
Trabajo Práctico N°2 Integración de Python con HTML y CSS

Resolver los siguientes problemas de manera incremental utilizando Python, HTML y CSS. La consigna es armar uno o más scripts de python que permitan generar de manera dinámica una web de artículos periodísticos.

- 1) Crear la clase base `ParserHtml`, inicializarla con una lista de tuplas de artículos y generar un HTML básico con f-strings. Los artículos deben tener nombre, autor y texto.
- 2) Filtrar artículos vacíos y normalizar el texto del autor quitando espacios y capitalizando el nombre).
- 3) Modificar el HTML generado para que tenga un `header` con título del sitio y un `footer` con la fecha de generación.
- 4) Incluir un bloque de `<style>` dentro del `<head>` para aplicar estilos simples como colores, márgenes y tipografía básica.
- 5) Separar los artículos por autor usando encabezados `<h3>`, generando una sección por autor con sus artículos debajo.
- 6) Incluir al comienzo de la página un índice con enlaces a cada autor usando anchors internos (`...`).
- 7) Reemplazar las tuplas por objetos de una clase `Articulo` con atributos `titulo`, `autor`, `texto` y un método `to_html()` que devuelva su representación.
- 8) Agregar un método para filtrar por una palabra clave que devuelva solo los artículos que contienen esa palabra en el texto.
- 9) Modificar la representación del artículo para mostrar solo los primeros 300 caracteres de `texto`, seguido de
- 10) Separar cada artículo en su propia página `.html` y desde el índice principal linkear a cada una.
- 11) Incluir un enlace para volver al índice en cada página de artículo individual.
- 12) Incorporar Bootstrap en todas las páginas generadas usando un link en el `<head>`. Incorporar el sistema de grillas agrupando los artículos de a 3 por fila y modificar la navbar agregando las clases de bootstrap.
- 13) Organizar el proyecto en múltiples archivos: uno para la lógica y otro para la ejecución del programa.
- 14) Centralizar la generación del pie de página en una función reusable, incluyendo el año actual.
- 15) Agregar al comienzo del índice una tabla con la cantidad de artículos publicados por cada autor.
- 16) Validar que el título y el texto tengan al menos 10 caracteres; si no, lanzar una excepción personalizada.

-
- 17) Crear una lista de letras de la A a la Z que filtre los autores por inicial del apellido.
 - 18) Crear un archivo que use `assert` para probar métodos clave tanto de la clase `Articulo` como `ParserHtml`.
 - 19) Agregar al final de cada artículo enlaces para navegar al anterior o siguiente artículo (si existen).
 - 20) Preparar el proyecto para su entrega. Como mínimo, generar un `.zip` con todos los archivos, usando un script de python. Opcionalmente, subirlo a un repositorio en GitHub usando Git.