ASCII

128 símbolos originalmente. Se han creado extensiones para cada propósito/lengua. Problema de que no es código único para todos los sitios.

UNICODE

UTF-8: Más usada y compleja, puesto que usa de 1 a 4 bytes.

* Los que pertenecen al ASCII ocupan 1 byte
* 2 bytes para algunas lenguas
* 3 bytes para símbolos de lenguas distintas como chino, japonés, etc
* 4 bytes para otros como símbolos matemáticos, lenguas muertas, etc.

ARCHIVOS BINARIOS

* Todos aquellos que no sean de texto plano. Son distintos para cada máquina. No puede ejecutar un .exe en Android.
* Ocupan menos espacio, ya que se optimiza la codificación a binario
* Más rápidos de manipular por tener un lenguaje más cercano al de la máquina.
* Acceso directo a los datos.
* Los datos no son fácilmente interpretables.
* Ideales para almacenar contenido cifrado por el uso de técnicas no textuales.

TEXTO PLANO

* Ideales para almacenar datos y exportar e importar estos a cualquier dispositivo electrónico.
* Directamente modificables, sin necesitar un software específico.
* Sencilla manipulación.
* No son tan susceptibles de contener virus informáticos.

**Hipertexto:** Posibilidad en un documento en la que determinadas palabras marcadas puedan abrir un documento relacionado con ellas.

Para visualizar las primeras webs, se usaba un intérprete de texto.

En 1993, CERN anuncia que la web será libre para todo el mundo.

Los recursos a utilizar (imágenes, vídeos, etc) deben estar acorde con las características de la red usada (velocidad, coste, etc). Se tienen que ajustar a los navegadores del usuario.

Firefox proviene de Netscape.

FUNCIONAMIENTO DE LAS APLICACIONES WEB

Una aplicación web está creada usando como base HTML, necesitando un navegador y conexión a una red.

IMPORTANTE (Desde el 2.2.2)

PROTOCOLO HTTP

Tiene que haber una comunicación entre varios equipos para poder transmitir una web. Para ello, se usan protocolos de comunicación como HTTP.

TCP/IP es el conjunto de distintos protocolos para la conexión.

FRONT-END Y BACK-END

El primero encargado del diseño, el segundo se encarga de los datos recibidos y los procesa.

HERRAMIENTAS PARA CREAR APLICACIONES HTML

* **Navegadores (plugins de visualización en móvil)**
* **Editor de texto/código/HTML/XML/online/visual**
* **Entornos de programación**

VERSIONES HTML

A pesar de que este lenguaje fue creado hace décadas, sigue actualizándose y usándose hoy en día. Filosofía de “estándar vivo”.

VALIDAR EL CÓDIGO

Los navegadores de hoy no son tan estrictos con las normas, es decir, son flexibles en caso de haber fallos intentando ignorarlo. Sin embargo, no podemos pasar esto por alto como desarrolladores web.

IMPORTANTE

* **Documentación externa→** A través de documentos PDF, Word, vídeo, etc explicar las cuestiones más importantes del código y de la web.
* **Documentación interna→** Uso de comentarios dentro del código HTML para ayudar a entenderlo con el uso de caracteres especiales.

**Página web:** Fichero HTML junto con los recursos asociados (fotos, vídeos, sonido, etc).

**Sitio web:** Conjunto de páginas web relacionadas entre sí a través de enlaces.

Tener en cuenta que la codificación aplicada en el archivo sea compatible con el programa (editor) que usemos.

ENLACES

Comprobar que los enlaces funcionan, porque puede ser un enlace roto (Broken Link)

* **Enlaces globales:** Enlaces a recursos externos.
* **Enlaces locales relativos**:Enlaces a recursos internos de la propia página.
* **Enlaces locales absolutos:** Van al directorio raíz del servidor.
* **Enlaces internos**: Navegación dentro del mismo documento. (href = “[archivo\_p\_web]#***id***”)