Invocation

# Invocación

La invocación de una función suspende la ejecución de la función actual, pasando el control y los parámetros a la nueva función. Además de los parámetros declarados, toda función recibe dos parámetros adicionales: this y arguments. El parámetro this es muy importante en la programación orientada a objetos, y su valor viene determinado por el parámetro *invocation pattern*. Existen cuatro patrones de invocación en JavaScript: el patrón de invocación a métodos, el patrón de invocación a funciones, el patrón de invocación a constructores y el patrón de invocación a aplicaciones. Los patrones difieren en cómo se inicializa el parámetro de bonificación this.

El operador de invocación es un par de paréntesis que siguen a cualquier expresión que produzca un valor de función. Los paréntesis pueden contener cero o más expresiones, separadas por comas. Cada expresión produce un valor de argumento. Cada uno de los valores de los argumentos se asignará a los nombres de los parámetros de la función. No se produce ningún error de ejecución cuando el número de argumentos y el número de parámetros no coinciden. Si hay demasiados valores de argumento, se ignorarán los valores de argumento adicionales. Si hay muy pocos valores de argumentos, el valor undefined se sustituirá por los valores que falten. No hay comprobación de tipo en los valores de los argumentos: se puede pasar cualquier tipo de valor a cualquier parámetro.

[Subtema 1 de 5: (Ver todo)](ch04s03.html)[siguiente](ch04s03_2.html)