Desarrollo de servicios en la nube con HTML5, Javascript y node.js Inicio Syllabus Foro Blog Módulos Modulo 1. Introducción a JavaScript de servidor y a node.js. Sentencias, Variables, Booleanos, Números, Strings y Funciones Modulo 0. Introducción al curso, al Prueba realizada programa y al Sistema Operativo UNIX Modulo 1. Introducción a JavaScript de Tu resultado en el test ha sido: 80% servidor y a node.js. Sentencias, Has superado el test. Variables, Booleanos, Números, Strings y Funciones Tus respuestas Tema 0: Transparencias del módulo Tema 1. Introducción a Javascript. Tipos Si tenemos las siguientes definiciones de funciones: y valores function f_1 (x) { function cero () { return 0; }; Tema 1. Cuestionario opcional function uno () { return 1; }; Tema 2. Programa, sentencia, variable y comentario if (x) { return uno; } else { return cero; } Tema 2. Cuestionario opcional }; Tema 3. Expresiones con variables Tema 3. Cuestionario opcional function f_2 (x) { return (x) ? function uno() { return 1; } : function cero() { return 0; }; Tema 4. Introducción node.js Tema 5. Booleano, igualdad y otros operadores lógicos Cómo se evaluará las siguiente expresión: f_2 ()() Tema 5. Cuestionario opcional 0 ✓ Tema 6. Sentencia IF/ELSE 0 1 Tema 7. Números Tema 7. Cuestionario obligatorio undefined Tema 8. Strings e internacionalización function cero() (I18N) function uno() Tema 8. Cuestionario opcional error_de_ejecución Tema 9. Funciones 7 Correct Tema 9. Cuestionario obligatorio ✓ Tema 10. Funciones como objetos y
. Si tenemos las siguientes definiciones de funciones: function f_1 (x) { Tema 10. Cuestionario obligatorio function cero () { return 0; }; Ejercicio P2P Opcional function uno () { return 1; }; Modulo 2. Introducción a JavaScript de servidor y a node.js. Bucles, Clases if (x) { return uno; } else { return cero; } predefinidas, Objetos, Propiedades y Métodos; Prototipos y Clases; Arrays; JSON; Funciones como Objetos y function f_2 (x) { Cierres (Closures) return (x) ? function uno() { return 1; } : function cero() { return 0; }; Modulo 3. Modulos node.js; **Expresiones Regulares; Eventos,** Cómo se evaluará las siguiente expresión: f_2 (7) Entorno de Ejecución y Concurrencia en node.js; Ficheros y Flujos 0 Modulo 4. Introducción a HTTP y a los 0 1 Servidores Web; Introducción a express y al Middleware Static; undefined Introducción a REST; Aplicaciones express.js y Composición de function cero() Middlewares; Formularios GET y POST; function uno() **Parámetros Ocultos** error_de_ejecución Modulo 5. Gestión de versiones de proyectos con git y GITHUB; Proyecto, Correct Espacio de Trabajo y Versiones (Commit); Arboles y Ramas de un Si tenemos las siguientes definiciones de funciones: proyecto; Repositorios Remoto y function f_1 (x) { colaboración a través de GITHUB function cero () { return 0; }; Modulo 6. Proyecto Quiz I: Patrón function uno () { return 1; }; Modelo-Vista-Controlador (MVC); generación del proyecto con expressgenerator; Primera Página y Primera if (x) { return uno; } else { return cero; } Pregunta; Despliegue en la nube }; (Heroku) function f_2 (x) { Modulo 7. Proyecto Quiz II: La Base de return (x)? function uno() { return 1; }: function cero() { return 0; }; Datos (DB), Tablas, sequelize.js y SQLite; Despliegue en Heroku utilizando Postgres; Presentación de Cómo se evaluará las siguiente expresión: f_1 (0)() Listas de Quizes y Autoload Modulo 8. Proyecto Quiz III: Gestión de . 0 Listas de Quizes, Creación, Edición y Borrado 0 1 Modulo 9. Proyecto Quiz IV: Creación y undefined Moderación de Comentarios a Quizes; function cero() Relaciones entre Tablas de la Base de Datos; Sesiones, Autenticación y function uno() Autorización; HTTP Seguro (HTTPS) error_de_ejecución Correct Si tenemos las siguientes definiciones de funciones: function f_1 (x) { function cero () { return 0; }; function uno () { return 1; }; if (x) { return uno; } else { return cero; } function f_2 (x) { return (x) ? function uno() { return 1; } : function cero() { return 0; }; Cómo se evaluará las siguiente expresión: f_1 (7) 0 @ 1 undefined function cero() function uno() error_de_ejecución Incorrect Si tenemos las siguientes definiciones de funciones: function f_1 (x) { function cero () { return 0; }; function uno () { return 1; }; if (x) { return uno; } else { return cero; } }; function f_2 (x) { return (x) ? function uno() { return 1; } : function cero() { return 0; }; Cómo se evaluará las siguiente expresión: f_1 ()() 0 1 undefined function cero() function uno() error_de_ejecución ncorrect 1 Si tenemos las siguientes definiciones de funciones: function f_1 (x) { function cero () { return 0; }; function uno () { return 1; }; if (x) { return uno; } else { return cero; } }; function f_2 (x) { return (x)? function uno() { return 1; }: function cero() { return 0; }; Cómo se evaluará las siguiente expresión: f_1 () 0 0 1 undefined function cero() function uno() error_de_ejecución Correct Si tenemos las siguientes definiciones de funciones: function f_1 (x) { function cero () { return 0; }; function uno () { return 1; }; if (x) { return uno; } else { return cero; } }; function f_2 (x) { return (x) ? function uno() { return 1; } : function cero() { return 0; }; Cómo se evaluará las siguiente expresión: f_2 () 0 0 1 undefined function cero() function uno() error_de_ejecución Correct Si tenemos las siguientes definiciones de funciones: function f_1 (x) { function cero () { return 0; }; function uno () { return 1; }; if (x) { return uno; } else { return cero; } }; function f_2 (x) { return (x)? function uno() { return 1; }: function cero() { return 0; }; Cómo se evaluará las siguiente expresión: f_2 (0)() . 0 0 1 undefined function cero() function uno() error_de_ejecución 7 Correct Si tenemos las siguientes definiciones de funciones: function f_1 (x) { function cero () { return 0; }; function uno () { return 1; }; if (x) { return uno; } else { return cero; } function f_2 (x) { return (x) ? function uno() { return 1; } : function cero() { return 0; }; Cómo se evaluará las siguiente expresión: f_1 (7)() 0 1 undefined function cero() function uno() error_de_ejecución Correct

> function f_2 (x) { return (x) ? function uno() { return 1; } : function cero() { return 0; }; Cómo se evaluará las siguiente expresión: f_2 (7)()

> > ¿Desea volver a acceder al test?

0 1

undefined function cero()

Si tenemos las siguientes definiciones de funciones:

function f_1 (x) {

};

0

function cero () { return 0; }; function uno () { return 1; };

if (x) { return uno; } else { return cero; }

error_de_ejecución 7 Correct

Mejorar nota

Has realizado 5 de 10 intentos

function uno()

Enhorabuena. Terminaste este módulo.

Telefónica Educación Digital

2012-2015 Miríada X | Aviso legal | Política de cookies | Política de privacidad

anterior

Siguiente 2