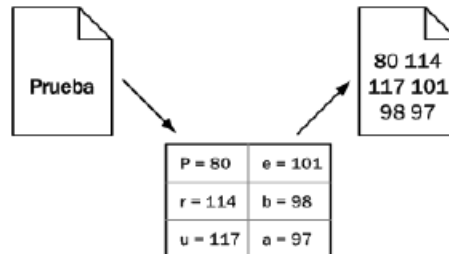


LMSGI – UD02

CARACTERÍSTICAS BÁSICAS

Codificación de caracteres → Tabla de Conversión (ejemplo, ASCII)



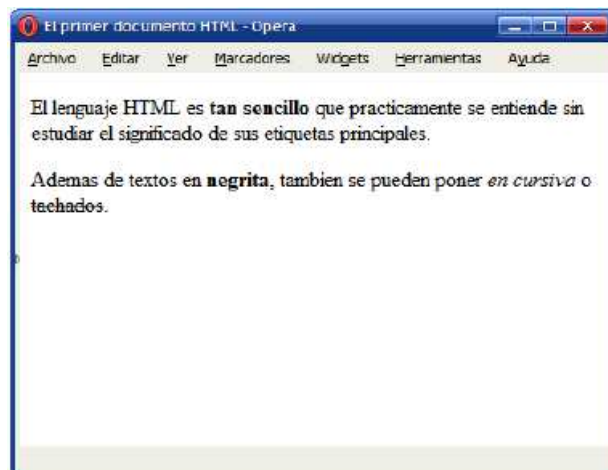
Codificación de contenidos de texto con formato → Utilizar una tabla de conversión similar a las que se utilizan para textos sin formato no es posible, ya que existen infinitos posibles estilos para aplicar al texto.

Soluciones:

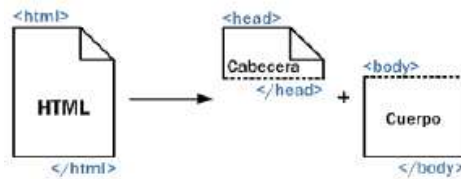
- ☒ Almacenar la información sobre el formato del texto en una zona especial reservada dentro del propio archivo. En esta zona se podría indicar dónde comienza y dónde termina cada formato.
- ☒ El archivo electrónico almacena tanto los contenidos como la información sobre el formato de esos contenidos:

```
<parrafo>  
    Contenido de texto con <importante>algunas  
    palabras</importante> resaltadas de forma especial.  
</parrafo>
```

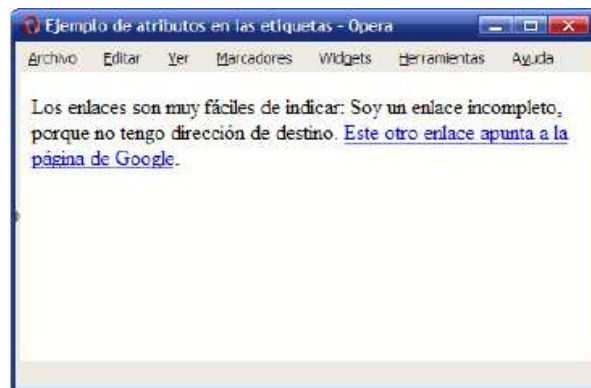
Siglas HTML → *HyperText Markup Language*



Documento HTML → Cabecera (*head*) + Cuerpo (*body*):



91 etiquetas (algunas obsoletas) → Insuficiente para crear páginas complejas
→ uso de atributos:



Tipos de atributos:

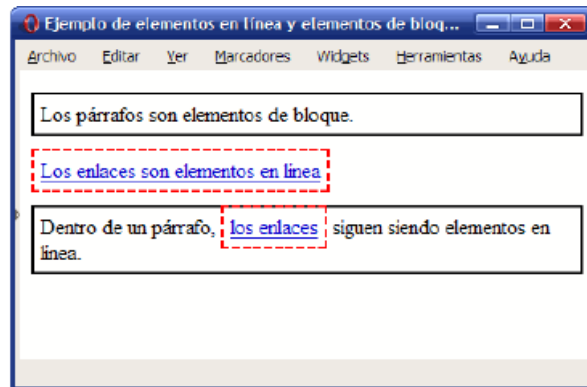
- ⊗ Atributos básicos: se pueden utilizar en casi todas las etiquetas.
Ejemplo: `id = "texto"`.
- ⊗ Atributos para internacionalización: los utilizan las páginas que muestran sus contenidos en varios idiomas o aquellas que quieren indicar de forma explícita el idioma de sus contenidos.
Ejemplo: `lang = "codigo de idioma"`.
- ⊗ Atributos de eventos: sólo se utilizan en las páginas web dinámicas creadas con JavaScript.
Ejemplo: `onclick`.
- ⊗ Atributos para los elementos que pueden obtener el foco: Cuando el usuario selecciona un elemento de la interfaz de una aplicación, se dice que "el elemento tiene el foco del programa".
Ejemplo: `accesskey = "letra"`.

Estructura de un elemento:



HTML clasifica a todos los elementos en dos grupos:

- ☒ Elementos en línea (*inline elements*): sólo ocupan el espacio necesario para mostrar sus contenidos.
- ☒ Elementos de bloque (*block elements*): siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea, aunque sus contenidos no lleguen hasta éste.



Un elemento de bloque no puede aparecer dentro de un elemento en línea. En cambio, un elemento en línea puede aparecer dentro de un elemento de bloque y dentro de otro elemento en línea.

Restricciones básicas que impone XHTML:

- 1) Las etiquetas se tienen que cerrar de acuerdo a como se abren:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con <a>un enlace</a></p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<p>Este es un párrafo con <a>un enlace</p></a>
```

- 2) Los nombres de las etiquetas y atributos siempre se escriben en minúsculas:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con <a href =  
"http://www.google.com"> un enlace </a></p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<P>Este es un párrafo con <A HREF =  
"http://www.google.com"> un enlace </A></P>
```

- 3) El valor de los atributos siempre se encierra con comillas:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con <a href =  
"http://www.google.com"> un enlace </a></p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<p>Este es un párrafo con <a href=http://www.google.com> un  
enlace </a></p>
```

- 4) Los atributos no se pueden comprimir:

Ejemplo correcto en XHTML:

```
<dl compact="compact">...</dl>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<dl compact>...</dl>
```

- 5) Todas las etiquetas deben cerrarse siempre:

Ejemplo correcto en XHTML:

```
<br></br> (también se puede utilizar <br/>)
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<br>
```