

<b>Echo komunikat</b>	Wyświetla <i>komunikat</i> na ekranie. <a href="#">Więcej...</a>
<b>Echo [on   off]</b>	Włącza i wyłącza komunikaty wyświetlane na ekranie. <a href="#">Więcej...</a>
<b>@</b>	Użyta na początku linii zapobiega jej wyświetleniu się. <a href="#">Więcej...</a>
<b>Rem komentarz</b>	Użyte na początku linii wstawia do programu komentarz. <a href="#">Więcej...</a>
<b>Goto etykieta</b>	Instrukcja sterująca nakazująca przeskok programu do miejsca w kodzie oznaczonym <i>etykieta</i> . <a href="#">Więcej...</a>
<b>:etykieta</b>	Zaznaczenie miejsca w kodzie, do którego program ma przeskakiwać. <a href="#">Więcej...</a>
<b>Shift</b>	Powoduje przesunięcie parametrów przy wywołaniu programu wsadowego. <a href="#">Więcej...</a>
<b>If [not] Exist plik komenda</b>	Instrukcja warunkowa, wykonująca <i>komendę</i> jeżeli istnieje <i>plik</i> . <a href="#">Więcej...</a>
<b>If [not] ciag1==ciag2 komenda</b>	Instrukcja warunkowa, wykonująca <i>komendę</i> jeżeli <i>ciag1</i> jest równy <i>ciag2</i> . <a href="#">Więcej...</a>
<b>If [not] Errorlevel liczba komenda</b>	Instrukcja warunkowa, wykonująca <i>komendę</i> jeżeli kod zwracany przez poprzednią instrukcję jest równy <i>liczba</i> . <a href="#">Więcej...</a>
<b>For %%x In (y) do komenda %%x</b>	Pętla wykonująca się tyle razy ile jest elementów w <i>y</i> przy każdym przejściu wykonująca <i>komendę</i> , która może być zastosowana do kolejnego elementu <i>x</i> pochodzącego ze zbioru <i>y</i> . <a href="#">Więcej...</a>
<b>Pushd sciezka</b>	Zapamiętuje katalog, a następnie przechodzi do folderu oznaczonego <i>sciezka</i> . <a href="#">Więcej...</a>
<b>Popd</b>	Przechodzi do katalogu zapamiętanego przez polecenie Pushd. <a href="#">Więcej...</a>
<b>Setlocal</b>	Informuje system, że zmiany dokonane przez program są lokalne. <a href="#">Więcej...</a>
<b>Endlocal</b>	Informuje system, o zakończeniu sekcji zmian lokalnych dokonanych przez program. <a href="#">Więcej...</a>
<b>Set zmienna = wartosc</b>	Ustala wartość zmiennej. <a href="#">Więcej...</a>
<b>Pause</b>	Zatrzymuje program i czeka na naciśnięcie klawisza przez użytkownika. <a href="#">Więcej...</a>
<b>Title tytuł</b>	Zmienia tytuł okna wiersza polecenia. <a href="#">Więcej...</a>
<b>Call program_wsadowy</b>	Wywołuje inny program wsadowy, czeka na jego zakończenie po czym wznowia swoją pracę. <a href="#">Więcej...</a>

SHIFT /2

przesunie %3 do %2, %4 do %3 itd. i pozostawi %0 oraz %1.

## IF exist przykład

```
1: @echo off
2: echo .>test.txt
3: dir t* /b
4: echo ----
5: if exist test.txt del test.txt
6: dir t* /b
7: echo ----
8: if not exist test.txt echo .>test.txt
9: dir t* /b
```

W drugiej linijce tworzymy plik **test.txt** wpisując do niego jedną kropkę. Następnie wyświetlamy zawartość katalogu poleceniem **dir** aby nie zaśmiecać ekranu wyświetlamy tylko pliki, których nazwa zaczyna się od litery **t**. Przełącznikiem **/b** sprawiamy, że polecenie **dir** nie wyświetla informacji o dysku, na którym dane pliki się znajdują. Trzeci wiersz pokazuje nam, że plik utworzył się i znajduje się na dysku. W linijce czwartej wyświetlamy na ekranie cztery myślniki aby wynik działania programu wsadowego był bardziej przejrzysty. W piątym wierszu pierwszy raz zastosowaliśmy instrukcję warunkową. Sprawdza ona mianowicie czy plik **test.txt** istnieje, jeżeli dokument znajduje się na dysku (jak jest w naszym przypadku) program usuwa go. Potwierdzeniem poprawnego działania programu jest próba ponownego wyświetlenia pliku **test.txt**, która kończy się niepowodzeniem i wyświetleniem na ekranie komunikatu o braku danego pliku. Wiersz ósmy jest kolejnym przykładem instrukcji **if**. Tym razem sprawdza ona czy dany plik nie istnieje. Ponieważ usunęliśmy go w wierszu piątym wynik porównania jest prawdziwy i wykonuje się instrukcja, która ponownie tworzy ten plik.

```
1: @echo off
2: echo .>test.txt
3: echo .>test2.txt
4: dir *.txt /b
5: echo ----
6: if not exist "dokumenty tekstowe" mkdir "dokumenty tekstowe"
7: move *.txt "dokumenty tekstowe">nul
8: dir t* /b
9: echo ----
10: cd "dokumenty tekstowe"
11: cd
12: dir t* /b
```

