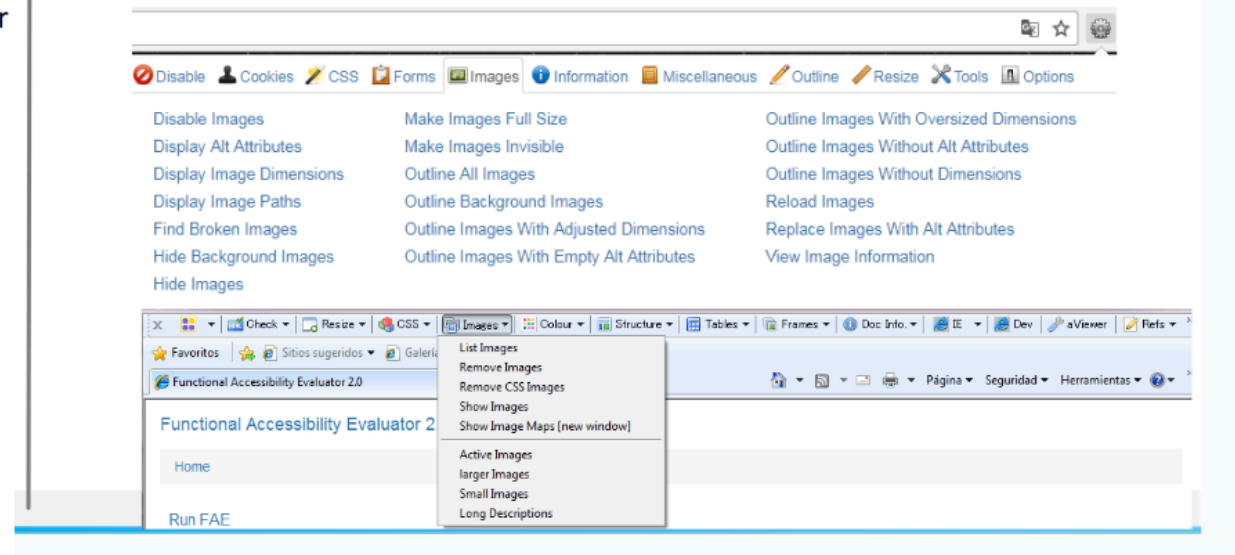
Con y sin hojas de estilo

Con y sin javascript

Otras configuraciones

## Evaluar el sitio en forma manual

- Hay complementos para los navegadores que permiten configurar la página mediante distintos seteos para analizar cómo se comporta la misma.



Material elaborado por la Prof. Ivana Harari, en su presentación "Introducción a la accesibilidad digital".

# ¿Cómo validamos?: con usuarios finales

## Evaluar el sitio con usuarios finales

- Es necesario testear el sitio con usuarios con discapacidad para analizar si es posible el acceso a toda la información.
- Trabajar con distintos perfiles del usuario, incluyendo adultos mayores, para observar su interacción y costos para llevar a cabo las tareas.
- Se debe definir la muestra, definir las actividades a realizar, diseñar la planilla de registro y luego analizar los resultados.

Material elaborado por la Prof. Ivana Harari, en su presentación "Introducción a la accesibilidad digital".

# Validadores de HTML y CSS

- Validador HTML: <http://validator.w3.org/>
- Validador de Hojas de Estilos: <http://jigsaw.w3.org/css-validator/>
- NU: <https://validator.w3.org/nu/>

# Validadores de accesibilidad web

- WAVE: <http://wave.webaim.org>
- TAW: <http://www.tawdis.net>
- Tingtun Checker: <http://checkers.eiii.eu/>
- [Lighthouse](#)
- Herramientas de la W3C: <https://www.w3.org/WAI/ER/tools/>

# Curso de accesibilidad web de la facultad



Curso a distancia  
**Accesibilidad Web**  
*XIII Edición*

**Gratuito / Cupos limitados**

**Inicio:** Lunes 13 / 9 . **Finaliza:** Sábado 14 / 11

**Inscripción en:** <https://actividades.linti.unlp.edu.ar>

**Contacto:** [accesibilidad@info.unlp.edu.ar](mailto:accesibilidad@info.unlp.edu.ar)

Abierto a la comunidad / Se entregarán certificados

+Info

# Primeras actividades

# Primera autoevaluación

- Dejamos un cuestionario de [Autoevaluación 1](#) relacionado estos temas.
- Dejamos un material grabado de años anteriores que incluye:
  - Una breve [introducción](#)
  - Una primera aproximación a [HTML](#) y [CSS](#).

# ACT 1 DE TEORIA

Está disponible en [catedras.linti](https://catedras.linti)

Dejamos una [guía](#)

**Solo podrán hacer la Parte II** quienes hayan realizado la Parte I

**Parte I:** desde el viernes 22 de agosto hasta el miércoles 27 de agosto - 23:59hs.

**Parte II:** desde el jueves 28 de agosto hasta el lunes 1 de septiembre - 23:59hs.



# Algunas referencias

- [Principios WAI](#)
- [Componentes WAI](#)
- [Descripción general de las WCAG 2](#)
- [Accessible Rich Internet Applications \(ARIA\)](#)
- [All the tags](#)
- [Curso HTML and CSS - Lecture 0 - CS50's Web Programming with Python and JavaScript](#)

**Seguimos la próxima ...**