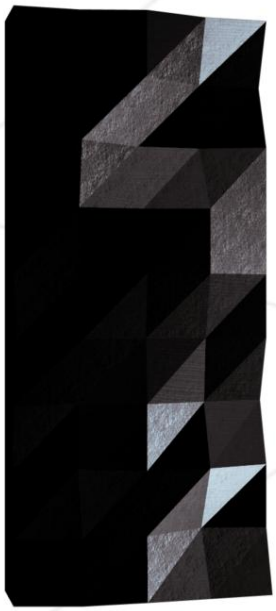
Machine Translated by Google



Entrenamiento de Python-Django - 0

Fuera de banda

Resumen: ¡Hoy vas a conquistar Silicon Valley con tus habilidades recién adquiridas en POO con Python!

Versión: 1.1

Machine Translated by Google

Contenido

Machine Translated by Google

Capítulo I

Preámbulo

Aquí está la letra de la "Canción del Software Libre":

Únase a nosotros ahora y comparta el software;
serán libres, hackers, serán libres.
Únase a nosotros ahora y comparta el software;
serán libres, hackers, serán libres.

Los acaparadores pueden conseguir montones
de dinero, eso es cierto, hackers, eso es cierto.
Pero no pueden ayudar a sus vecinos; eso no está
bien, hackers, eso no está bien.

Cuando tengamos suficiente software libre, a
nuestra llamada, hackers, a nuestra llamada,
eliminaremos esas licencias sucias, cada vez
más, hackers, cada vez más.

Únase a nosotros ahora y comparta el software;
serán libres, hackers, serán libres.
Únase a nosotros ahora y comparta el software;
serán libres, hackers, serán libres.

Machine Translated by Google

Capítulo II

Reglas generales

- Su proyecto debe realizarse en una máquina virtual.
- Su máquina virtual debe tener todo el software necesario para completar su proyecto. Estos software deben configurarse e instalarse.
- Puede elegir el sistema operativo que utilizará para su máquina virtual.
- Debe poder utilizar su máquina virtual desde una computadora del clúster.
- Debe utilizar una carpeta compartida entre su máquina virtual y su máquina host.
- Durante tus evaluaciones utilizarás esta carpeta para compartir con tu repositorio.

Sus funciones no deben cerrarse inesperadamente (fallo de segmentación, error de bus, doble liberación, etc.), salvo por comportamientos indefinidos. Si esto ocurre, su proyecto se considerará no funcional y recibirá un 0 en la evaluación.

Te animamos a crear programas de prueba para tu proyecto, aunque este trabajo no tenga que entregarse ni se califique. Esto te permitirá evaluar fácilmente tu trabajo y el de tus compañeros. Estas pruebas te resultarán especialmente útiles durante la defensa. De hecho, durante la defensa, puedes usar tus propias pruebas o las del compañero que estás evaluando.

Envía tu trabajo al repositorio Git asignado. Solo se calificará el trabajo en el repositorio Git. Si Deepthought se encarga de calificar tu trabajo, lo hará después de la evaluación por pares. Si se produce un error en alguna sección de tu trabajo durante la calificación de Deepthought, la evaluación se detendrá.

Machine Translated by Google

Capítulo III

Reglas específicas de hoy

- No hay código en el ámbito global. ¡Queremos funciones!
- A menos que se especifique lo contrario, cada archivo escrito en Python deberá terminar con un bloque

```
si __nombre__ == '__principal__':
    # Sus pruebas y su manejo de errores
```

- Cualquier excepción no detectada anulará el trabajo, incluso en el caso de un error que usted haya cometido.
  requerido para realizar la prueba.
- Queda prohibida cualquier importación, salvo las especificadas en el apartado 'Funciones autorizadas'
  sección del encabezado de cada ejercicio.
- Tendrás que utilizar el intérprete de Python3.

Machine Translated by Google

Capítulo IV

Ejercicio 00

	Ejercicio 00
Ejercicio 00: ¡Conquistando Silicon Valley!	
Directorio de entrega: ex00/	
Archivos a entregar: render.py, myCV.template, settings.py Funciones permitidas:	
import sys, os, re	

Acabas de completar tu increíble curso de desarrollo y nuevas perspectivas están a punto de cambiar tu vida para siempre. Llegas a Silicon Valley con un solo objetivo: desarrollar tu revolucionario generador de currículums con tecnología innovadora y convertirte en el Bill Gates de la industria de la búsqueda de empleo.

Ahora todo lo que queda por hacer es... crear tu tecnología.

Cree un programa render.py que tome un archivo con la extensión .template como parámetro. Este programa deberá leer el contenido del archivo, reemplazar algunos patrones con valores definidos en un archivo settings.py (no será necesario un bloque `if __name__ == '__main__':` para este archivo) y escribir el resultado en un archivo con la extensión .html.

