



Seguir

Sé el primero de tus amigos
en seguir a Miríada X.



Follow @miriadax



Mi Página

Cursos

Universidades e instituciones

Conócenos

Soporte

Desarrollo de servicios en la nube con HTML5, Javascript y node.js

[Inicio](#) | [Syllabus](#) | [Foro](#) | [Blog](#)

Módulos

- **Modulo 0. Introducción al curso, al programa y al Sistema Operativo UNIX**
- **Modulo 1. Introducción a JavaScript de servidor y a node.js. Sentencias, Variables, Booleanos, Números, Strings y Funciones**

- ✓ Tema 0: Transparencias del módulo
- ✓ Tema 1. Introducción a Javascript. Tipos y valores
- ✓ Tema 1. Cuestionario opcional
- ✓ Tema 2. Programa, sentencia, variable y comentario
- ✓ Tema 2. Cuestionario opcional
- ✓ Tema 3. Expresiones con variables
- ✓ Tema 3. Cuestionario opcional

Modulo 1. Introducción a JavaScript de servidor y a node.js. Sentencias, Variables, Booleanos, Números, Strings y Funciones

Tema 9. Cuestionario obligatorio

Prueba realizada

Tu resultado en el test ha sido: 100%

Has superado el test.

Tus respuestas

Si tenemos las siguientes definiciones de funciones y variables:

```
function f_1 (x) { return x; };
```

```
function f_2 () { return +arguments[0]; };
```

```
function f_3 (x) { return (x || 8) ; }
```

```
function f_4 (x) { x="8"; return x; }
```

Cómo se evaluarán la siguiente expresión: f_2 (null)

☐ 7

☐ "7"

✓ Tema 4. Introducción node.js

✓ Tema 5. Booleano, igualdad y otros operadores lógicos

✓ Tema 5. Cuestionario opcional

✓ Tema 6. Sentencia IF/ELSE

✓ Tema 7. Números

✓ Tema 7. Cuestionario obligatorio

✓ Tema 8. Strings e internacionalización (I18N)

✓ Tema 8. Cuestionario opcional

✓ Tema 9. Funciones

➔ Tema 9. Cuestionario obligatorio

✓ Tema 10. Funciones como objetos y cierres

Tema 10. Cuestionario obligatorio

Ejercicio P2P Opcional

● **Modulo 2. Introducción a JavaScript de servidor y a node.js. Bucles, Clases predefinidas, Objetos, Propiedades y Métodos; Prototipos y Clases; Arrays; JSON; Funciones como Objetos y Cierres (Closures)**

● **Modulo 3. Modulos node.js; Expresiones Regulares; Eventos, Entorno de Ejecución y Concurrencia en node.js; Ficheros y Flujos**

● **Modulo 4. Introducción a HTTP y a los Servidores Web; Introducción a**

☐ 8

☐ "8"

☐ undefined

☐ Null

☐ error_de_ejecución

☒ 0



Si tenemos las siguientes definiciones de funciones y variables:

```
function f_1 (x) { return x; };
```

```
function f_2 () { return +arguments[0]; };
```

```
function f_3 (x) { return (x || 8) ; }
```

```
function f_4 (x) { x="8"; return x; }
```

Cómo se evaluarán la siguiente expresión: f_2 (7,8)

☒ 7

☐ "7"

☐ 8

☐ "8"

☐ undefined

☐ null

☐ error_de_ejecución



Si tenemos las siguientes definiciones de funciones y variables:

```
function f_1 (x) { return x; };
```

```
function f_2 () { return +arguments[0]; };
```

```
function f_3 (x) { return (x || 8) ; }
```

```
function f_4 (x) { x="8"; return x; }
```

Cómo se evaluarán la siguiente expresión: f_1 (7,8)

☒ 7

☐ "7"

☐ 8

☐ "8"

☐ undefined

express y al Middleware Static; Introducción a REST; Aplicaciones express.js y Composición de Middlewares; Formularios GET y POST; Parámetros Ocultos

