Universidad de La Laguna. Escuela Técnica Superior de Ingeniería Informática
Tercero del Grado de Informática
PROCESADORES DE LENGUAJES. EXAMEN 2ª PARTE
08/04/2015 8 páginas

Nombre: _____ Fecha 08/04/2015_____

1. jQuery uses CSS selectors to select elements? True or false?
2. Which sign does jQuery use as a shortcut for jQuery?
3. With jQuery, look at the following selector: `$("div")`. What does it select?
4. The jQuery `html()` method works for both HTML and XML documents. True or false?
5. What is the correct jQuery code to set the background color of all p elements to red?
6. With jQuery, look at the following selector: `$("div.intro")`. What does it select?
7. Which jQuery method is used to hide selected elements?
8. Which jQuery method is used to set one or more style properties for selected elements?
9. Which jQuery method is used to perform an asynchronous HTTP request?
10. What is the correct jQuery code for making all div elements 100 pixels high?
11. Which jQuery function is used to prevent code from running, before the document is finished loading?
12. Which jQuery method should be used to deal with name conflicts?
13. Which jQuery method is used to switch between adding/removing one or more classes (for CSS) from selected elements?
14. Look at the following jQuery selector: `$("div#intro .head")`. What does it select?
15. Is jQuery a W3C standard?
16. What method of the `req` object returns a boolean value that is `true` if the request's `X-Requested-With` header field is `XMLHttpRequest`, indicating the intent to retrieve data from the URL without having to do a full page refresh?.

    Fill the gap:

    ```
    app.get('/chuchu', function (req, res) {
      var isAjaxRequest = req._____________;
      ...
    }
    ```

17. La siguiente vista incluye el código JavaScript de un request Ajax:

    ```
    $ cat views/index.ejs
    <!doctype html>
    <html>
      <head>
        <title><%- title %></title>
      </head>
    ```

```
  <body>
    <h1><%- title  %></h1>
    <ul>
      <li><a href="http://jquery.com/" id="jq">jQuery</a>
      <li><a href="/chuchu">Visit chuchu!</a>
    </ul>
    <div class="result"></div>
    <script src="https://code.jquery.com/jquery-2.1.3.js"></script>
    <script>
      $( document ).ready(function() {
          $( "#jq" ).click(function( event ) {
              event.preventDefault();
              $.get( "/chuchu", {nombres: ["juan", "pedro"]}, function( data ) {
                $( ".result" ).html( data["answer"]);
                console.log(data);
              }, 'json');
          });
      });
    </script>
  </body>
</html>
```

Supongamos que el servidor express.js responde al request con:

```
res.send('{"answer": "Server responds: hello world!"}')
```

18. ¿Como podemos acceder en el servidor express.js al objeto `{nombres: ["juan", "pedro"]` enviado desde el cliente como segundo argumento de `$.get`?

19. ¿Cuando se ejecuta la callback asociada con la llamada a `$.get`?

20. ¿Que contendrá el parámetro `data` pasado a la callback?

21. ¿Que hace el cuarto parámetro de `$.get`?

22. ¿Que argumentos espera el método `template` de Underscore? ¿Cual es la función de cada uno de ellos?

23. ¿Que diferencia hay entre `<% ... %>`, `<%= ... %>` y `<%- ... %>`?

24. El atributo `templateSettings` de Underscore puede ser usado para configurar los templates. Rellene las expresiones regulares que faltan para que se usen delimitadores con llaves como `{{ ... }}` para evaluar, `{{= ... }}` para interpolar y `{{- ... }}` para interpolar y escapar el HTML:

```
_.templateSettings = {
       interpolate: /____________________/gim,
       evaluate:    /____________________/gim
       escape:      /____________________/gim
 }
```

25. Queremos mostrar una lista de `items` en una tabla con dos columnas. En la primera columna va el número de orden y en la segunda el nombre del item. El template Underscore se carga desde el elemento `#usageList` del DOM y el resultado del template se vuelca en el elemento `#target` como sigue:

```
var items = [ {name:"Alejandro"}, {name:"......."}, {name:"Zacarias"} ];
var template = usageList.innerHTML;
target.innerHTML = _.template(template,{items:items});
```

Complete el identificador del `<script>`, el identificador del `<div>` usado para la salida y la parte que falta entre las etiquetas `<tbody>`:

```
<script type="text/html" id='_________'>
    <table>
      <thead>
         <tr>
            <th> Id </th> <th>Name</th>
         </tr>
      </thead>
      <tbody>









    </tbody >
   </table>
</script>
<!-- Create your target -->
<div id="______"></div>
```

26. ¿Que diferencias y que similitudes hay entre los cookies y localStorage?