Deberás poder replicar el siguiente ejemplo con tu programa .

```
$> cat settings.py name =
"duoquadragintian" $> cat file.template
<p>"-¿Quién eres tú?

-¡Un {nombre}!"</p>
$> python3 render.py file.template $> cat file.html
<p>"-¿Quién eres?

-¡Un duoquadragintian!</p>
```

Será necesario gestionar los errores, incluida una extensión de archivo incorrecta, un archivo inexistente o un número incorrecto de argumentos.

Machine Translated by Google

Tendrás que entregar un archivo myCV.template que, una vez convertido a HTML, deberá incluir al menos la estructura completa de un (doctype, cabecera y cuerpo de página, título de la página, nombre y apellidos del titular del currículum, su edad y profesión. Por supuesto, estos datos no aparecerán directamente en el archivo .template.

ayuda(globales), expansión de palabras clave...

Machine Translated by Google

Capítulo V

Ejercicio 01

	Ejercicio 01
Ejercicio 01; Startup innovadora busca pasantía. Se requieren 10 años de exp.	
Directorio de entrega: ex01/	
Archivos a entregar: intern.py	
Funciones permitidas: Ninguna	

No puedes emprender este viaje solo. Eliges contratar a alguien para que te prepare café.
Te puedo ayudar, seria mejor un pasante (son mas baratos).

Crea la clase Intern que contiene las siguientes funcionalidades:

Un constructor que toma una cadena de caracteres como parámetro y asigna su valor a un atributo de nombre. "¿Mi nombre? No soy nadie, soy becario, no tengo nombre." se implementará como valor predeterminado.

Un método __str__() que devolverá el atributo Nombre de la instancia.

Una clase Coffee con un método __str__() simple que devolverá la cadena de caracteres "Este es el peor café que jamás has probado".

Un método work() que generará solo una excepción (use el tipo básico (Exception)) con el texto "Solo soy un pasante, no puedo hacer eso...".

Un método make_coffee() que devolverá una instancia de la clase Coffee que habrá implementado en la clase Intern.

En tus pruebas, tendrás que instanciar la clase Intern dos veces: una sin nombre y la otra con el nombre "Mark".

Muestra el nombre de cada instancia. Pídele a Mark que te prepare un café y muestra el resultado. Pide al otro becario que trabaje. Tendrás que gestionar la excepción en tu...
prueba.

Machine Translated by Google

Capítulo VI

Ejercicio 02

	Ejercicio 02
Ejercicio 02: 5 clases 1 taza.	
Directorio de entrega: ex02/	
Archivos a entregar: Beverages.py	
Funciones permitidas: Ninguna	

El café es bueno. ¡Es mejor elegir tu bebida! Crea una clase HotBeverage con las siguientes funcionalidades:

Un atributo de precio con un valor de 0,30.

Atributo Aname con el valor “bebida caliente”.

Un método description() que devuelve una descripción de instancia. El valor de la descripción Será "Solo un poco de agua caliente en una taza".

Un método __str__() que devuelve una descripción de instancia en este formato:

```
nombre: <atributo de nombre>
precio: <atributo de precio limitado a dos decimales> descripción: <descripción
de la instancia>
```

Por ejemplo, una instancia de HotBeverage se vería así:

```
Nombre: bebida caliente
Precio: 0,30
Descripción: Solo un poco de agua caliente en una taza.
```

Luego crea las siguientes clases derivadas HotBeverage:

Café :

nombre: "café"

precio: 0,40

Descripción: "Un café, para mantenerse despierto."

Machine Translated by Google

Té :

nombre: "té"

precio: 0.30

Descripción: "Solo un poco de agua caliente en una taza".

Chocolate :

nombre: "chocolate"

precio: 0,50

Descripción: "Chocolate, dulce chocolate..."

Capuchino:

nombre: "capuchino"

precio: 0,45

descripción: “¡Un po' di Italia nella sua tazza!”

Debes redefinir SÓLO lo que es necesario, lo que necesitas cambiar, redefinir... (cf. DRY).

En sus pruebas, instancia cada clase entre: Bebida Caliente, Café, Té, Chocolate y Capuchino y exhibirlos.

Machine Translated by Google

Capítulo VII

Ejercicio 03

	Ejercicio 03
Ejercicio 03: ¡Gloriosa máquina de café!	
Directorio de entrega: ex03/	
Archivos a entregar: machine.py, Beverages.py Funciones	
permitidas: importar aleatorio	

¡Ya está! ¡Tu empresa está en marcha! Tu primera recaudación de fondos te ofreció un local. Tienes un becario para el café y una planta verde de nivel 10 en la entrada del edificio para mantener todo en orden.

