

Poniżej znajdziesz krótkie charakterystyki poszczególnych grup funkcji arkuszowych Excel'a. Pominięta jest tylko grupa funkcji inżynierskich wymagająca zainstalowania dodatku Analysis Toolpack. W arkuszu można korzystać z ponad 200 funkcji. Tutaj przedstawione są jedynie opisy funkcji najczęściej używanych (tych, które powinienś znać). Dla bardziej skomplikowanych funkcji lub ich mniej typowego użycia dołączone są również przykłady.

## Funkcje matematyczne i trygonometryczne

Za pomocą funkcji matematycznych i trygonometrycznych można wykonywać proste i złożone obliczenia matematyczne takie, jak obliczanie całkowitej sumy zakresu komórek lub sumy zakresu komórek w zakresie, które spełniają warunek w innym zakresie albo zaokrąglanie liczb.

**MOD** - wartość reszty dzielenia liczb całkowitych; wynik ma znak dzielnika

Składnia: **MOD(liczba ; dzielnik)**

**liczba**                      liczba, dla której należy wyznaczyć resztę

**dzielnik**                    liczba, przez którą należy podzielić; jeśli dzielnik jest równy **0**, **MOD** podaje wartość błędu **#DZIEL/0!**

## MODUŁ.LICZBY - wartość bezwzględną liczby

Składnia: **MODUŁ.LICZBY(liczba)**

**liczba**                                      liczba rzeczywista, której wartość bezwzględną należy obliczyć

## SUMA - suma wszystkich liczb w liście argumentów.

Składnia: **SUMA(liczba1 ; liczba2 ; ...)**

**liczba1, liczba2,...**                      do 30 argumentów, które chcemy zsumować

### Uwagi:

- Obliczane są liczby, wartości logiczne, reprezentacje tekstowe, które zostaną wpisane bezpośrednio do listy argumentów.
- Jeśli argument jest tablicą lub adresem , to sumy obliczane są tylko w tej tablicy lub adresie. Pominięte zostaną puste komórki, wartości logiczne, tekst lub wartości błędów w tablicy lub adresie.
- Argumenty, które są wartościami błędów lub tekstem, których nie można przekształcić, powodują występowanie błędów.

*Przykłady:*

**SUMA("3" ; 2 ; PRAWDA)** jest równe **6**: tekst przekształcany w liczby, a wartość **PRAWDA** w liczbę **1**

Jeżeli **A1** zawiera **"3"**, a **B1** zawiera **PRAWDA**, wtedy:

**SUMA(A1 ; B1 ; 2)** jest równe **2**, ponieważ adresy nie są przekształcane do wartości nieliczbowych w adresach

**SUMA.JEŻELI** - sumuje komórki spełniające podane kryteria.

Składnia: **SUMA.JEŻELI(zakres ; kryteria ; suma\_zakres)**

**zakres**               zakres komórek, które należy przeliczyć

**kryteria**             kryteria w postaci liczby, wyrażenia lub tekstu określające, które komórki będą dodane

**suma\_zakres**       komórki do zsumowania; sumowane tylko wtedy, gdy odpowiadające im komórki w zakresie spełniają kryterium. Jeśli **suma\_zakres** zostaje pominięte, to sumowane są komórki w zakresie

*Przykład:*

Zakładając, że **A1:A4** zawiera odpowiednio wartości: **100, 200, 300, 400**, a **B1:B4**: **7, 14, 21, 28**, to:

**SUMA.JEŻELI(A1:A4 ; ">160" ; B1:B4)** zwróci **63**

**ILOCZYN** - mnoży wszystkie argumenty i podaje w wyniku ich iloczyn.

Składnia: **ILOCZYN(liczba1 ; liczba2 ; ...)**

**liczba1, liczba2,...** liczby (do 30), które chcemy pomnożyć

- Obliczane są argumenty będące liczbami, wartościami logicznymi lub tekstowymi reprezentacjami liczb. Argumenty typu numery błędów czy teksty, których nie można przekształcić na liczby, spowodują błędy.
- Jeśli argument jest tablicą lub adresem, to obliczane będą wyłącznie liczby zawarte w danej tablicy czy adresie. Puste komórki, wartości logiczne, tekst lub wartości błędów są ignorowane.

*Przykłady:*

Jeśli komórki **A2:C2** zawierają **5, 15 i 30**:

**ILOCZYN(A2:C2)** jest równe **2250**

**ILOCZYN(A2:C2 ; 2)** jest równe **4500**

**SILNIA** - oblicza wartość silni.

Składnia: **SILNIA(liczba)**

**liczba** nieujemna liczba całkowita, której silnię należy obliczyć. Jeśli liczba nie jest całkowita, zostanie obcięta do liczby całkowitej

Przykład:

**SILNIA(1,9)** jest równe **SILNIA(1)** jest równe 1

**PIERWIASTEK** - dodatni pierwiastek kwadratowy liczby.

Składnia: **PIERWIASTEK(liczba)**

**liczba**                      liczbą, dla której obliczany będzie pierwiastek. Dla argumentu ujemnego, funkcja podaje w wyniku wartość błędu **#LICZBA!**

**POTĘGA** - wartość liczby podniesionej do potęgi.

Składnia: **POTĘGA(liczba ; potęga)**

**liczba**                      podstawa potęgi (dowolna liczba rzeczywista)

**potęga**                      wykładnik potęgi, do jakiej podnoszona jest podstawa

Operatora "^" można używać zamiast **POTĘGA**, aby wskazać, do której potęgi ma być podniesiona podstawa, jak np. **5^2**.