Instrukcję omawianą możemy również zastosować do sprawdzania istnienia katalogów. Na początku tworzymy dwa pliki tekstowe. Następnie w wierszu szóstym sprawdzamy czy katalog **dokumenty tekstowe** istnieje. Jeżeli go nie ma tworzymy go. W linijce siódmej przenosimy oba wcześniej utworzone pliki do ww. katalogu. Przy przenoszeniu plików komenda **move** wyświetla na ekranie pliki, których zmienia lokalizację. Aby zapobiec pojawieniu się tych komunikatów przekierowujemy je za pomocą komendy **>nul**. W wierszu dziesiątym wchodzimy do utworzonego katalogu. W linijce jedenastej wyświetlamy za pomocą polecenia **cd** bez żadnych parametrów aktualny katalog.

```

1: @echo off
2: echo .>test.txt
3: dir *.txt /b
4: if exist test.txt (
5: echo Plik istnieje.
6: ) else (
7: echo Plik nie istnieje.
8: )
9: del test.txt
10: echo ----
11: dir *.txt /b
12: if exist test.txt (
13: echo Plik istnieje.
14: ) else (
15: echo Plik nie istnieje.
15: )

```

W instrukcji warunkowej możemy również zastosować komendę **else**, która wykonuje się jeżeli warunek postawiony nie został spełniony. W wierszu drugim tworzymy plik **test.txt**. Wiersze od czwartego do ósmego sprawdzają czy dany plik istnieje i wyświetlają komunikat. W linii dziewiątej kasujemy plik a następnie w wierszach od dwunastego do piętnastego powtarzamy ciąg instrukcji, sprawdzający istnienie pliku. Tym razem pliku nie ma dlatego też wyświetli się inny komunikat niż poprzednio.

#### Przykład menu

```

@echo off
:menu
title Menu
echo Menu
echo 1) Pierwsza opcja
echo 2) Kolejna opcja
set /p letter=
if %letter% == 1 goto opcja1
if %letter% == 2 goto opcja2

:opcja1
echo opcja 1
pause
goto menu:
exit

:opcja2
echo opcja 2
pause
goto menu:
exit

```

ASSOC Wyświetla lub modyfikuje skojarzenia rozszerzeń plików.

ATTRIB Wyświetla lub zmienia atrybuty pliku.

BREAK Wyłącza lub włącza rozszerzone sprawdzanie klawiszy CTRL+C.

BCDEDIT Ustawia właściwości w bazie danych rozruchu w celu kontroli ładowania rozruchu.

CACLS Wyświetla lub modyfikuje listy kontroli dostępu (ACL) plików.

CALL Wywołuje jeden program wsadowy z innego.

CD Wyświetla lub zmienia nazwę bieżącego katalogu.

CHCP Wyświetla lub ustawia numer aktywnej strony kodowej.

CHDIR Wyświetla lub zmienia nazwę bieżącego katalogu.

CHKDSK Sprawdza dysk i wyświetla raport o jego stanie.

CHKNTFS Wyświetla lub modyfikuje stan sprawdzania dysku przy rozruchu.

CLS Czyści ekran.

CMD Uruchamia nowe wystąpienie interpretera poleceń systemu Windows.

COLOR Ustawia domyślny kolor tła i pierwszego planu konsoli.

COMP Porównuje zawartość dwóch plików lub zestawów plików.

COMPACT Wyświetla i pozwala na modyfikację listy plików, skompresowanych na partycji NTFS.

CONVERT Konwertuje woluminy FAT na woluminy NTFS. Nie można konwertować bieżącego dysku.

COPY Kopiuje jeden lub wiele plików do innej lokalizacji.

DATE Wyświetla lub ustawia datę.

DEL Usuwa jeden lub wiele plików.

DIR Wyświetla listę plików i podkatalogów katalogu.

DISKCOMP Porównuje zawartość dwóch dyskietek.

DISKCOPY Kopiuje zawartość jednej dyskietki na drugą.

DISKPART Wyświetla lub konfiguruje właściwości partycji dysków.

DOSKEY Edytuje wiersze poleceń, przywraca polecenia Windows i tworzy makra.

DRIVERQUERY Wyświetla bieżący stan i właściwości sterownika urządzenia.

ECHO Wyświetla komunikaty lub wyłącza i włącza echo poleceń.

ENDLOCAL Kończy lokalne zmiany środowiska w plikach wsadowych.

ERASE Usuwa jeden lub wiele plików.

EXIT Kończy działanie programu CMD.EXE (interpretera poleceń).

FC Porównuje dwa pliki lub zestawy plików i wyświetla różnice między nimi.

FIND Wyszukuje ciąg tekstowy w pliku lub plikach.

FINDSTR Wyszukuje ciągi znaków w plikach.

FOR Uruchamia podane polecenie dla każdego pliku z zestawu plików.