- **Modulo 5. Gestión de versiones de proyectos con git y GITHUB; Proyecto, Espacio de Trabajo y Versiones (Commit); Arboles y Ramas de un proyecto; Repositorios Remoto y colaboración a través de GITHUB**
- **Modulo 6. Proyecto Quiz I: Patrón Modelo-Vista-Controlador (MVC); generación del proyecto con express-generator; Primera Página y Primera Pregunta; Despliegue en la nube (Heroku)**
- **Modulo 7. Proyecto Quiz II: La Base de Datos (DB), Tablas, sequelize.js y SQLite; Despliegue en Heroku utilizando Postgres; Presentación de Listas de Quizes y Autoload**
- **Modulo 8. Proyecto Quiz III: Gestión de Listas de Quizes, Creación, Edición y Borrado**
- **Modulo 9. Proyecto Quiz IV: Creación y Moderación de Comentarios a Quizes; Relaciones entre Tablas de la Base de Datos; Sesiones, Autenticación y Autorización; HTTP Seguro (HTTPS)**

- ☐ null
- ☐ error_de_ejecución



Si tenemos las siguientes definiciones de funciones y variables:

```
function f_1 (x) { return x; };  
function f_2 () { return +arguments[0]; };  
function f_3 (x) { return (x || 8) ; }  
function f_4 (x) { x="8"; return x; }
```

Cómo se evaluarán la siguiente expresión: f_3 (7)

- ☒ 7
- ☐ "7"
- ☐ 8
- ☐ "8"
- ☐ undefined
- ☐ null
- ☐ error_de_ejecución



Si tenemos las siguientes definiciones de funciones y variables:

```
function f_1 (x) { return x; };  
function f_2 () { return +arguments[0]; };  
function f_3 (x) { return (x || 8) ; }  
function f_4 (x) { x="8"; return x; }
```

Cómo se evaluarán la siguiente expresión: f_3 (7,8)

- ☒ 7
- ☐ "7"
- ☐ 8
- ☐ "8"
- ☐ undefined
- ☐ null
- ☐ error_de_ejecución



Si tenemos las siguientes definiciones de funciones y variables:

```
function f_1 (x) { return x; };  
function f_2 () { return +arguments[0]; };  
function f_3 (x) { return (x || 8) ; }  
function f_4 (x) { x="8"; return x; }
```

Cómo se evaluarán la siguiente expresión: f_4 ()

- ☐ 7
- ☐ "7"
- ☐ 8
- ☒ "8"
- ☐ undefined
- ☐ null
- ☐ error_de_ejecución



Si tenemos las siguientes definiciones de funciones y variables:

```
function f_1 (x) { return x; };  
function f_2 () { return +arguments[0]; };  
function f_3 (x) { return (x || 8) ; }  
function f_4 (x) { x="8"; return x; }
```

Cómo se evaluarán la siguiente expresión: f_1 (null)

- ☐ 7
- ☐ "7"
- ☐ 8
- ☐ "8"
- ☐ undefined
- ☒ null
- ☐ error_de_ejecución



Si tenemos las siguientes definiciones de funciones y variables:

```
function f_1 (x) { return x; };  
function f_2 () { return +arguments[0]; };  
function f_3 (x) { return (x || 8) ; }  
function f_4 (x) { x="8"; return x; }
```

Cómo se evaluarán la siguiente expresión: f_4 (7,8)

- ☐ 7
- ☐ "7"
- ☐ 8
- ☒ "8"
- ☐ undefined
- ☐ null
- ☐ error_de_ejecución



Si tenemos las siguientes definiciones de funciones y variables:

```
function f_1 (x) { return x; };  
function f_2 () { return +arguments[0]; };  
function f_3 (x) { return (x || 8) ; }  
function f_4 (x) { x="8"; return x; }
```

Cómo se evaluarán la siguiente expresión: f_1 ()

- ☐ 7
- ☐ "7"
- ☐ 8
- ☐ "8"
- ☒ undefined
- ☐ null
- ☐ error_de_ejecución



Si tenemos las siguientes definiciones de funciones y variables:

```
function f_1 (x) { return x; };  
function f_2 () { return +arguments[0]; };  
function f_3 (x) { return (x || 8) ; }  
function f_4 (x) { x="8"; return x; }
```

Cómo se evaluarán la siguiente expresión: f_3 ()

- ☐ 7
- ☐ "7"
- ☒ 8



- ☐ "8"
- ☐ undefined
- ☐ null
- ☐ error_de_ejecución




 anterior

Siguiente 

Telefonica

Telefónica Educación Digital

 Santander

uni>ersia

2012-2015 Miríada X

Aviso legal

Política de cookies

Política de privacidad