27. ¿Que diferencias hay entre localStorage y sessionStorage?

28. ¿En que consiste las reglas de segregación conocidas como *same origin policy*? ¿Como se relacionan con localStorage?

29. ¿Como guardo en localStorage el contenido de una variable `chuchu`? ¿Como leo el contenido almacenado en localStorage con clave `chuchu`?

30. ¿Como borro una clave almacenada en localStorage?

31. Una vez instalado el Heroku toolbelt nos debemos autenticar en heroku con el cliente. ¿Cual es el comando para autenticarnos?

32. ¿Cual es el comando para crear nuestra aplicación en Heroku (suponemos que ya esta bajo el control de `git`? ¿Qué remoto tendremos definido después de crear nuestra aplicación en Heroku?

33. ¿Cual es el comando para desplegar nuestra aplicación en Heroku?

34. Si la versión que queremos publicar en heroku no está en la rama `master` sino que está en la rama `tutu` ¿Como debemos modificar el comando anterior?

35. ¿Con que comando puedo abrir una ventana en el navegador que visite la aplicación desplegada? ¿Que formato tiene la URL para nuestra aplicación?

36. ¿Con que comando puedo ver los logs de la aplicación desplegada?

37. ¿Como se debe llamar el fichero en el que explicito que comando debe usarse para arrancar nuestra aplicación en Heroku?

38. Heroku se percata que nuestra aplicación es una aplicación desarrollada con `Node.js` por la presencia de un cierto fichero. ¿De que fichero estamos hablando?

39. ¿Cual es la mejor forma de ejecutar en local una aplicación express.js que va a ser desplegada en Heroku?

40. Conteste:

    a) Explique que hace el siguiente fragmento de código:

```
> RegExp.prototype.bexec = function(str) {
      var i = this.lastIndex;
      var m = this.exec(str);
      if (m && m.index == i) return m;
      return null;
    }
[Function]
```

    b) Rellene las salidas que faltan:

```
> re = /d(b+)(d)/ig
/d(b+)(d)/gi
> z = "dBdXXXXDBBD"
'dBdXXXXDBBD'
> re.lastIndex = 3
> re.bexec(z)
-------------------------------------------------------
> re.lastIndex = 7
> re.bexec(z)
-------------------------------------------------------
```

41. Estos son los tokens usados en el analizador para el subconjunto de JavaScript. Rellene las expresiones regulares que faltan:

```
var WHITES              = /___/g;
var ID                  = /____________/g;
var NUM                 = /\b\d+(\.\d*)?([eE][+-]?\d+)?\b/g;
var STRING              = /('___________'|"___________")/g;
var ONELINECOMMENT      = /______/g;
var MULTIPLELINECOMMENT = /__________________/g;
var TWOCHAROPERATORS    = /(===|!==|[+][+]|--|==|[<>=+-/*%]=]|&&|[|][|])/g;
var ONECHAROPERATORS    = /([*\/=()&|;:,{}[\]])/g;
var tokens = [WHITES, ID, NUM, STRING, ONELINECOMMENT,·
              MULTIPLELINECOMMENT, TWOCHAROPERATORS, ONECHAROPERATORS ];
```

42. Defina matemáticamente el cierre de Kleene de un conjunto A

43. Defina matemáticamente el concepto de gramática

44. Defina matemáticamente el concepto de derivación

45. Defina matemáticamente el concepto de lenguaje generado por una gramática

46. Defina el concepto de árbolde análisis sintáctico de una frase. ¿Que relación existe entre derivaciones y árboles de análisis sintáctico?

47. Encuentre una gramática equivalente a esta:

```
A: A 'a' | 'b'
```

pero que no sea recursiva por la izquierda.

48. Escriba un analizador sintactico descendente recursivo para la gramática resultante del ejercicio anterior

49. Dada la gramática:

```
A: 'b' 'c' | 'b' 'd'
```

encuentre una equivalente en la que se hayan eliminado los factores comunes

50. Escriba un analizador sintactico descendente recursivo para la gramática resultante del ejercicio anterior

51. Recuerde el **analizador sintáctico descendente predictivo recursivo** para la gramática:

- $\Sigma = \{;, =, ID, P, ADDOP, MULOP, COMPARISON, (,), NUM\}$
- $V = \{statements, statement, condition, expression, term, factor\}$
- Productions:
  a) statements → statement ';' statements | statement
  b) statement → ID '=' expression | P expression | IF condition THEN statement·
  c) condition → expression COMPARISON expression
  d) expression → expression ADDOP term | term
  e) term → term MULOP factor | factor
  f) factor → '(' expression ')' | ID | NUM
- Start symbol: *statements*

