



escuela**arte**granada

TEMA 3: OPTIMIZACIÓN DE PÁGINAS WEB

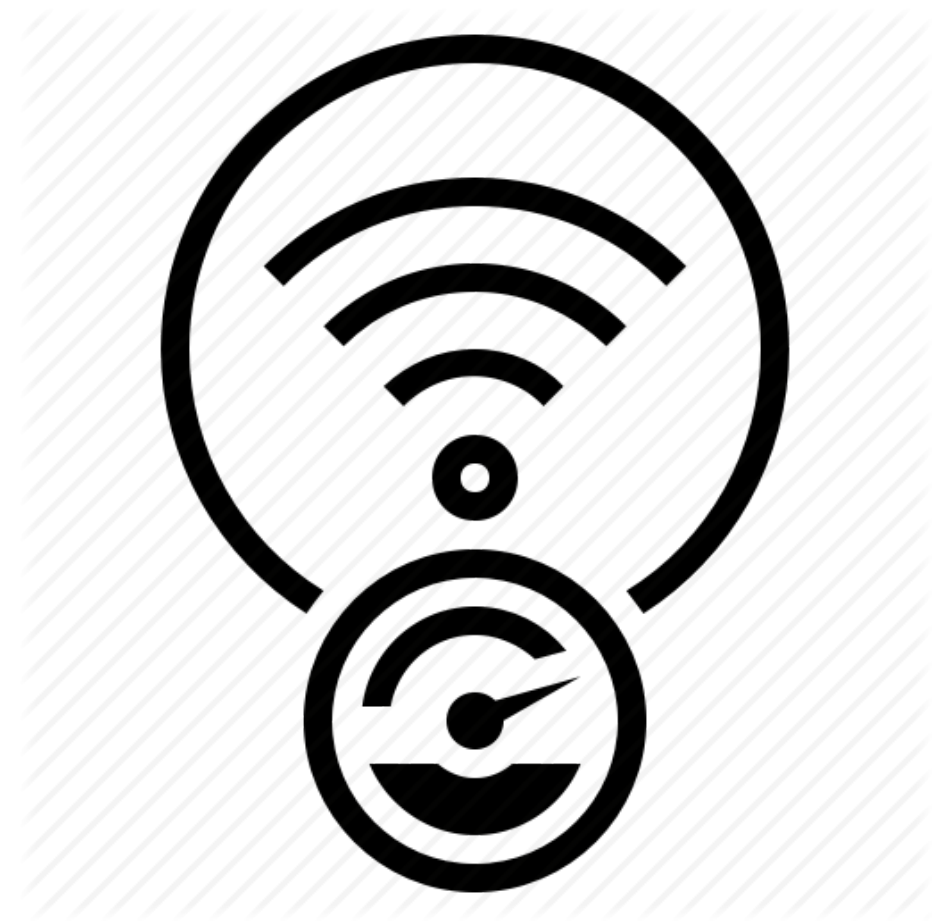
Despliegue de Aplicaciones Web
2º Desarrollo de Aplicaciones Web
Curso 2018/2019 – Profesor: Borja Molina Zea

Índice

- Apache
 - Httpd.conf
 - DocumentRoot
 - Alias_module
 - DirectoryIndex
 - Códigos de error
- PHP
 - Php.ini
 - Reporte de errores
 - Fechas
 - Tamaño de los ficheros
 - Límites de memoria y tiempo de ejecución

Tiempos de carga

- Es importante medir y optimizar los tiempos de carga de la webs.
- Una de las variables que más influye en la experiencia de uso es el tiempo de carga de una web



Fuentes

- Las fuentes pueden llegar a consumir gran parte del tiempo que nuestra página tarda en cargar, por lo tanto se hace necesario optimizar su uso.
- Para ello especificamos el valor `font-display` en nuestro CSS. Posibles valores: `auto`, `block`, `swap`, `fallback` y `optional`

Navegador	Comportamiento
Chrome	Esconde el texto durante tres segundos, si pasados ese tiempo el texto sigue sin cargar entonces utiliza la fuente del sistema
Firefox	
Safari	Esconde el texto hasta que la fuente esté cargada
Edge	Nada más cargar utiliza la fuente por defecto y una vez que la fuente está cargada entonces la cambia.

