



Clases y Objetos en Php.

Definición de un clase

Para definir una clase en PHP se utiliza la palabra reservada **class** seguido del *nombre de clase* y *dos llaves*, entre las cuales se guardará toda la información del esquema.

Esta información se guarda en **propiedades** (las variables dentro de las clases), **métodos** (funciones dentro de las clases) y **constantes**.

Definición de un clase

```
class Persona{  
    private $documento;  
    private $nombre;  
  
    public function __construct(){  
    }  
    public function setDocumento($documento){  
        $this->documento = (int) $documento;  
    }  
    public function setNombre($nombre){  
        if ( ctype_alpha($nombre) ) { $this->nombre = $documento };  
    }  
}
```

Definición de un clase

Visibilidad de las propiedades

Las propiedades deben ser definidas según su visibilidad como: **public**, **private** o **protected**.

- **public**: permite el acceso desde cualquier lugar.
- **protected**: solo desde la clase y sus subclases.
- **private**: solo desde la clase donde se define.

Visibilidad de atributos

```
class Persona{  
    private $documento;  
    private $nombre;  
  
    public function __construct(){  
    }  
    public function setDocumento($documento){  
        $this->documento = (int) $documento;  
    }  
    public function setNombre($nombre){  
        if ( ctype_alpha($nombre) ) { $this->nombre = $documento };  
    }  
}
```

Instanciar un Objeto

Para instanciar un objeto hacemos uso **new** seguido del nombre de la clase.

```
$oPersona=new Persona();
```

Una vez instanciado podemos acceder a sus atributos mediante →, siempre y cuando estos sean públicos.

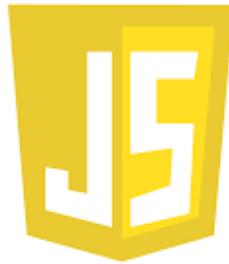
```
$oPersona→apellido
```

Instanciar un Objeto

La referencia a **atributos o métodos dentro de la clase** se realiza mediante la instrucción **\$this**.

```
class Persona{  
    private $mysqli;  
  
    public function __construct(){  
        $this->mysqli = new mysqli('127.0.0.1', 'root', "", 'sistemapersonal');  
    }  
}
```

JavaScript



¿Cómo el navegador representa una página web?

¿Cómo el navegador representa una página web?

Para acceder a un sitio web el navegador solicita su archivo indice (index.html, index.php, etc) que contiene el código HTML de la página.

Este código podrá tener referencias que indicarán al navegador que debe solicitar otros archivos necesarios para el sitio. Por ejemplo cuando escribimos `<link href="/css/styles.css" rel="stylesheet">` el navegador interpreta que debe realizar una nueva petición para descargar el archivo **styles.css**). Ocurrirá lo mismo con la etiqueta `` y `<script>`.

Una vez realizadas todas las peticiones de recursos necesarias comenzará el proceso de renderizado para mostrar en pantalla el sitio web.

¿Cómo el navegador representa una página web?

El proceso de renderizado y presentación se realizará en cinco pasos:

- Construcción de DOM.
- Construcción de CSSOM.
- Construcción del árbol de renderizado.
- Fase de diseño.
- Fase de pintura.

¿Cómo el navegador representa una página web?

Construcción del DOM

El navegador recibe un documento HTML que es básicamente un archivo de texto con un encabezado de respuesta Content-Type = text/html; charset=UTF-8.

Cuando el navegador lee el documento HTML, al encontrar un elemento HTML, crea un objeto JS llamado Node . Finalmente, todos los elementos HTML se convertirán en Node .

Una vez que el navegador ha creado nodos a partir del documento HTML, debe crear una estructura "similar a un árbol" de estos objetos de nodo.

El DOM (Document Object Model) es una API web de alto nivel proporcionada por el navegador para representar una página web y exponerla para que los desarrolladores manipulen sus elementos.

¿Cómo el navegador representa una página web?

Construcción del CSSOM

Después de construir el DOM, el navegador lee todos los CSS y construye un CSSOM (modelo de objetos CSS), una estructura similar a un árbol.

Cada nodo de este árbol contiene información de estilo que se copiará al elemento DOM al que apunta.

Construcción del árbol de renderizado

DOM y CSSOM se combinan para formar un árbol de renderizado que contiene los nodos que deben mostrarse en la página. Desde la raíz del árbol se recorre cada nodo visible y se aplica la regla CSSOM correspondiente.

Finalmente, se genera el árbol de renderizado con los nodos visibles, su contenido y estilo.

¿Cómo el navegador representa una página web?

Fase de Diseño

En esta fase se calcula la posición exacta de los nodos y su tamaño respecto a la ventana gráfica del navegador. De esta forma, se genera un modelo de caja que conoce las posiciones y el tamaño exacto. Este proceso también se conoce como diseño o reflujo .