Rellene las partes que faltan de código CoffeeScript del método que se encarga de reconocer el lenguaje generado por `expression`. En este código se ha eliminado la recursión por la izquierda y factorizado por la izquierda:

```
expression = ->
  result = term()
  while _________ and _______________ is "ADDOP"
    type = _________·_____
    match "_____"
    right = ______
    result =
      type: ____
      left: result
      right: right
  result
```

Explique como se han modificado las reglas de producción de expression

expression → expression ADDOP term | term

para dar lugar a este código.

52. Rellene las partes que faltan de código CoffeeScript del método que se encarga de reconocer el lenguaje generado por `statement` para la gramática definida anteriormente:

```
statement = ->
  result = null
  if _________ and _________.____ is "ID"
    left =
      type: "ID"
      value: _________._____

    _____ ____
    _____ ___
    right = __________()
    result =
      type: "="
      left: left
      right: right
  else if lookahead and lookahead.type is "P"
    match "P"
    right = ____________
    result =
      type: "P"
      value: right
  else if lookahead and lookahead.type is "IF"
    match ____
    left = ___________
    match "THEN"
    right = ___________
    result =
      type: "IF"
      left: left
      right: right
  else # Error!
    throw "Syntax Error. Expected identifier but found " +
      (if lookahead then lookahead.value else "end of input") +
      " near '#{input.substr(lookahead.from)}'"
  result
```

53. Rellene las partes que faltan del código CoffeeScript que reconoce el sublenguaje generado por *condition*:

```
condition = ->
  left = ___________
  type = _________._____
  match "COMPARISON"
  right = __________()
  result =
    type: type
    left: left
    right: right
  result
```

54. Escriba una función en CoffeeScript que retorne el factorial de un número

55. ¿Como se interpola en una cadena en CoffeeScript?.
    Como traducimos a una cadena de CoffeeScript este fragmento de código JavaScript:

```
"Filling the " + container + " with " + liquid
```

56. Traduzca a JavaScript el siguiente código CoffeeScript:

```
outer = 1
changeNumbers = ->
  inner = -1
  outer = 10
inner = changeNumbers()
```

57. Complete las salidas que faltan:

```
coffee> f = (n...) -> n
[Function]
coffee> f(1,2,3)
-----------------
coffee> a = [1,2,3]
coffee> f(a)
-----------------
coffee> f(a...)
-----------------
```

58. En CoffeeScript se usan Comprehensions para hacer bucles:

```
((x)->x*x)(y) for y in [1..6] when y % 2 == 0
```

    También se pueden usar para iterar en un objeto:

```
yearsOld = max: 10, ida: 9, tim: 11

ages = for child, age of yearsOld
  "#{child} is #{age}"
```

    a) ¿Cual es la salida de estos dos códigos?
    b) Escriba una comprehension que retorne los cubos de lo números impares entre 1 y 10

59. Explique este fragmento de código

```
Array::sum = -> @reduce ((a, b) -> a + b)
```

    y muestre un ejemplo de uso

60. Traduzca a Jade el siguiente código HTML:

```
<div id="content">
  <div class="block">
    <input id="bar" class="foo1 foo2"/>
  </div>
</div>
```

61. ¿Que hace el punto después de un elemento? como en:

```
p.
  foo bar
  hello world
```

62. Supongamos que el objeto interpolado es:

```
{"name": "Hello <em>World</em>"}
```

¿Que diferencia hay entre estas dos interpolaciones?

a) `li= name`

b) `li!= name`

63. ¿Que diferencia hay entre estas dos interpolaciones?

a) `li Say #{name}`

b) `li Say !{name}`

64. Traduzca a Jade el siguiente fragmento HTML:

```
<input type="text" placeholder="your name"/>
```

65. Rellene las líneas que faltan en esta página HTML:

a)
```
!DOCTYPE html>
<html>
<head>
<title>MathJax TeX Test Page</title>
<script type="text/x-mathjax-config">
  MathJax.Hub.Config({tex2jax: {inlineMath: [['$','$'], ['\\(','\\)']]}});
</script>
<script type="text/javascript"
  src="http://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML"
</script>
</head>
<body>
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
</body>
</html>
```

para que la página resultante produzca esta frase:

When $a \neq 0$*, there are two solutions to* $ax^2+bx+c=0$ *and they are* $x = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$.