Fuentes

- Valores de **font-display**:
 - **Auto**
 - **Block**: el texto se esconde durante un breve periodo de tiempo hasta que la fuente es cargada.
 - **Swap**: el navegador utiliza una fuente por defecto y una vez que la fuente está cargada la cambia.
 - **Fallback**: el navegador esconde el texto durante 100ms, si la fuente no ha terminado de cargar utiliza la fuente por defecto. El navegador volverá a preguntar por la carga de la fuente durante tres segundos.
 - **Optional**: el navegador decide, basado en la velocidad de conexión del usuario, si carga las fuentes externas o no.

Imágenes - WebP

- WebP es un formato de imagen desarrollado por Google y caracterizado por estar enfocado a la web debido a su escaso peso. Consigue una compresión alrededor del 25-35% con respecto JPEG y PNG
- Youtube mejoró los tiempos de carga un 10% al utilizar WebP. Facebook consiguió un ahorro en el tamaño de los ficheros de hasta un 35%.
- Para transformar las imágenes podemos utilizar el programa de línea de comandos cwebp:
 - <https://developers.google.com/speed/webp/docs/using>
 - <https://storage.googleapis.com/downloads.webmproject.org/releases/webp/index.html>

Imágenes - WebP

```
<picture>  
  <source type="image/webp" srcset="img/im1.webp">  
  <source type="image/jpeg" srcset="img/im1.jpg">  
    
</picture>
```

Di no a los GIFs

- Para mejorar los tiempos de carga es preferible utilizar videos en vez de gifs, los videos solo empiezan a cargar si el usuario hace click en ellos por el contrario los gifs cargan nada más entrar en la web.
- Para convertir un gif en video podemos utilizar herramientas como FFmpeg o conversores online como <https://ezgif.com/gif-to-mp4>
- Se puede simular el comportamiento de un gif añadiendo los atributos “autoplay loop muted playsinline” a la etiqueta del video

Carga perezosa de imágenes

- Los recursos pueden ser cargados de dos formas: en el momento que vayan a ser usado (carga perezosa o lazy load) o cargarlos cuando son declarados en vez de cuando van a ser usados.
- Mediante la librería lazysizes <https://github.com/aFarkas/lazysizes> podemos conseguir este comportamiento de una manera sencilla.
- Primero se cargan las imágenes que antes debe de ver el usuario (según el flujo secuencial del código HTML)
- Para usar la lazysizes:
 - Cargar la librería en el header y con el atributo async
 - Aplicar la clase lazyload a las imágenes
 - Utilizar el atributo data-src (en vez de src) para especificar la ruta