**EXP** - wartość e do potęgi liczba. ( $e = 2,71828182845904$ ).

Składnia: **EXP(liczba)**

**liczba**                                      wykładniki potęgi

**EXP** jest funkcją odwrotną do **LN**, logarytmu naturalnego liczby.

**LN** - wartość logarytmu naturalnego (o podstawie  $e=2,71828182845904$ ).

Składnia: **LN(liczba)**

**liczba**                                      liczba rzeczywista dodatnia, której logarytm naturalny należy obliczyć

**LOG** - wartość logarytmu liczby przy zadanej podstawie.

Składnia: **LOG(liczba, podstawa)**

**liczba**                                      dodatnia liczba rzeczywista, której logarytm należy obliczyć

**podstawa**                                      postawa logarytmu (*domyślnie przyjmuje się 10*)

**LOG10** - wartość logarytmu liczby przy podstawie 10.

Składnia: **LOG10(liczba)**

**liczba**                                      dodatnia liczba rzeczywista, której logarytm przy podstawie 10 należy wyznaczyć

**ZAOKR** - zaokrągla liczbę do określonej liczby cyfr.

Składnia: **ZAOKR(liczba ; liczba\_cyfr)**

**liczba**                                   liczbą do zaokrąglenia

**liczba\_cyfr**                           liczba cyfr, do których ma być zaokrąglona liczba

*Przykład:*

**ZAOKR(-1,475 ; 2)** jest równe **-1,48**

**ZAOKR(21,5 ; -1)** jest równe **20**

**LICZBA.CAŁK** - obcina liczbę do części całkowitej usuwając część ułamkową.

Składnia: **LICZBA.CAŁK(liczba, liczba\_cyfr)**

**liczba**                                   liczba, która ma być obcięta

**liczba\_cyfr**                           liczba określającą dokładność obcinania (*domyślnie 0*)

Funkcje **LICZBA.CAŁK** i **ZAOKR.DO.CAŁK** są podobne w tym, że obydwie obliczają wartości całkowite.

**LICZBA.CAŁK** usuwa ułamkową część liczby, **ZAOKR.DO.CAŁK** zaokrągla liczby w dół do najbliższej liczby całkowitej względem ułamkowej części liczby. Funkcje **ZAOKR.DO.CAŁK** i **LICZBA.CAŁK** są różne tylko w przypadku liczb ujemnych: **LICZBA.CAŁK(-4.3)** podaje w wyniku **-4**, a **ZAOKR.DO.CAŁK(-4.3)** podaje w wyniku **-5**, ponieważ **-5** jest liczbą mniejszą.

**ZAOKR.DO.CAŁK** - zaokrągla liczbę w dół do najbliższej liczby całkowitej.

Składnia: **ZAOKR.DO.CAŁK(liczba)**

**liczba**

liczba rzeczywista, którą należy zaokrąglić

*Przykład:*

**ZAOKR.DO.CAŁK(8,9)** jest równe **8**

**ZAOKR.DO.CAŁK(-8,9)** jest równe **-9**

Formuła podająca dziesiątą część liczby rzeczywistej > 0 w komórce **A1**:

**A1-ZAOKR.DO.CAŁK(A1)**

**ZAOKR.DÓŁ** - zaokrągla liczbę w dół w kierunku zera.

Składnia: **ZAOKR.DÓŁ(liczb ; liczba\_cyfr)**

**liczba**

liczbą rzeczywistą, która ma być zaokrąglona w dół

**liczba\_cyfr**

liczba cyfr, do ilu liczba ma być zaokrąglona

*Przykład:*

**ZAOKR.DÓŁ(76,9 ; 0)** jest równe **76**

**ZAOKR.DÓŁ(31415,92654 ; -2)** jest równe **31400**



**ZAOKR.W.DÓŁ** - zaokrągla liczbę w dół, w kierunku zera, do najbliższej wielokrotności znaczenia.

Składnia: **ZAOKR.W.DÓŁ(liczba ; znaczenie)**

<b>liczba</b>	wartość liczbowa, którą należy zaokrąglić
<b>znaczenie</b>	wielokrotność, do której należy zaokrąglić

**Uwagi:**

- Jeśli argumenty liczba i znaczenie są różnych znaków, **ZAOKR.W.DÓŁ** podaje wartość błędu **#LICZBA!**.
- Niezależnie od znaku liczby, wartość jest zaokrąglana w dół przy odsuwaniu się od zera. Jeśli liczba jest dokładną wielokrotnością argumentu znaczenie, nie zostanie zaokrąglona.