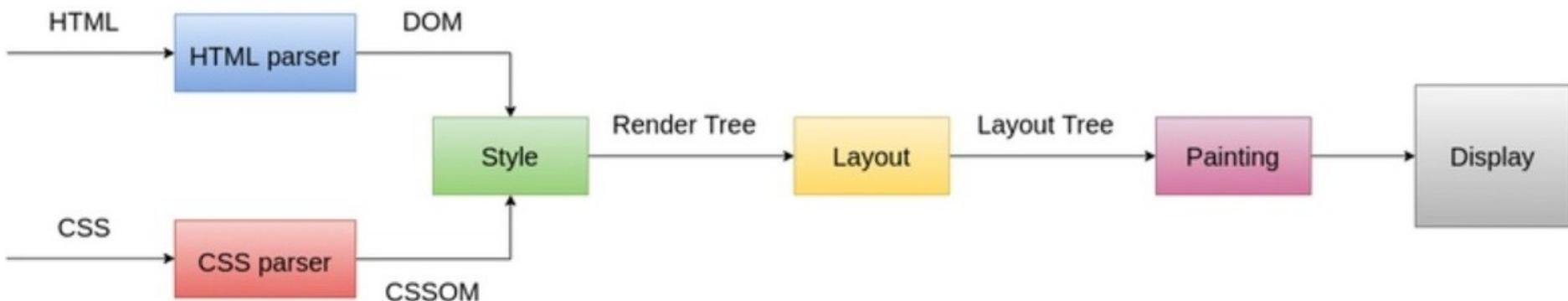
Fase de Pintura

Como conocemos los nodos visibles, su estilo y geometría, toda esta información se utiliza para renderizar los nodos desde el árbol de renderizado a píxeles reales en la pantalla. Este proceso se conoce como "Pintado" y utiliza la capa de interfaz de usuario.

Programación Avanzada

Taller de PHP

¿Cómo el navegador representa una página web?



JavaScript



Algo de JavaScript

Dom – Document Object Model

Es la representación en un conjunto estándar de objetos de la interfaz gráfica de nuestra aplicación.

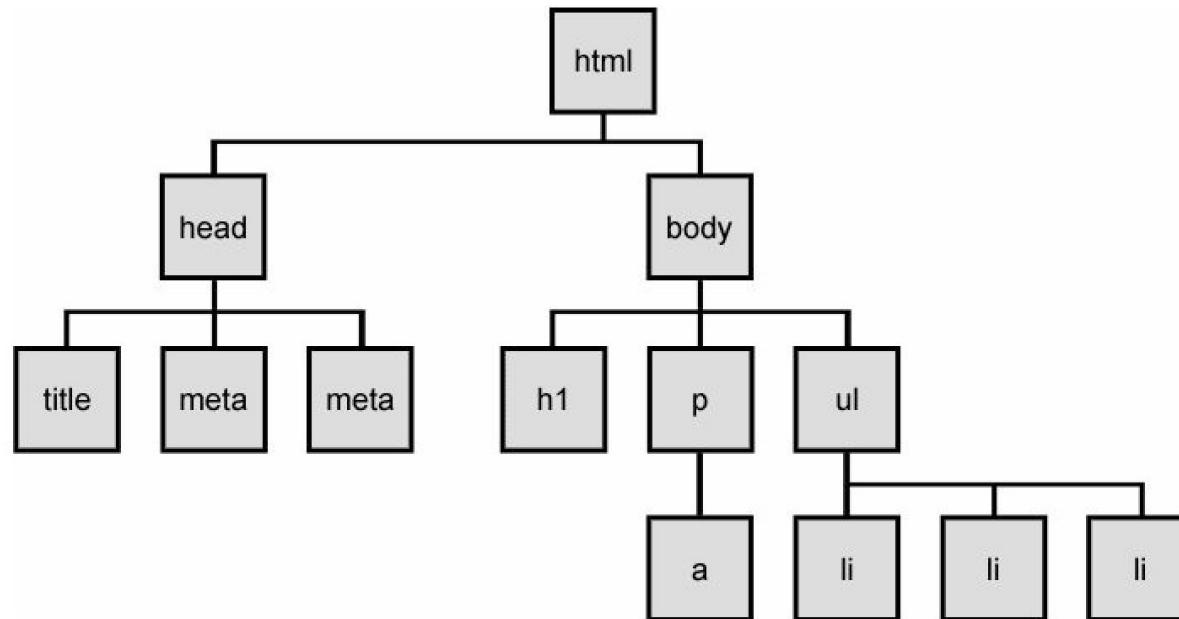
A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML.

Cada vez que el estado de la aplicación cambia, también lo hará su interfaz.

Esto implica actualizar el DOM, lo cual es una tarea costosa en cuanto a rendimiento.

Cuantos más cambios de estado sean necesario reflejar en él, más lento irá nuestra web.

Dom – Document Object Model



JavaScript

Si tenemos el siguiente elemento html:

```
<input type="text" id="nombre" name="nombre" value="Fede">
```

Podemos seleccionarlo desde javascript de la siguiente manera:

```
const nombre = document.getElementById ('nombre');  
console.log(nombre.value)
```

JavaScript

Si tenemos los siguientes elementos html:

```
<input type="text" id="nombre" name="nombre" value="Fede">  
<input type="text" id="correo" name="nombre" value="Fede">  
<input type="text" id="nombre" name="nombre" value="Fede">
```

Podemos seleccionarlos desde javascript de la siguiente manera:

```
const inputs = document.getElementsByTagName('input'); ← colección de objetos HTML  
const arrayDeNodos = Array.from(inputs);           ← array de objetos HTML.  
arrayDeNodos.forEach(cadalInput => { console.log(cadalInput); });
```

JavaScript - Clases

```
class FormPersonas {  
  
    constructor() { //codigo }  
  
    init() { //codigo }  
  
}  
window.addEventListener('DOMContentLoaded', () => {  
    new FormPersonas().init();  
});
```