Imágenes responsivas

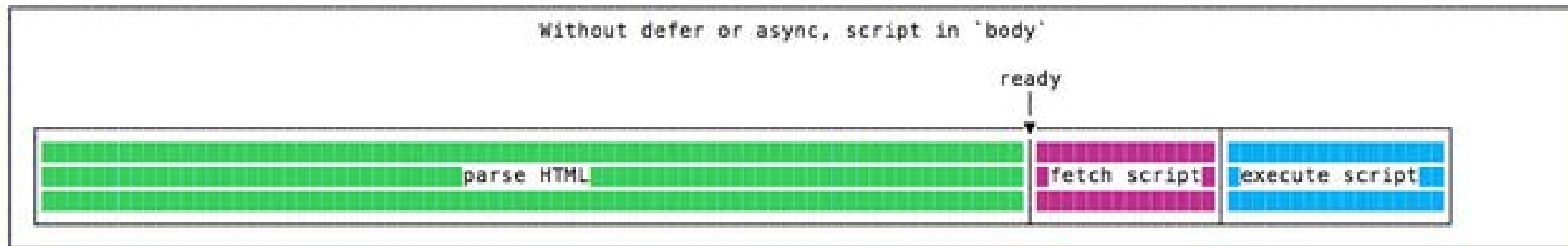
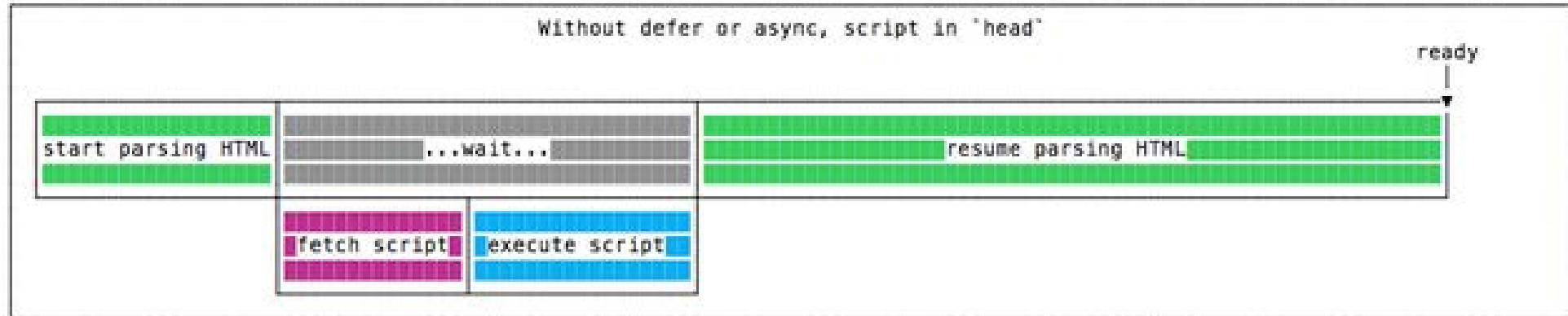
- En vez de tener una única imagen (de alta resolución) para todos los dispositivos podemos tener varias imágenes y servir aquella que se ajuste a la resolución del cliente, de esta forma clientes con resoluciones más bajas tardarán menos en descargar las imágenes y no perderán calidad.
- Para conseguir este comportamiento utilizamos el atributo `srcset` para especificar la lista de las distintas imágenes que podrá descargarse el navegador y éste en función de la resolución de pantalla se descargará la más adecuada.

Pre-carga de recursos críticos

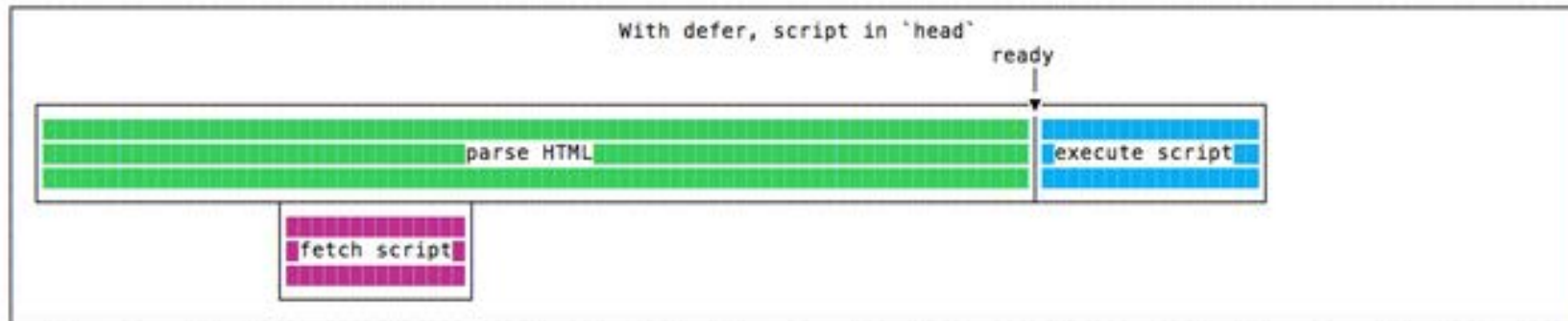
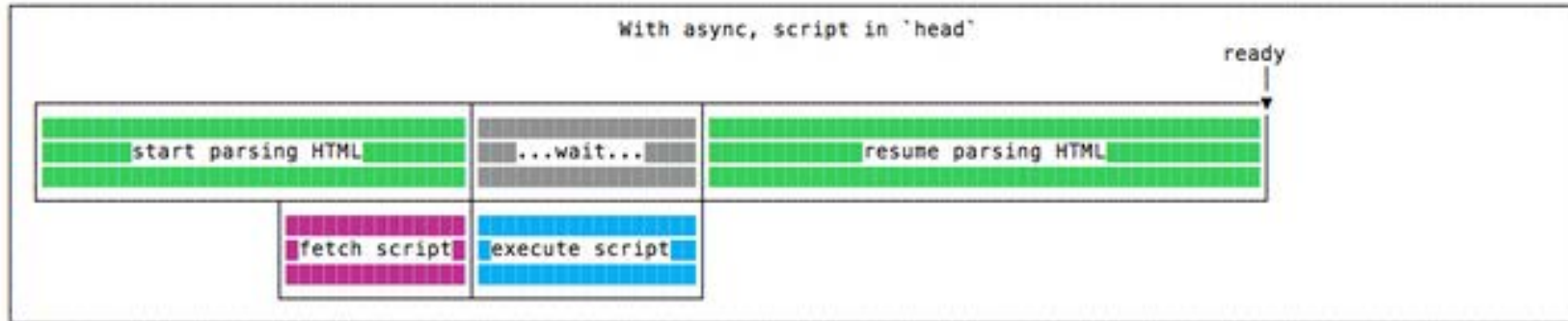
- A los recursos más importantes se les debe de aplicar `rel = "preload"` de forma que el explorador de prioridad a su carga sin retrasar el evento *window.onload*. Solo se puede especificar para los recursos que cargamos con la etiqueta `link`, normalmente usada para cargar hojas de estilos pero que podemos utilizar para cargar otro tipo de elementos como imágenes.
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Link_prefetching_FAQ

```
<link rel="preload" as="style" href="css/style.css">
```

