OutputInterface + const VERBOSITY QUIET + const VERBOSITY_NORMAL + const VERBOSITY_VERBOSE + const VERBOSITY_VERY _VERBOSE + const VERBOSITY_DEBUG + const OUTPUT_NORMAL + const OUTPUT_RAW + const OUTPUT_PLAIN + write(\$messages, bool \$newline=false, int \$options=0) + writeln(\$messages, int \$options=0) + setVerbosity(int \$level) + getVerbosity() + isQuiet() + isVerbose() + isVeryVerbose() + isDebug() + setDecorated(bool \$decorated) + isDecorated() + setFormatter(OutputFormatter Interface \$formatter) + getFormatter() Output OutputStyle __construct(?int _construct(OutputInterface \$verbosity=self::VERBOSITY **NullOutput** \$output) _NORMAL, bool \$decorated + newLine(int \$count=1) =false, OutputFormatterInterface + setFormatter(OutputFormatter + createProgressBar \$formatter=null) Interface \$formatter) (int max=0) + setFormatter(OutputFormatter + getFormatter() + write(\$messages, bool Interface \$formatter) \$newline=false, int + setDecorated(bool + getFormatter() \$type=self::OUTPUT_NORMAL) \$decorated) + setDecorated(bool + writeln(\$messages, + isDecorated() \$decorated) ConsoleOutputInterface int \$type=self::OUTPUT + setVerbosity(int + isDecorated() _NORMAL) \$level) + setVerbosity(int + setVerbosity(int + getErrorOutput() + getVerbosity() \$level) \$level) + setErrorOutput(OutputInterface + isQuiet() + getVerbosity() + getVerbosity() \$error) + isVerbose() + isQuiet() + setDecorated(bool + section() + isVeryVerbose() \$decorated) + isVerbose() + isDecorated() + isDebug() + isVeryVerbose() + setFormatter(OutputFormatter + writeln(\$messages, + isDebug() Interface \$formatter) int \$options=self:: + writeln(\$messages, OUTPUT_NORMAL) + getFormatter() int \$options=self:: + write(\$messages, bool OUTPUT_NORMAL) + isQuiet() \$newline=false, int + write(\$messages, bool + isVerbose() \$options=self::OUTPUT NORMAL) \$newline=false, int + isVeryVerbose() \$options=self::OUTPUT_NORMAL) + isDebug() # doWrite(string \$message, # getErrorOutput() bool \$newline) SymfonyStyle + const MAX_LINE_LENGTH + __construct(InputInterface \$input, OutputInterface \$output) StreamOutput + block(\$messages, string TrimmedBufferOutput \$type=null, string \$style _construct(\$stream, =null, string \$prefix=' ' BufferedOutput int \$verbosity=self bool \$padding=false, bool construct(int \$maxLength, ::VERBOSITY_NORMAL, bool ?int \$verbosity=self::VERBOSITY \$escape=true) \$decorated=null. OutputFormatter NORMAL, bool \$decorated=false, + fetch() + title(string \$message) Interface \$formatter=null) OutputFormatterInterface \$formatter=null) # doWrite(string \$message, + section(string \$message) + getStream() + fetch() bool \$newline) + listing(array \$elements) # doWrite(string \$message, # doWrite(string \$message, + text(\$message) bool \$newline) bool \$newline) + comment(\$message) # hasColorSupport() + success(\$message) + error(\$message) + warning(\$message) et 21 de plus... ConsoleOutput ConsoleSectionOutput + __construct(int \$verbositv =self::VERBOSITY_NORMAL, bool \$decorated=null, OutputFormatter + __construct(\$stream, Interface \$formatter=null) array &\$sections, int + section() \$verbosity, bool \$decorated, OutputFormatterInterface + setDecorated(bool \$formatter) \$decorated) + clear(int \$lines=null) + setFormatter(OutputFormatter Interface \$formatter) + overwrite(\$message) + setVerbosity(int + getContent() \$level) + addContent(string + getErrorOutput() \$input) + setErrorOutput(OutputInterface # doWrite(string \$message, \$error) bool \$newline) # hasStdoutSupport() # hasStderrSupport()