*Przykłady:*

**ZAOKR.W.DÓŁ(2,5 ; 1)** jest równe 2

**ZAOKR.W.DÓŁ(-2,5 ; -2)** jest równe -2

**ZAOKR.W.DÓŁ(1,5 ; 0,1)** jest równe 1,5

**ZAOKR.W.DÓŁ(0,234 ; 0,01)** jest równe 0,23

**ZAOKR.GÓRA** - zaokrągla liczbę w górę, dalej od zera.

Składnia: **ZAOKR.GÓRA(liczba ; liczba\_cyfr)**

**liczba**                                    liczba rzeczywista, która ma być zaokrąglona w górę

**liczba\_cyfr**                            liczba cyfr, do ilu liczba ma być zaokrąglona w górę

*Przykłady:*

**ZAOKR.GÓRA(3,2 ; 0)** jest równe **4**

**ZAOKR.GÓRA(31415,92654 ; -2)** jest równe **31500**

**ZAOKR.W.GÓRĘ** - wartość liczby zaokrąglona w górę, dalej od zera, do najbliższej wielokrotności cyfry znaczącej.

Składnia: **ZAOKR.W.GÓRĘ(liczba ; znaczenie)**

**liczba**                                    wartość liczbowa, którą należy zaokrąglić

**znaczenie**                            wielokrotność, do której należy zaokrąglić

**Uwagi:**

- Niezależnie od znaku liczby, zaokrąglenie następuje przy oddalaniu się od zera. Jeśli liczba jest dokładną wielokrotnością znaczenia, nie zachodzi żadne zaokrąglenie.
- Jeśli liczba i znaczenie mają różne znaki, **ZAOKR.W.GÓRĘ** podaje wartość błędu **#LICZBA!**.

Przykład:

**ZAOKR.W.GÓRĘ(2,5 ; 1)** jest równe 3

**ZAOKR.W.GÓRĘ(-2,5 ; -2)** jest równe -4

**ZAOKR.W.GÓRĘ(1,5 ; 0,1)** jest równe 1,5

**ZAOKR.W.GÓRĘ(0,234 ; 0,01)** jest równe 0,24

**KOMBINACJE** - liczba kombinacji dla danej liczby obiektów.

Składnia: **KOMBINACJE(liczba, liczba\_wybrana)**

**liczba**                                   liczbą obiektów

**liczba\_wybrana**                   liczbą obiektów w każdej kombinacji

**Uwagi:**

- Argumenty liczbowe obcina się do liczb całkowitych.
- Jeśli **liczba** < 0, **liczba\_wybrana** < 0, albo **liczba** < **liczba\_wybrana**, **KOMBINACJE** podaje wartość błędu **#LICZBA!**.

**PERMUTACJE** - liczba permutacji dla podanej liczby obiektów, które można wybrać z szerszej grupy obiektów.

Składnia: **PERMUTACJE**(liczba ; liczba\_wybrana)

**liczba**                               liczbą obiektów

**liczba\_wybrana**               liczba całkowita, która opisuje liczbę obiektów w każdej permutacji

**Uwagi:**

- Obydwa argumenty są w razie potrzeby zaokrąglane są sprowadzane do liczb całkowitych.
- Jeżeli **liczba** lub **liczba\_wybrana** jest argumentem nieliczbowym, funkcja daje w wyniku wartość błędu **#ARG!**.
- Jeżeli **liczba**  $\leq 0$  lub jeśli **liczba\_wybrana**  $< 0$ , funkcja daje w wyniku wartość błędu **#LICZBA!**.
- Jeżeli **liczba**  $<$  **liczba\_wybrana**, funkcja daje w wyniku wartość błędu **#LICZBA!**.

**LOS** - liczba losowa o równomiernym rozkładzie i wartości większej od 0 i mniejszej od 1.

Składnia: **LOS**( )

## Uwagi:

- Nową liczbę losową otrzymamy po każdym przeliczeniu arkusza.
- Aby wygenerować losową liczbę rzeczywistą z przedziału od **a** do **b**, należy użyć następującego wzoru:

$$\text{LOS()}*(b-a)+a$$

- Jeśli chce się wykorzystać funkcję **LOS** do wygenerowania liczby losowej bez zmiany liczby przy każdym przeliczaniu komórek, można wprowadzić **=LOS()** do paska formuły i wybrać przycisk **F9**, aby zmienić formułę na liczbę losową.

## RZYMSKIE - zamienia cyfry arabskie na rzymskie, jako tekst.

Składnia: **RZYMSKIE(liczba)**

**liczba** zapisana w systemie cyfr arabskich, którą należy zamienić

## Uwagi:

- Jeśli liczba jest ujemna, to podana zostanie wartość błędu **#ARG!**.
- Jeśli liczba jest większa od **3999**, to podana zostanie wartość błędu **#ARG!**.

**PI** - podaje liczbę 3,14159265358979 z dokładnością do 15 cyfr.

Składnia: **PI( )**

**RADIANY** - zamienia stopnie na radiany.

Składnia: **RADIANY(kąt)**

**kąt**

kąt podany w stopniach, który należy zamienić

**STOPNIE** - przekształca radiany na stopnie.

Składnia: **STOPNIE(kąt)**

**kąt**

miara kąta wyrażona w radianach

*Przykład:*

**STOPNIE(PI()) = 180**

**COS** - podaje wartość cosinusa kąta.

Składnia: **COS(liczba)**

**liczba**

kąt w radianach. Jeśli kąt jest w stopniach, należy pomnożyć go przez **PI()/180**, by uzyskać radiany.