Sin embargo, no todo es perfecto: tu becario prepara un café horrible y la mitad del salario mínimo es caro para este lodo. ¡Ha llegado el momento de invertir en equipo nuevo que te lleve al éxito personal!

Crea la clase CoffeeMachine que contiene:

• Un constructor.

• Una clase EmptyCup que hereda HotBeverage, con el nombre "taza vacía", un precio de 0,90 y la descripción "¡¿Una taza vacía?! ¡Devuélveme mi dinero!".

Copie el archivo Beverages.py del ejercicio anterior en esta carpeta de ejercicios para utilizar las clases que contiene.

• Una clase BrokenMachineException que hereda la excepción con el texto "Esta máquina de café debe ser reparada". Este texto debe estar definido en el generador de la excepción.

• Un método repair() que repara la máquina para que pueda volver a servir bebidas calientes.

• Un método serve() que tendrá las siguientes especificaciones:

Parámetros: un parámetro único (distinto de sí mismo) que será una clase derivada de HotBeverage.

Machine Translated by Google

Retorno: Alternativamente (aleatoriamente), el método devuelve una instancia del conjunto de clases en parámetro y, alternativamente, una instancia de EmptyCup.

Obsolescencia: La máquina es barata y se estropea después de servir 10 bebidas.

Cuando está fuera de servicio: la llamada al método serve() debe generar un CoffeeMachine.BrokenMach Escriba una excepción hasta que se llame al método repair().

Solución: después de llamar al método repair(), el método serve() puede volver a funcionar sin generar la excepción antes de que vuelva a fallar después de servir 10 bebidas.

En tus pruebas, instancia la clase CoffeeMachine. Solicita varias bebidas del archivo Beverages.py y muestra la bebida que te sirven hasta que la máquina se averíe (luego gestionarás la excepción generada). Repara la máquina y reinicia hasta que se averíe de nuevo (gestiona la excepción de nuevo).

Machine Translated by Google

Capítulo VIII

Ejercicio 04

	Ejercicio 04
Ejercicio 04: Una clase básica ft. RMS.	
Directorio de entrega: ex04/	
Archivos a entregar: elem.py	
Funciones permitidas: Ninguna	

Ahora es momento de trabajar en tu visibilidad web. Te gustaría usar tus recién adquiridos conocimientos de Python para diseñar eficientemente tu contenido HTML, pero te gustaría recibir consejos de alguien superior para aprender a hacerlo. Decides sacrificar a tu becario por los dioses de la programación.

Ahora que tienes una máquina de café, se ha vuelto bastante inútil... Simplemente quemalo...

San Ignacio se presenta ante vosotros para deciros una preciosa revelación:

"Los elementos HTML comparten casi la misma estructura (etiqueta, contenido, atributos). Sería prudente crear una clase que pueda ensamblar todos esos comportamientos y especificaciones compartidos para usar la fuerza heredada en Python para derivar esta clase de manera simple y fácil sin tener que reescribir todo.

Solo entonces San Ignacio ve la Mac en la que estás trabajando. Asustado, huye sin más, dejando atrás solo un archivo de pruebas y una clase incompleta. A toda prisa, completas la clase Elem (los huecos que faltan se indican con [...]) con las siguientes especificaciones:

Machine Translated by Google

- Un constructor que toma como parámetro el nombre del elemento, los atributos HTML y el tipo (simple o etiquetas dobles).
- Un método __str__() que devuelve el código HTML del elemento.
- Un método add_content() que permite agregar los elementos al final del contenido.
- Una subclase de excepción dentro de ella.

Si el trabajo está bien hecho, podrás representar cualquier elemento HTML y su contenido con tu clase Elem. Ahora, la recta final:

- El archivo tests.py proporcionado en el tarbal en el apéndice del tema debe funcionar correctamente (sin error de afirmación, la salida de la prueba debe indicar explícitamente su éxito).
  Claro, no somos tan crueles como para probar funcionalidades que no se requieren explícitamente en este ejercicio. Jajaja... No lo somos... en serio.
- También debes replicar y mostrar la siguiente estructura con la ayuda de tu Clase Elem:

```
<html>
    <cabeza>
        <título>
            "¡Hola tierra!" </title> </
        head> <body>
    <h1>


            "¡Oh, no, otra vez no!"
```

Machine Translated by Google

Capítulo IX

Ejercicio 05

	Ejercicio 05
Ejercicio 05: ¡Crea tus propios elementos!	
Directorio de entrega: ex05/	
Archivos a entregar: elem.py, elements.py Funciones	
permitidas: Ninguna	

