tWebPage

La clase tWebPage es el contenedor de toda la página, en ella se procesa todo el código necesario para crear los controles que contiene y generar el HTML final que enviaremos al navegador como respuesta a la llamada a nuestro CGI.

Es el único control que que retorna una cadena en el método **Create()**, la cadena contiene todo el código HTML, JS y JQuery necesario para que el navegador visualice la página.

A la estructura básica de HTML;

```
<html>
 <head>
  .
  .
 </head>
 <body>
  .
  .
 </body>
</html>
```

Materialize añade **header**, **main** y **footer**, todas dentro de <body>.

Como Néfele se creo basándonos en Materialize mantiene esta estructura, por lo que tenemos propiedades para incluir contenido en cualquiera de las secciones.

Néfele añade en el <head> **script** para agrupar todos los códigos JS y JQuery que tienen que estar necesariamente en el <head> y **style** para agrupar los stilos CSS que hemos añadido directamente a tWebPage.

También añadimos **script** al final del <body> para los códigos JS y JQuery que no necesitan estar en el <head>.

```
<html>
 <head>
  <script>     /* Códigos JS y JQuery que necesitan cargarse al principio de
la página */
   .
   .
  </script>
  <style>     /* Estilos CSS que hemos añadido a tWebPage */
   .
   .
  </style>
 </head>
 <body>
  <header>
   .
   .
```

```
  </header>
  <main>
   .
   .
  </main>
  <footer>
   .
   .
  </footer>
  <script>   /* Códigos JS y JQuery del resto de la página */
   .
   .
  </script>
 </body>
</html>
```

De esta manera todo el contenido de la página esta dentro de <body> pero estructurado en tres areas para facilitar el tratamiento con las librerías JS de Materialize.

Propiedades

Nombre	Init	Descripción	Versión
cClassId	“wpg”		0.1
Contenido de la Página			
cInHead		Contenido a Añadir en el <Head> de la página	0.1
cCSS		Contenido de Estilo CSS para incluir en el <Head> de la página	0.1
aHeadScript	{}	Array de Script para incluir en el <Head>	0.1
aHeadLinks	{}	Array de Links para incluir en el <Head>	0.1
aHeadMetas	{}	Array de Meta para incluir en el <Head>	0.5
cHeadMeta		Método rápido de añadir un elemento a aHeadMetas	0.5
cInHeader		Contenido para incluir en el <Header> de la página	0.1
cInMain		Contenido para incluir en el <Main> de la página	0.1
cInFooter		Contenido a Añadir en el <Footer>, si contiene algo **NO** se utiliza el Footer estándar de Materialize	0.1
aScript	{}	Array de Script para incluir en la sección <Script> del <Body>	0.1
cScript		Método rapido de añadir un elemento a aScript	0.3
Control de Versiones			
cVersionMaterialize		Versión del MaterializeCSS a utilizar “1.0.0”	0.1
cVersionJquery		Versión del JQuery que vamos a utilizar “3.4.1.min”	0.1
cVersionCoreFiles		Versión a usar de nefelecorecss.css y nefelecorejs.js	0.1
Automatizaciones de la Página			
aOnReady	{}	Código a Ejecutar cuando este lista la página	0.1
cOnReady		Método rapido de añadir un elemento a aOnReady	0.3
cAutoJS		Función del proyecto a llamar en la cabecera para cargar funciones JS	0.1