*Przykład:*

**COS(60\*PI()/180)** jest równe **0,5**, cosinus 60 stopni

**SIN** - podaje sinus danego kąta.

Składnia: **SIN(liczba)**

**liczba** kąt w radianach. Jeśli kąt jest w stopniach, należy pomnożyć go przez **PI()/180**, by uzyskać radiany.

*Przykład:*

**SIN(30\*PI()/180)** jest równe **0,5**, sinus 30 stopni

**TAN** - podaje tangens kąta.

Składnia: **TAN(liczba)**

**liczba** kąt w radianach. Jeśli kąt jest w stopniach, należy pomnożyć go przez **PI()/180**, by uzyskać radiany.

*Przykład:*

**TAN(45\*PI()/180)** jest równe **1**

# Funkcje statystyczne

Statystyczne funkcje arkusza roboczego są przeznaczone do wykonywania analizy statystycznej na zakresach danych. Np.: funkcja może dostarczyć informacji statystycznych na temat linii prostej rysowanej w grupie wartości takich, jak nachylenie linii, przecięcie **y** albo o rzeczywistych punktach tworzących linię prostą.

**MAX** - zwraca maksymalną wartość z listy argumentów.

Składnia: **MAX(liczba1 ; liczba2 ; ...)**

**liczba1 ; liczba2 ;...** do 30 argumentów, dla których wyszukiwana jest wartość maksymalna

## Uwagi:

- Jeśli argument jest tablicą lub adresem, wykorzystywane są tylko liczby z tej tablicy lub adresu. Puste komórki, wartości logiczne, tekst lub wartości błędów w tabeli lub adresie są pomijane.
- Jeśli argumenty nie zawierają liczb, funkcja **MAX** wyświetla wartość **0**.



**MIN** - podaje wartość najmniejszej liczby w liście argumentów.

Składnia: **MIN**(liczba1 ; liczba2 ; ...)

**liczba1 ; liczba2 ;...** do 30 liczb, dla których należy znaleźć wartość minimalną

Jeśli argumenty nie zawierają liczb, **MIN** podaje wartość **0**.

**ŚREDNIA** - podaje średnią arytmetyczną argumentów.

Składnia: **ŚREDNIA**(liczba1 ; liczba2 ; ...)

**liczba1 ; liczba2 ;...** do 30 argumentów liczbowych, dla których należy wyznaczyć średnią

**Uwagi:**

- Argumenty powinny stanowić liczby, nazwy, tablice lub adresy komórek zawierających liczby.
- Jeśli argument w postaci tablicy lub adresu zawiera tekst, wartości logiczne lub puste komórki, wartości te są ignorowane, jednakże komórki z wartością zerową są uwzględniane.

**ŚREDNIA.GEOMETRYCZNA** - podaje średnią geometryczną tablicy lub zakresu danych dodatnich.

Składnia: **ŚREDNIA.GEOMETRYCZNA(liczba1 ; liczba2 ; ...)**

**liczba1 ; liczba2 ;...** do 30 argumentów, dla których chcemy obliczyć średnią. Można również stosować pojedynczą tablicę lub adres tablicy, zamiast argumentów rozdzielonych średnikami.

**Uwagi:**

- Jeżeli tablica lub adres argumentu zawiera tekst, wartości logiczne lub puste komórki, wówczas wartości te będą ignorowane, jednakże komórki z wartością zerową zostaną uwzględnione.
- Jeżeli jakaś dana  $\leq 0$ , ŚREDNIA.GEOMETRYCZNA daje w wyniku wartość błędu **#LICZBA!**

**MEDIANA** - podaje wartość mediany. (Liczba w środku zbioru liczb tzn. połowa liczb ma wartości większe niż mediana i połowa ma wartości mniejsze).

Składnia: **MEDIANA(liczba1 ; liczba2 ; ...)**

**liczba1 ; liczba2 ;...** do 30 liczb, dla których należy wyznaczyć medianę

Jeśli liczba danych liczbowych w zbiorze jest parzysta, to MEDIANA oblicza średnią dwóch liczb środkowych.

*Przykład:*

**MEDIANA(1; 2; 3; 4; 5; 6)** jest równe **3,5** (średnia z 3 i 4)

## **ILE.LICZB** - ilość liczb w liście argumentów.

Składnia: **ILE.LICZB(wartość1, wartość2,...)**

**wartość1,** do 30 argumentów, które mogą zawierać lub odwoływać się do różnych typów  
**wartość2,...** danych

### **Uwagi:**

- Zliczane są argumenty liczbowe, zerowe, wartości logiczne lub tekstowe, jednakże ignorowane są wartości błędów lub tekstowe, niemożliwe do przedstawienia w postaci liczbowej.
- Jeśli argument jest tablicą lub adresem, zlicza się jedynie liczby, ignorowane są puste komórki, wartości logiczne i tekstowe.

## **ILE.NIEPUSTYCH** - zlicza liczbę niepustych wartości w liście argumentów.

Składnia: **ILE.NIEPUSTYCH(wartość1, wartość2,...)**

**wartość1,** do 30 argumentów przedstawiających wartości, które należy zliczyć. Wartością jest dowolny  
**wartość2,...** typ informacji, włącznie z pustym tekstem (""), ale z wyłączeniem pustych komórek. Jeśli argument jest tablicą lub adresem, puste komórki nie będą brane pod uwagę

**LICZ.PUSTE** - zlicza puste komórki w wybranym zakresie.

Składnia: **LICZ.PUSTE(zakres)**

**Zakres**                    zakres, z którego mają być zliczane puste komórki  
Zliczane są także komórki z formułami podającymi w wyniku "" (pusty tekst). Nie są zliczane komórki z wartością równą zero.

