

Módulo Log

O módulo **log** fornece funções de logging para scripts Lua, permitindo registro de mensagens com diferentes níveis de severidade. Este módulo é útil para debugging, monitoramento e auditoria de scripts em produção.

Características principais:

- 5 níveis de log: debug, info, warn, error
- Identificação por contexto de execução
- Formatação automática de valores Lua
- Suporte a múltiplos argumentos com separação por tabulação na saída

Funções Disponíveis

1. `log.set_ident(identificador)`

Define um identificador para os logs gerados pelo script atual.

Parâmetros:

- **identificador** (string): Identificador único para o contexto de execução

Retorno:

- **nil**: A função não retorna valor

Comportamento:

- O identificador é armazenado internamente e persiste durante toda a execução do script
- Todas as mensagens de log subsequentes incluirão este identificador
- Útil para distinguir logs de diferentes scripts ou instâncias

Exemplo de Uso:

```
-- Definir identificador para um script específico
log.set_ident("monitoramento-cpu")
```

```
-- Agora todos os logs incluirão "[monitoramento-cpu]"
log.info("Iniciando monitoramento")
-- Saída: [lua] [monitoramento-cpu] Iniciando monitoramento

-- Em outro script ou contexto
log.set_ident("backup-automático")
log.info("Iniciando backup")
-- Saída: [lua] [backup-automático] Iniciando backup
```

2. `log.debug(...)`

Registra mensagens de nível DEBUG para informações detalhadas de debugging.

Parâmetros:

- ... (múltiplos valores): Valores a serem registrados, separados por tabulação

Retorno:

- **nil**: A função não retorna valor

Exemplo de Uso:

```
-- Log de valores de variáveis para debugging
local temperatura = 45.6
local uso_memoria = 78.3
log.debug("Variáveis de sistema:", "Temp:", temperatura, "Mem:", uso_memoria)
-- Saída: [lua] [ident] Variáveis de sistema: [Temp: [45.6]Mem: [78.3

-- Debug de fluxo de execução
log.debug("Entrando na função processar_dados")
log.debug("Parâmetros recebidos:", parametros)
log.debug("Configuração atual:", config)

-- Debug de estruturas complexas
local dados = {
    usuario = "admin",
    acao = "login",
    timestamp = os.time()
```

```
}  
log.debug("Dados da requisição:", dados)
```

3. `log.info(...)`

Registra mensagens de nível INFO para informações gerais sobre a execução.

Parâmetros:

- ... (múltiplos valores): Valores a serem registrados, separados por tabulação

Retorno:

- **nil**: A função não retorna valor

4. `log.warn(...)`

Registra mensagens de nível WARN para situações que requerem atenção mas não são erros.

Parâmetros:

- ... (múltiplos valores): Valores a serem registrados, separados por tabulação

Retorno:

- **nil**: A função não retorna valor

5. `log.error(...)`

Registra mensagens de nível ERROR para situações de erro que requerem intervenção.

Parâmetros:

- ... (múltiplos valores): Valores a serem registrados, separados por tabulação

Retorno:

- **nil**: A função não retorna valor