¡Felicitaciones! Ahora puedes generar cualquier elemento HTML y su contenido. Sin embargo, generar cada elemento especificando cada atributo con cada instanciación resulta un poco tedioso. Aquí tienes la oportunidad de usar el legado para crear clases pequeñas que sean más fáciles de usar. Crea las siguientes clases derivadas de la clase Elem creada en el ejercicio anterior:

- html, cabeza, cuerpo
- título
- meta
- imagen
- mesa, th, tr, td
- ul, ol, li
- h1
- h2
- pag
- división
- durar
- hora
- br

Machine Translated by Google

El constructor de cada clase podrá tomar el contenido como primer argumento y:

```
imprimir( Html( [Cabeza(), Cuerpo()] ) )
```

se mostrará:

```
<html>
   <cabeza></cabeza>
   <cuerpo></cuerpo>
</html>
```

Sé inteligente y reutiliza las funcionalidades que has codificado en la clase Elem. Debes usar la herencia.

Demuestre el funcionamiento de estas clases con varias pruebas (a su elección) que cubrirán todas sus funcionalidades. Tras codificarlas, no necesitará especificar el nombre ni el tipo de etiqueta, lo cual resulta muy práctico. Nunca más tendrá que instanciar directamente la clase Elem. De hecho, a partir de ahora, esto está prohibido.

Para comprender los beneficios de las clases derivadas de d'Elem en comparación con el uso directo de d'Elem, tomemos la estructura del documento HTML del ejercicio anterior.
Debes replicarlo usando tus nuevas clases.

```
<html>
   <cabeza>
      <título>
         "¡Hola tierra!" </title> </
      head> <body>
   <h1> "¡Oh,
   no, otra
      vez
         no!" </h1> <img src="http://

      i.imgur.com/pfp3T.jpg" /> </body> </html>
```

Mucho más sencillo, ¿no? :)

Machine Translated by Google

Capítulo X

Ejercicio 06

	Ejercicio 06
Ejercicio 06: Validación.	
Directorio de entrega: ex06/	
Archivos a entregar: Page.py, elem.py, elements.py Funciones	
permitidas: Ninguna	

Aunque has progresado muchísimo, tu trabajo aún necesita un poco de limpieza. Un poco más de adaptación. Eres así: te encantan las restricciones y los desafíos. ¿Por qué no imponer una norma a la estructura de tus documentos HTML? Empieza a copiar las clases de los dos ejercicios anteriores en la carpeta de este ejercicio.

Cree una clase de página cuyo constructor acepte como parámetro una instancia de una clase que herede Elem. Su clase de página debe implementar un método is_valid() que debe ser "Verdadero" si se cumplen todas las siguientes reglas y, en caso contrario, "Falso".

- Si, en la ruta del árbol, un nodo no tiene uno de los siguientes tipos: html, head, body, title, meta, img, table, th, tr, td, ul, ol, li, h1, h2, p, div, span, hr, br o Text, el árbol no es válido.

- HTML debe contener estrictamente un encabezado y luego un cuerpo.
- El encabezado solo debe contener un Título y sólo un Título.
- Body y Div solo deben contener los siguientes tipos de elementos: H1, H2, Div, Table, Ul, Ol, Span o Texto.
- Título, H1, H2, Li, Th, Td solo deben contener un Texto y solo este Texto.
- P sólo debe contener texto.
- El espacio debe contener únicamente texto o alguna P.
- Ul y Ol] deben contener al menos un Li y sólo algo de Li.

Machine Translated by Google

- Tr debe contener al menos un Th o Td y solo algunos Th o Td. El Th y el Td deben ser mutuamente excluyentes.
- Tabla: sólo debe contener Tr y sólo algunos Tr.

Su clase de página también debe poder:

- Mostrar su código HTML al imprimir una instancia. Atención: el código HTML mostrado debe ir precedido de un doctype si solo el tipo del elemento raíz es HTML.
- Escriba su código HTML en un archivo mediante un método write_to_file que toma el nombre del archivo como parámetro. Atención: El código HTML escrito en el archivo debe ir precedido de un doctype si el tipo del elemento raíz es HTML.

Demuestre cómo funciona su clase Page con una serie de pruebas: usted elegirá...
que cubrirá todas las funcionalidades.

Machine Translated by Google

Capítulo XI

Presentación y evaluación por pares

Entrega tu tarea en tu repositorio de Git como de costumbre. Solo se evaluará el trabajo dentro de tus repositorios durante la defensa. No dudes en verificar los nombres de tus carpetas y archivos para asegurarte de que sean correctos.

El proceso de evaluación se realizará en el computador del evaluado.
grupo.