**LICZ.JEŻELI** - zlicza liczbę komórek wewnątrz zakresu, które spełniają podane wymagania.

Składnia: **LICZ.JEŻELI(zakres ; kryteria)**

**zakres**                    zakres komórek, z którego mają być zliczane komórki

**kryteria**                    kryteria w postaci liczby, wyrażenia lub tekstu określające, które komórki będą obliczane

*Przykłady:*

Jeżeli **A3:A6** zawiera odpowiednio "jabłka", "pomarańcze", "brzoskwinie", "jabłka", to:

**LICZ.JEŻELI(A3:A6;"jabłka")** jest równe **2**

Jeżeli **B3:B6** zawiera odpowiednio **32, 54, 75, 86**.

**LICZ.JEŻELI(B3:B6;">55")** jest równe **2**

## Funkcje tekstowe

Za pomocą funkcji tekstowych można manipulować ciągami tekstowymi w formułach. Np.: można zmienić wielkość liter albo ustalić długość ciągu tekstowego. Można także łączyć lub wiązać datę z ciągiem tekstowym. Poniższa formuła jest przykładem sposobu użycia funkcji **DZIŚ** z funkcją **TEKST** do utworzenia komunikatu zawierającego bieżącą datę sformatowaną w postaci **dd-mmm-rr**.

= "Raport budżetu z dnia "&TEKST(DZIŚ(),"dd-mmm-rr")

**ZNAK** - podaje znak o podanym numerze kodu.

Składnia: **ZNAK(liczba)**

**liczba**

liczba od 1 do 255 określająca kod ASCII znaku

*Przykład:*

**ZNAK(65)** jest równe **A**

**KOD** - zwraca kod liczbowy pierwszego znaku w szeregu.

Składnia: **KOD(tekst)**

**tekst**

tekst, dla którego ma zostać wyświetlony kod pierwszego znaku

*Przykłady:*



**Z.WIELKIEJ.LITERY** - zmienia na wielką literę pierwszą literę w tekście następujące po znaku innym niż litera. Wszystkie inne litery zastępowane są literami małymi.

Składnia: **Z.WIELKIEJ.LITERY**(tekst)

**tekst** tekst lub formułą, której wynikiem jest tekst lub adres do komórki zawierającej tekst, który ma zostać częściowo przekształcony na tekst pisany wielkimi literami

*Przykład:*

**Z.WIELKIEJ.LITERY("to jest TYTUŁ")** jest równe **To Jest Tytuł**

**DŁ** - podaje liczbę znaków w łańcucha tekstowym.

Składnia: **DŁ**(tekst)

**tekst** tekst, którego długość należy znaleźć (spacje liczy się jako znaki)

**FRAGMENT.TEKSTU** - pobiera określoną liczbę znaków z łańcucha tekstowego, począwszy od podanej pozycji.

Składnia: **FRAGMENT.TEKSTU**(tekst ; liczba\_początkowa ; liczba\_znaków)

**tekst** łańcuch tekstowy, zawierającym znaki, które należy wydzielić

**liczba\_początkow** położenie pierwszego znaku, który należy wydzielić z tekstu (pierwszy znak w tekście ma  
**a** liczbę\_początkową 1)

- Jeśli **liczba\_początkowa** jest większa niż długość tekstu, **FRAGMENT.TEKSTU** podaje wartość "".
- Jeśli **liczba\_początkowa** jest mniejsza niż długość tekstu, ale **liczba\_początkowa + liczba\_znaków** przewyższają długość tekstu, wynikiem **FRAGMENT.TEKSTU** są znaki do końca tekstu.
- Jeśli **liczba\_początkowa** jest mniejsza niż 1, **FRAGMENT.TEKSTU** podaje wartość błędu **#ARG!**.

**liczba\_znaków**      ile znaków powinno zostać pobrane z tekstu. Gdy **liczba\_znaków < 0**, **FRAGMENT.TEKSTU** podaje wartość błędu **#ARG!**

*Przykłady:*

**FRAGMENT.TEKSTU("ABCDEFGHIJK" ; 1 ; 5)** jest równe **ABCDE**  
**FRAGMENT.TEKSTU("ABCDEFGHIJK" ; 7 ; 20)** jest równe **GHIJK**  
**FRAGMENT.TEKSTU("1234" ; 5 ; 5)** jest równe ""



**LEWY** - podaje pierwsze znaki w łańcuchu tekstowym.

Składnia: **LEWY**(tekst ; liczba\_znaków)

**tekst**                      łańcuch tekstowy zawierający znaki, które należy wydzielić

**liczba\_znaków**        ile znaków ma być wynikiem działania **LEWY**.

- **Liczba\_znaków** musi być większa lub równa 0.
- Jeśli **liczba\_znaków** jest większa niż długość tekstu, wynikiem jest cały tekst.
- Jeśli argument **liczba\_znaków** zostanie pominięty, jego wartość zostanie domyślnie przyjęta za **1**.

*Przykład:*

**LEWY("ABCDEFGHIJK" ; 4)** jest równe **ABCD**

Jeśli **A1** zawiera **ABCDEFGHIJK**, to:

**LEWY(A1)** jest równe **A**

**PRAWY** - podaje ostatnie (najbardziej z prawej strony) znaki w łańcuchu tekstowym.

Składnia: **PRAWY**(tekst ; liczba\_znaków)

**tekst**                      łańcuch tekstowy zawierający znaki, które należy wydzielić

**liczba\_znaków**           określa liczbę znaków do pobrania.

- **Liczba\_znaków** musi być większa lub równa 0.
- Jeśli **liczba\_znaków** jest większa niż długość tekstu wynikiem jest cały tekst.
- Jeśli argument **liczba\_znaków** zostanie pominięty, jego wartość zostanie domyślnie przyjęta za **1**.

