

Universidad de La Laguna. Escuela Técnica Superior de Ingeniería Informática
Tercero del Grado de Informática

PROCESADORES DE LENGUAJES. REPASO 2ª PARTE

2015/2016 2 páginas

Nombre: _____ Fecha 2015/2016 _____

1.
 - a) Escriba un conversor de temperaturas usando objetos y herencia.
 - b) Defina una clase `Medida` cuyos objetos tienen un valor numérico (p. ej. `4.5e2`) y un tipo dado por una cadena (p. ej. `"Km"` por Kilómetros o `"m3"` por metros cúbicos o `"F"` por grados Fahrenheit).
 - c) La clase `Temperatura` hereda de `Medida`.
 - d) Las clases `Celsius`, `Fahrenheit` heredan de `Temperatura`
 - e) Si `num` está en Celsius, la expresión `(num * 9/5)+32` nos da los grados Fahrenheit; si `num` está en Fahrenheit, la expresión `(num - 32)*5/9` nos da los grados Celsius
 - f) En la entrada se podrán escribir entradas como `-1.43e2f to C` para convertir de Fahrenheit a Celsius.
2. ¿Que argumentos espera el método `template` de Underscore? ¿Cual es la función de cada uno de ellos?
3. ¿Que diferencia hay entre `<% ... %>`, `<%= ... %>` y `<%- ... %>`?
4. Escriba una expresión JavaScript que permita reemplazar todas las apariciones de palabras consecutivas repetidas (como `hello hello`) por una sólo aparición de la misma
5. Escriba una expresión regular que reconozca las cadenas delimitadas entre dobles comillas en las que se permiten comillas escapadas en su interior
6. ¿Como se llama el método que permite obtener una representación como cadena de un objeto? ¿Que parámetros espera? ¿Como afectan dichos parámetros?
7. Defina el término *code smelling*
8. ¿Que diferencia hay entre un *code smell* y un *bug*?
9. Explique el code smell *Duplicated Code*
10. Explique el Switch smell
11. ¿Como creamos el directorio con el esqueleto inicial para las pruebas con mocha?
12. Rellene lo que falta:

```
var assert = chai._____;

suite('temperature', function() {
  test('[1,{a:2}] == [1,2]', function() {
    assert._____( [1, {a:2}], [1, {a:2}]);
  });
  test('5X = error', function() {
    original.value = "5X";
    calculate();
    assert._____(converted.innerHTML, /ERROR/);
  });
});
```

13.