Carga asíncrona



Carga asíncrona



SEO friendly

- La técnica que se preocupa por hacer que tu sitio web pueda ser encontrado y entendido de forma eficaz por los distintos motores de búsqueda recibe el nombre de **SEO**, de sus siglas en ingles *Search Engine Optimization*



Títulos y descripciones


- Los motores de búsqueda utilizan los título y descripciones de las webs para catalogarlas.
- Intenta ser preciso. Evitando el uso de palabras poco específicas.
- Utiliza la etiqueta `<title>` en todas las páginas
- Utiliza la etiqueta `<meta>` en todas las páginas
- No utilices el mismo título para distintas páginas.
- No hagas *keyword stuffing* o los buscadores podrían penalizarte

 **Not recommended** — too vague


```
<title>Donut recipe</title>  
<meta name="description" content="A donut recipe.">
```

 **Recommended** — precise!

```
<title>Mary's Maple Bar Fast-Baking Recipe</title>  
<meta  
  name="description"  
  content="Mary's maple bar recipe is simple and sweet,  
          with just a touch of serendipity. Topped with  
          bacon, this sticky donut is to die for.">
```


 **Not recommended** – too vague

```
<p>
  Get cooking and learn how to make a maple bar
  <a href="https://donut-be-crazy.com/recipes/maple-bar-recipe">
    here
  </a>.
</p>
```

 **Recommended** – precise!

```
<p>
  Get cooking and learn how to
  <a href="https://donut-be-crazy.com/recipes/maple-bar-recipe">
    make a delicious maple bar with Mary
  </a>!
</p>
```

Localización web

- Si tu web está en múltiple idiomas hazlo explícito en la cabecera

```
<!doctype html>
<html lang="en">
  <head>
    <title>Mary's Maple Bar Fast-Baking Recipe</title>
    <meta name="Description" content="Mary's maple bar
recipe is simple and sweet, with just a touch of
serendipity. Topped with bacon, this sticky donut
is to die for.">
    <link rel="alternate" hreflang="en"
href="https://donut-be-hangry.com/recipes/maple-bar-recipe"/>
    <link rel="alternate" hreflang="es"
href="https://es.donut-be-hangry.com/recipes/maple-bar-recipe" />
```

Una página dos enlaces

- En ocasiones podemos tener una misma página con dos direcciones, en estos casos deberos de hacer una de las direcciones canónica.

```
<!doctype html>
<html lang="en">
<head>
  <link rel="canonical" href="https://copycat.com/" />
  ...
```

Di no a los códigos de error

- Los buscadores penalizan aquellas páginas que devuelve códigos de error como 404 o 5XX, asegúrate que tu web esté disponible el máximo tiempo posible, de lo contrario no solo afectarás a la experiencia del usuario sino que los buscadores penalizarán tu web.