*Przykłady:*

**PRAWY**("ABCDEFGHIJK" ; 3) jest równe **IJK**

**PRAWY**("ABCDEFGHIJK") jest równe **K**

**ZNAJDŹ** - odnajduje jeden łańcuch tekstowy wewnątrz innego łańcucha i podaje numer znaku, przy którym pojawia się po raz pierwszy szukany tekst .

Składnia: **ZNAJDŹ**(szukany\_tekst ; obejmujący\_tekst ; punkt\_startu)

**tekst** łańcuch tekstowy zawierający znaki, które należy wydzielić

**szukany\_tekst** tekst, który należy znaleźć

**obejmujący\_tekst** tekst, zawierający tekst, który należy znaleźć

- Jeśli **szukany\_tekst** jest "" (pusty), to funkcja **ZNAJDŹ** dopasowuje pierwszy znak w przeszukiwanym łańcuchu (znak oznaczony **punkt\_startu** lub **1**).
- **Szukany\_tekst** nie może zawierać żadnych znaków zastępczych.

**punkt\_startu** znak, od którego należy rozpocząć przeszukiwania (pierwszy znak w ma ner 1). Jeśli argument zostanie pominięty, to przyjmuje się, że jego wartość wynosi **1**

Jeśli **szukany\_tekst** nie występuje w **obejmujący\_tekst**, funkcja **ZNAJDŹ** podaje w wyniku wartość błędu **#ARG!**.

*Przykłady:*

**ZNAJDŹ**("c" ; "ABCDEFGHJKLM") jest równe **#ARG!**

**ZNAJDŹ**("C" ; "ABCDEFGHJKLM") jest równe **3**

**ZNAJDŹ("C" ; "ABCDEFGHIJK" ; 5)** jest równe **#ARG!**

**PODSTAW** - funkcja zastępuje argumentem nowym tekstem fragment tekstu w łańcuchu tekstowym.

Składnia: **PODSTAW**(tekst ; poprzedni\_tekst ; nowy\_tekst ; numer\_wystąpienia)

**tekst** tekst lub adres komórki zawierającej tekst, w którym mają być zastąpione znaki

**poprzedni\_tekst** tekst do zastąpienia

**nowy\_tekst** tekst, którym zostanie zastąpiony **poprzedni\_tekst**

**numer\_wystąpienia** określa, które wystąpienie argumentu **poprzedni\_tekst** zostanie zastąpione przez **nowy\_tekst**; jeśli wybierze się **numer\_wystąpienia**, to tylko to wystąpienie **poprzedni\_tekst** zostanie zastąpione; w innym przypadku każde pojawienie się **poprzedni\_tekst** jest zamieniane przez **nowy\_tekst**

Funkcję **PODSTAW** stosuje się, gdy trzeba zamienić wybrany tekst w łańcuchu tekstowym, funkcję **ZASTĄP** zaś, kiedy trzeba zastąpić dowolny tekst występujący w określonym miejscu łańcucha tekstowego.

*Przykłady:*

**PODSTAW("ABCABCABC"; "A"; "X")** jest równe **XBCXBCXBC**

**PODSTAW("ABCABCABC" ; "A" ; "X" ; 2)** jest równe **ABCXBCABC**

**POWT** - powtarza tekst określoną ilość razy.

Składnia: **POWT**(tekst ; liczba\_powtórzeń)

**tekst** tekst, który ma być powtarzany

**liczba\_powtórzeń** liczba dodatnia określająca ilość powtórzeń tekstu; jeśli argument ma wartość **0**, funkcja podaje w wyniku ""; wynik działania funkcji nie może być dłuższy niż **255** znaków

*Przykład:*

**POWT("\*-" ; 3)** jest równe **\*-\*-\***

**ZASTĄP** - zastępuje część łańcucha tekstowego innym łańcuchem tekstowym.

Składnia: **ZASTĄP**(poprzedni\_tekst ; numer\_znaku ; liczba\_znaków ; nowy\_tekst)

**poprzedni\_tekst** tekst, w którym chce się zastąpić niektóre znaki

**numer\_znaku** pozycja znaku w tekście określonym przez argument **poprzedni\_tekst**, od której chce się zastąpić tekst za pomocą **nowy\_tekst**

**liczba\_znaków** liczba znaków w **poprzedni\_tekst** do zastąpienia przez **nowy\_tekst**

**nowy\_tekst** tekst, który zastąpi znaki w **poprzedni\_tekst**

*Przykłady:*

**ZASTĄP("ABCDEFGHGIJK"; 1; 3; "\*\*\*")** jest równe **\*DEFGHIJK**

**ZASTĄP("ABCDEFGHGIJK " ; 3 ; ; "\*\*\*")** jest równe **AB\*DEFGHIJK**

**TEKST** - przekształca wartość w tekst w określonym formacie liczbowym.

Składnia: **TEKST(wartość ; format\_tekst)**

**wartość**                      wartość liczbowa, formułą obliczaną jako liczba lub adresem komórki zawierającej wartość liczbową

**format\_tekst**                format liczbowy w postaci tekstowej (nie może zawierać gwiazdki (\*) i nie może być formatem **Ogólny**)

Formatowanie komórki przy pomocy polecenia na karcie **Liczby** z pola dialogu **Formatuj komórki** zmienia tylko format, a nie wartość. Zastosowanie tej funkcji spowoduje przekształcenie wartości w sformatowany tekst, a wynik nie będzie już dłużej obliczany jako liczba.

