



### Basic Commands & Syntax

#### Essential Commands

<b>ECHO</b>	Displays messages on the console. <b>Example:</b> <code>ECHO Hello, world!</code>
<b>@</b>	Suppresses command echoing. <b>Example:</b> <code>@ECHO OFF</code>
<b>SET</b>	Defines or modifies environment variables. <b>Example:</b> <code>SET myVar=value</code>
<b>PAUSE</b>	Pauses script execution and waits for user input. <b>Example:</b> <code>PAUSE</code>
<b>CLS</b>	Clears the console screen. <b>Example:</b> <code>CLS</code>
<b>EXIT</b>	Exits the batch script. <b>Example:</b> <code>EXIT</code>
<b>REM</b> or <b>::</b>	Adds comments to the script. <b>Example:</b> <code>REM This is a comment</code> or <code>:: This is also a comment</code>

#### Variables

<code>SET variable_name=value</code>	- Assigns a value to a variable.
<code>%variable_name%</code>	- Accesses the variable's value.
<b>Example:</b>	<pre>SET name=John ECHO Hello, %name%</pre>
<code>SET /P variable_name=Prompt message:</code>	- Prompts the user for input.
<b>Example:</b>	<pre>SET /P input=Enter your name: ECHO Your name is %input%</pre>
<code>!variable_name!</code>	- Enables delayed expansion (needed within loops).
<b>Example:</b>	<pre>SETLOCAL EnableDelayedExpansion FOR /L %%i IN (1,1,5) DO (     SET /A num+=1     ECHO !num! ) ENDLOCAL</pre>

#### Operators

Arithmetic	<code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>%</code> (Modulo)
Comparison	<code>EQU</code> (equal), <code>NEQ</code> (not equal), <code>LSS</code> (less than), <code>LEQ</code> (less than or equal), <code>GTR</code> (greater than), <code>GEQ</code> (greater than or equal)
Logical	<code>AND</code> , <code>OR</code> , <code>NOT</code> (within <code>IF</code> statements)

### Control Flow

#### Conditional Statements

<code>IF condition (commands) ELSE (commands)</code>	- Executes commands based on a condition.
<b>Example:</b>	<pre>IF %errorlevel% NEQ 0 (     ECHO An error occurred. ) ELSE (     ECHO Script completed successfully. )</pre>
<code>IF EXIST filename (commands)</code>	- Checks if a file exists.
<b>Example:</b>	<pre>IF EXIST myfile.txt (     ECHO File exists. ) ELSE (     ECHO File does not exist. )</pre>
<code>IF DEFINED variable (commands)</code>	- Checks if a variable is defined.
<b>Example:</b>	<pre>IF DEFINED myVar (     ECHO Variable is defined. ) ELSE (     ECHO Variable is not defined. )</pre>

#### Looping

<code>FOR %variable IN (set) DO command</code>	- Iterates over a set of items.
<code>FOR /L %variable IN (start,step,end) DO command</code>	- Iterates a numerical range.
<code>FOR /F %variable IN (file) DO command</code>	- Parses a file line by line.
<b>Examples:</b>	<pre>FOR %%i IN (apple banana cherry) DO ECHO %%i  FOR /L %%i IN (1,1,5) DO ECHO %%i  FOR /F "tokens=*" %%i IN (myfile.txt) DO ECHO %%i</pre>

#### GOTO and Labels

<code>GOTO label</code>	- Jumps to a specified label.
<code>:label</code>	- Defines a label.
<b>Example:</b>	<pre>:start ECHO Doing something... GOTO end  :end ECHO Script finished.</pre>

### File and Directory Operations

## File Manipulation

<code>COPY</code>	Copies files. <b>Example:</b> <code>COPY file.txt newfile.txt</code>
<code>MOVE</code>	Moves files. <b>Example:</b> <code>MOVE file.txt newlocation\file.txt</code>
<code>DEL</code>	Deletes files. <b>Example:</b> <code>DEL file.txt</code>
<code>RENAME</code> or <code>REN</code>	Renames files. <b>Example:</b> <code>REN oldfile.txt newfile.txt</code>
<code>TYPE</code>	Displays the content of the file. <b>Example:</b> <code>TYPE file.txt</code>

## Directory Manipulation

<code>MKDIR</code> or <code>MD</code>	Creates a new directory. <b>Example:</b> <code>MKDIR newdirectory</code>
<code>RMDIR</code> or <code>RD</code>	Removes a directory. <b>Example:</b> <code>RMDIR directorytodelete</code>
<code>CHDIR</code> or <code>CD</code>	Changes the current directory. <b>Example:</b> <code>CD newdirectory</code>
<code>DIR</code>	Lists files and directories in the current directory. <b>Example:</b> <code>DIR</code>

## File Attributes

<code>ATTRIB</code> or <code>B</code>	Displays or changes file attributes. <b>Example:</b> <code>ATTRIB +R file.txt</code> (sets read-only attribute)
---------------------------------------	--

## Advanced Techniques

### Subroutines

<code>CALL :label</code>	- Calls a subroutine.
<code>:label</code>	- Defines the subroutine.
<code>EXIT /B</code>	- Exits the subroutine (returns to the caller).

**Example:**

```
CALL :mySubroutine
ECHO Back in main script
EXIT

:mySubroutine
ECHO Inside subroutine
EXIT /B
```

### Error Handling

<code>IF %ERRORLEVEL% NEQ 0</code>	- Checks the error level of the last command.
------------------------------------	---

**Example:**

```
command_that_might_fail
IF %ERRORLEVEL% NEQ 0 (
    ECHO An error occurred.
    EXIT /B %ERRORLEVEL%
)
```

### Command Chaining

<code>command1 &amp;&amp; command2</code>	- Executes <code>command2</code> only if <code>command1</code> succeeds.
<code>command1    command2</code>	- Executes <code>command2</code> only if <code>command1</code> fails.

**Example:**

```
dir && echo Directory listing successful
del non_existent_file || echo File not found
```

### Piping and Redirection

<code>&gt;</code>	Redirects output to a file (overwrites existing). <b>Example:</b> <code>DIR &gt; filelist.txt</code>
<code>&gt;&gt;</code>	Redirects output to a file (appends). <b>Example:</b> <code>ECHO Additional info &gt;&gt; filelist.txt</code>
<code> </code>	Pipes output from one command to another. <b>Example:</b> <code>DIR   FIND "txt"</code>