Nombre	Init	Descripción	Versión
lAutoReloaded	.t.	Obliga a que la página se recargue al ir atrás en el navegador	0.1
nTimeOut	0	Segundos que se puede mantener la página quieta	0.1
cMainPage		Página a la que ir cuando se esta nTimeOut tiempo quieta	0.1
aParamsAuto	{}	Parámetros a incluir en la llamada del TimeOut	0.1
nDelay	0	Tiempo de espera para servir la página	0.1
Aspecto de la Página			
lValign	.f.	Fuerza el centrado Vertical del contenido del <main>	0.1
lContainer	.t.	Se habilita el modo Contenedor o No en la página	0.1
cTitle		Titulo de la página	0.1
cClrText	“black”	Color del texto de la Página	0.1
cClrPane	“white”	Color del fondo de la Página	0.1
cIcon		Favicon de la página	0.1
cLanguage	“es-ES”	Idioma de la página	0.4
lFullScreen	.f.	Muestra la página a pantalla completa	0.1
lNoBackButton	.f.	Anula el botón back del navegador	0.1
cMeasureType	xc_Pixel	Unidad de medida base para los controles	0.1
Parametros del Footer estandar de Materialize			
lFooter	.t.	Incluir Footer estandar de Materialize	0.1
aLinks	{}	Lista de Links en el Footer Estandar, es un array de arays {cURL,cText}	0.1
cTitleFooter		Titulo en el Footer	0.1
cClrFootText	“white”	Color del Titulo del Footer	0.1
cClrFootPane	“pink”	Color del fondo del Footer	0.1
cSubTitle		Subtitulo del Footer	0.1
cClrSubText	“grey”	Color del texto del Subtitulo del Footer	0.1
cCopyright		Texto del Copyright en el Footer	0.1
cClrCrText	“white”	Color del Texto del Copyright	0.1
cClrCrPane		Color del fondo del Copyright	0.1
aBreadCrumbs		Array de path para un BreadCrumbs en el footer	0.3
ShadowSheet - Bloqueo temporal de la página mientras se realizan procesos			
lShadowSheet	.t.	Si queremos activar el Shadowsheet	0.1
cShadowSheetTitle		Titulo a presentar en el ShadowSheet	0.1
cShadowSheetSubTitle		Subtitulo a presentar ne el ShadowSheet	0.1
Banner de información de cookies			
cCookieBannerText		Texto principal que aparece en el banner	0.1
cCookieBannerClose		Texto del botón cerrar	0.1
cCookieBannerMore		Texto del enlace saber más	0.1
cCookieBannerLink		Link al pulsar el texto anterior	0.1
Extras			
lFaIcons	.f.	Carga Font Awesome	0.1
lAlerts	.f.	Carga js de SweetAlert	0.1
lGoogleChart	.F.	Inicializa la página para que pueda contener Google Chart	0.1
lTourShow	.t.	Si se mostrará o no el Tour	0.1
aTour	{}	Array con todos cId de los elementos del tour, si esta vació no se tomara en cuenta lTourShow	0.1

Nombre	Init	Descripción	Versión
cAnalytics		ApiKey de Google Analitics	0.1
cGpsField		Nombre del control que guardará la información GPS	0.1
One Signal			
cOneSignalKey		App key de onesignal	0.1
cOneSignalKeySafari		App key onesignal para navegadores safari	0.1
Ventanas Modal			
cModalMinTop	“4%”	Mínimo Top para ventana modal	0.1
cModalMaxTop	“10%”	Máximo Top para ventana modal	0.1
Progressive Web App			
oManifest		Objeto de la clase wManifest para la configuración del archivo .webmanifest y los metas necesarios para utilizar nuestra web como PWA	0.5
lManifest	.f.	Vamos a utilizar webManifest	0.5

Métodos

Nombre	Descripción	Versión
New()	Instancia la clase para irle añadiendo los controles	0.1
Create()	Monta todo el código necesario para que el navegador pueda visualizar la página y lo retorna como una cadena	0.1
AddHTML(cText)	Nos Añade en el HTML el texto	0.1
ProccessPhp(cPhpFile, aParams, cId)	Añade al código HTML una llamada por Ajax a un modulo PHP, esto nos permite añadir a nuestro proyecto Néfele funcionalidades que ya tengamos operativas en PHP y que aun no hemos pasado a Néfele	0.1
Solo operativas en Modo Debugger		
LogDebug(cText,lComillas)	Nos Envía cText a la Consola del navegador y a Néfele Console, si lComillas == .F. podemos pasarle una instrucción que se ejecuta en la página	0.1
LogError(cText)	Nos Añade un Mensaje de Error en la Consola del Navegador y en Néfele Console	0.1
Assert(cText)	Nos Añade un Comentario al HTML	0.1
AddTimer()	Añade un timer JS a la página	0.3

From:
https://www.nefele.dev/wiki/ - **Nefele Project**

Permanent link:
https://www.nefele.dev/wiki/controles/twebpage

Last update: **04/05/2023 19:26**