*Przykłady:*

**TEKST(2,715 ; "0,00 zł")** jest równe **2,72 zł**

**TEKST("31.11.2002" ; "dd mmmm rrrr")** jest równe **31 listopad 2002**

## Funkcje daty i czasu

Funkcje daty i czasu pozwalają na pobieranie daty i czasu systemowego oraz wykonywanie wszelkiego typu operacji na datach i godzinach: obliczanie różnic, konwersję, itp.

**TERAZ** - podaje wartość bieżącej daty i czasu.

Składnia: **TERAZ**( )

**Uwagi:**

- numery kolejne pochodzą z zakresu od **1** do **65 380**, co odpowiada datom od **1 stycznia 1900** do **31 grudnia 2078**
- część całkowita liczby przedstawiają datę, a część ułamkowa - czas
- funkcja **TERAZ** zmienia się tylko, gdy przeliczamy arkusz lub wykonujemy makroinstrukcja zawierająca tę funkcję

**DZIŚ** - podaje aktualną datę.

Składnia: **DZIŚ**( )

Numer kolejny jest kodem daty-czasu wykorzystywanym przez do obliczeń daty i czasu.

**DATA** - podaje numer kolejny podanej daty.

Składnia: **DATA(rok ; miesiąc ; dzień)**

**rok**                      liczba z zakresu od **1900** do **2078**

**miesiąc**                miesiąc roku; jeśli miesiąc jest większy niż **12**, to liczbę tę dodaje się do pierwszego miesiąca podanego roku

**dzień**                    dzień miesiąca; jeśli dzień jest większy niż liczba dni w miesiącu, to liczbę tę dodaje się do pierwszego dnia miesiąca

**ROK** - podaje rok odpowiadający argumentowi (liczba całkowita od 1900 do 2078).

Składnia: **ROK(numer\_kolejny)**

**numer\_kolejny**            kod daty-czasu przy obliczaniu daty i czasu

**MIESIĄC** - podaje miesiąc odpowiadający numerowi kolejnemu (liczba całkowita od 1 do 12).

Składnia: **MIESIĄC(numer\_kolejny)**

**numer\_kolejny**            kod daty-czasu stosowanym w obliczeniach daty i czasu

**DZIEŃ.TYG** - podaje dzień tygodnia odpowiadający numerowi kolejnemu.

Dzień jest wyrażony jako liczba całkowita, zmieniająca się od **1** (*niedziela*) do **7** (*sobota*).

Składnia: **DZIEŃ.TYG(numer\_kolejny ; typ\_wyniku)**



**numer\_kolejny** kod daty-czasu używany do obliczeń daty i czasu. Można podać w postaci tekstu, jak np.: **11-Lis-2002** lub **11-11-02**, zamiast w postaci liczby

**Typ\_wyniku** jest liczbą, która określa typ obliczanej wartości

<b>Typ_wyniku:</b>	<b>Obliczona liczba:</b>
<b>1</b> lub <b>brak</b>	liczby od <b>1</b> ( <i>niedziela</i> ) do <b>7</b> ( <i>sobota</i> )
<b>2</b>	liczby od <b>1</b> ( <i>poniedziałek</i> ) do <b>7</b> ( <i>niedziela</i> )
<b>3</b>	liczby od <b>0</b> ( <i>poniedziałek</i> ) do <b>6</b> ( <i>niedziela</i> )

**CZAS** - podaje w wyniku numer kolejny podanego czasu.

Numer kolejny obliczany przez funkcję **CZAS** jest ułamkiem dziesiętnym w zakresie od **0** do **0,99999999**, reprezentującym czas od **0:00:00** do **23:59:59**.

Składnia: **CZAS(godzina ; minuta ; sekunda)**

**godzina** liczba od **0** do **23** reprezentującą godziny

**Minuta** liczba od **0** do **59** reprezentującą minuty

**Sekunda** liczba od **0** do **59** reprezentującą sekundy

**GODZINA** - podaje godzinę odpowiednio do podanej godziny (liczba całkowita od 0 do 23).

Składnia: **GODZINA(numer\_kolejny)**

**numer\_kolejny**                    kod daty-czasu stosowany w obliczeniach daty i czasu

**MINUTA** - podaje wartość minut odpowiadającą podanej godzinie (liczba całkowita od 0 do 59).

Składnia: **MINUTA(numer\_kolejny)**

**numer\_kolejny**                    kod daty-czasu stosowany w obliczeniach daty i czasu

**SEKUNDA** - podaje sekundy, którym przyporządkowany jest czas (liczba całkowita od 0 do 59).

Składnia: **SEKUNDA(numer\_kolejny)**

**numer\_kolejny**                    kod daty-czasu stosowany w obliczeniach daty i czasu

## Funkcje logiczne

Funkcje logiczne są przeznaczone do sprawdzania jednego lub wielu warunków. Np.: można użyć funkcji **JEŻELI** do ustalenia, czy warunek jest prawdą czy fałszem: Jedna wartość jest zwracana, gdy warunek jest spełniony, a inna wartość, gdy warunek nie jest spełniony.