FORMAT Formatuje dysk do użytku w systemie Windows.

FSUTIL Wyświetla lub konfiguruje właściwości systemu plików.

FTYPE Wyświetla i modyfikuje typy plików używane w skojarzeniach rozszerzeń plików.

GOTO Przekazuje sterowanie interpretera poleceń systemu Windows do wiersza o podanej etykiecie w pliku wsadowym.

GPRESULT Wyświetla informacje zasad grupy dla komputera lub użytkownika.

GRAFTABL Włącza wyświetlanie przez system Windows znaków rozszerzonych w trybie graficznym.

HELP Podaje informacje o poleceniach systemu Windows.

ICACLS Umożliwia wyświetlanie i modyfikowanie list ACL dla plików oraz katalogów, a także wykonywanie i przywracanie kopii zapasowych tych list.

IF Dokonuje warunkowego przetwarzania w plikach wsadowych.

LABEL Tworzy, zmienia lub usuwa etykietę(etykiety) woluminu dysku.

MD Tworzy katalog.

MKDIR Tworzy katalog.

MKLINK Tworzy łącza symboliczne i łącza twarde.

MODE Konfiguruje urządzenia systemowe.  
 MORE Wyświetla informacje ekran po ekranie.  
 MOVE Przenosi jeden lub wiele plików z jednego katalogu do drugiego.  
 OPENFILES Wyświetla pliki otwarte przez użytkowników zdalnych dla udziału plików.  
 PATH Wyświetla lub ustawia ścieżkę przeszukiwania dla plików wykonywalnych.  
 PAUSE Wstrzymuje przetwarzanie pliku wsadowego i wyświetla komunikat.  
 POPD Przywraca katalog zapisany przez PUSH.D.  
 PRINT Drukuje plik tekstowy.  
 PROMPT Zmienia znak zachęty wiersza polecenia systemu Windows.  
 PUSH.D Zapisuje bieżący katalog, a następnie zamienia go.  
 RD Usuwa katalog.  
 RECOVER Odzyskuje możliwe do odczytania informacje z uszkodzonego dysku.  
 REM Oznacza komentarze w pliku wsadowym lub pliku CONFIG.SYS.  
 REN Zmienia nazwę pliku lub plików.  
 RENAME Zmienia nazwę pliku lub plików.  
 REPLACE Zamienia pliki.  
 RMDIR Usuwa katalog.  
 ROBOCOPY Zaawansowane narzędzie do kopiowania plików i drzew katalogów.  
 SET Wyświetla, ustawia lub usuwa zmienne środowiskowe systemu Windows.  
 SETLOCAL Rozpoczyna lokalne zmiany środowiska w plikach wsadowych.  
 SC Wyświetla lub konfiguruje usługi (procesy w tle).  
 SCHEDULE Planuje uruchamianie poleceń i programów na komputerze.  
 SHIFT Zmienia pozycje parametrów w plikach wsadowych.  
 SHUTDOWN Zezwala na lokalne lub zdalne wyłączenie komputera.  
 SORT Sortuje wprowadzone dane.  
 START Uruchamia program lub polecenie w oddzielnym oknie.  
 SUBST Przypisuje ścieżce literę dysku.  
 SYSTEMINFO Wyświetla właściwości i konfigurację komputera.  
 TASKLIST Wyświetla wszystkie uruchomione zadania łącznie z usługami.  
 TASKKILL Kończy lub zatrzymuje uruchomiony proces lub aplikację.  
 TIME Wyświetla lub ustawia czas systemowy.  
 TITLE Ustawia tytuł okna dla sesji CMD.EXE.  
 TREE Wyświetla strukturę katalogów dysku lub ścieżki.  
 TYPE Wyświetla zawartość pliku tekstowego.  
 VER Wyświetla wersję systemu Windows.  
 VERIFY Instruuje system Windows, aby sprawdzał poprawność zapisu plików na dysku.  
 VOL Wyświetla etykietę oraz numer seryjny dysku.  
 XCOPY Kopiuje pliki i katalogi.  
 WMIC Wyświetla informacje WMI wewnątrz interaktywnej powłoki poleceń.

Nasze programy wsadowe na pewno chcielibyśmy uruchamiać będąc w każdym katalogu albo też korzystając z menu **Uruchom**. Dlatego też powinniśmy umieścić je w jednym folderze a następnie zmodyfikować zmienną **path** tak aby przeglądała ten katalog w poszukiwaniu naszych programów. O ile przeniesienie wszystkich programów wsadowych do jednego katalogu jest dość proste to zmodyfikowanie zmiennej systemowej może sprawić niejakie trudności. Jednak kłopot ten jest tylko pozorny. Aby dodać do zmiennej **path** kolejny **folder**, w którym umieszczone są nasze programy wpisujemy w konsoli następujące polecenie: **set path=%path%;folder**. (Pomiędzy %path% a folder znajduje się średnik, folder jest natomiast ścieżką do katalogu z programami) Przykładowe wywołanie mogłoby wyglądać następująco: **set path=%path%;c:\moje\_programy**