¡Oh, no!

No encontramos la página que buscas. Vuelve a comprobar la URL, utiliza el cuadro de búsqueda o vuelve a la página de inicio.

Ni Flash ni Applets

- No utilices animaciones Flash ni Applets de Java en tu web su uso a parte de anticuado hará que los motores de búsqueda califiquen negativamente tu web.



Robots.txt

- El fichero robots.txt le dice a los buscadores qué páginas de tu web debe indexar y cuales no.
- Definir robots.txt no es una tarea difícil pero si crucial, los buscadores pueden penalizar tu web si no comprenden tu fichero robots.txt
- robots.txt siempre debe estar situado en el fichero raíz de nuestro servidor y debe llamarse robots.txt (con la primera r en minúscula)

User-agent

- robots.txt está formado por bloques de directivas, cada directiva empieza con la linea user-agent
- user-agent determina a qué motor de búsqueda afectará la directiva, por ejemplo Googlebot, bingbot o DuckDuckBot
- Lista con los principales user-agent

User-agent

- Podemos especificar que una directiva afectará a todos los buscadores con user-agent: *
- Los buscadores escogieran siempre la directiva más específica, imaginemos que tenemos user-agent:* y user-agent: Googlbob entonces Google cogerá los parámetros de la segunda mientras que el resto de buscadores cogerán las de la primera
- Tras especificar el user-agent debemos añadir a nuestra directiva alguno de los siguientes comandos: Disallow, Allow, Sitemap y Crawl-delay

Disallow

- Especifica que carpetas y ficheros no deben de ser indexados

User-agent: *
Disallow:
Permite indexar todo el contenido a todos los navegadores

User-agent: bingbot
Disallow: /admin/
Niega indexar el contenido de la carpeta admin a user-agent de Bing

User-agent: *
Disallow: /
Niega todo el contenido a todos los buscadores

User-agent: *
Disallow: private.html
Niega indexar el fichero /private.html a cualquier buscador

Disallow

- Cada directorio requiere una nueva línea

```
User-agent: *  
Disallow: /cgi-bin/  
Disallow: /tmp/  
Disallow: /~joe/
```

* y \$

- A parte de para seleccionar todos los agentes podemos utilizar el asterisco para seleccionar todos los ficheros o directorios que empiecen con una determinada raíz.
- \$ nos permite dar permiso o no según el tipo de fichero

```
User-agent: *  
Disallow: /privado*/  
Especifica que no se indexe ningún  
directorio que empiece por "privado"
```

```
User-agent: *  
Disallow: *.svg$  
Quita permisos a todos los buscadores a  
todos los fichero .svg
```

Allow

- Funcionamiento contrario a Disallow
- Permite sobrescribir lo especificado por Disallow

User-agent: *

Disallow: /private/

Allow: /private/info.html

No indexa ningún contenido de la carpeta private excepto el fichero info.html de la misma

Crawl-delay y Sitemap

- **Crawl-delay** especifica el número de segundos que un buscador debe esperar entre página y página, útil para disminuir carga al servidor
- Finalmente la directiva Sitemap indica la ruta dónde se encuentra nuestro fichero en formato XML que determina la estructura de nuestra web
- Podemos ver ejemplos de ficheros robots.txt de cualquier página, por ejemplo: <https://www.escuelaartegranada.com/robots.txt>

Meta name robots

- También podemos utilizar la etiqueta meta para definir el comportamiento de los buscadores `<meta name="robots" content="VALORES">`
- En name debemos especificar el nombre de los buscadores (o robots para hacer referencia a todos)
- En content especificamos si permitimos o no la indexación y el rastreo Index o NoIndex y Follow o NoFollow

Meta name robots

- content = "Index, Follow". Es el comportamiento por defecto permite a los navegadores indexar nuestra web y seguir los links
- content = "NoIndex, Follow". No queremos que la página aparezca en los buscadores pero si las páginas a las que hace referencia.
- content = "Index, NoFollow". Queremos que la página aparezca en los navegadores pero no quieres que se relacionen dicha página con los enlaces que hay dentro de la misma.
- content = "NoIndex, NoFollow". Efecto similiar a Disallow utilizando robots.txt