**JEŻELI** - podaje w wyniku jedną wartość, jeśli argument **logiczna\_test** zostanie obliczony jako **PRAWDA** a inną, jeśli wartością tą będzie **FAŁSZ**. Funkcję **JEŻELI** wykorzystuje się w celu przeprowadzenia testu logicznego na wartościach i formułach. Wynik testu decyduje o wartości otrzymanej w wyniku funkcji **JEŻELI**.

Składnia: **JEŻELI(logiczna\_test ; wartość\_jeżeli\_prawda ; wartość\_jeżeli\_fałsz)**

**logiczna\_test** dowolna wartością lub wyrażeniem zwracające wartość **PRAWDA** lub **FAŁSZ**

**wartość\_jeżeli\_prawda** wartość wyniku, gdy wartość testu jest **PRAWDA**. W przypadku, gdy **wartość\_jeżeli\_prawda** zostanie pominięty, to wynikiem będzie **PRAWDA**

**wartość\_jeżeli\_fałsz** wartość wyniku, gdy wartość testu jest **FAŁSZ**. W przypadku, gdy **wartość\_jeżeli\_fałsz** zostanie pominięty, to wynikiem będzie **FAŁSZ**

Można zagnieździć do 7 funkcji **JEŻELI** stosując je jako argumenty **wartość\_jeżeli\_prawda** i **wartość\_jeżeli\_fałsz** w celu spełnienia bardziej złożonych warunków.

*Przykład:*

**JEŻELI(CZY.LICZBA(A1) ; JEŻELI (A1>=0 ; "Dodatnia" ; "Ujemna") ; JEŻELI(CZY.TEKST(A1) ; "To jest tekst" ; "Ani liczba, ani tekst" ) )**

**LUB** - zwraca wartość logiczną PRAWDA ,jeśli choć jeden argument ma wartość logiczną PRAWDA; jeśli wszystkie argumenty mają wartość logiczną FAŁSZ, funkcja zwraca wartość logiczną FAŁSZ.

Składnia: **LUB(wartość\_logiczna1 ; wartość\_logiczna2 ; ...)**

**wartość\_logiczna1 ;** warunki (do 30) poddane sprawdzeniu pod kątem wartości logicznej  
**wartość\_logiczna2 ;...**

#### Uwagi:

- Argumenty powinny być wartościami logicznymi lub adresami zawierającymi wartości logiczne.
- Jeśli którykolwiek z argumentów zawiera tekst, liczby lub puste komórki, wartości te są pomijane.
- Jeśli określony zakres nie zawiera wartości logicznych, funkcja **LUB** wyświetla wartość błędu **#ARG!**.

#### Przykłady

**LUB(1+1=1;2+2=5)** jest równe FAŁSZ

Sprawdzenie czy komórka **A1** zawiera pojedynczy znak **c** lub **s**:

**JEŻELI(LUB(A1="c";A1="s"),,)**

**ORAZ** - wynikiem funkcji jest PRAWDA, jeśli wszystkie jej argumenty mają wartość PRAWDA ; wynikiem funkcji jest FAŁSZ, jeśli co najmniej jeden z argumentów ma wartość FAŁSZ.

Składnia: **ORAZ(logiczny1 ; logiczny2 ; ...)**

**logiczny1 ; logiczny2 ; ...** do 30 warunków logicznych, które mogą przyjmować wartości logiczne **PRAWDA** lub **FAŁSZ**

#### Uwagi:

- Argumenty powinny być wartościami logicznymi lub adresami zawierającymi wartości logiczne.
- Pomija się wartości adresów zawierające tekst lub puste.
- Podaje wartość błędu **#ARG!**, jeśli wprowadzony zakres nie zawiera wartości logicznych.

#### Przykład:

Należy wyświetlić **B4**, o ile zawiera ona liczbę z przedziału **1** do **100**, a komunikat w przeciwnym przypadku.

Jeżeli **B4** zawiera **104**, to:

**JEŻELI(ORAZ(1>=B4;B4<=100);B4;"Wartość spoza zakresu".)** jest równe "Wartość spoza zakresu".

Jeśli **B4** zawiera **50**, to:

**JEŻELI(ORAZ(1>=B4;B4<=100);B4;"Wartość spoza zakresu")** jest równe 50

**NIE** - odwraca wartość swojego argumentu.

Składnia: **NIE(zm\_logiczna)**

**zm\_logiczna** wartość lub wyrażeniem, które musi przyjmować wartości **PRAWDA** lub **FAŁSZ**

*Przykład:*

**NIE(1+1=2)** jest równe FAŁSZ

Funkcje bazy danych

W Excel'u występuje kilkanaście funkcji używanych do analizy danych w listach lub w bazach danych. Każda z tych funkcji wymaga podania trzech argumentów: **baza danych**, **pole** i **kryteria**. Argumenty te podają adresy zakresów arkusza, wykorzystywanych przez funkcje bazy danych.

Składnia: **BDfunkcja(baza\_danych; pole; kryteria)**

gdzie:

- **baza\_danych**  
zakres komórek tworzących bazę danych. Baza danych jest listą powiązanych informacji, w którym wiersze są rekordami, a kolumny polami. Pierwszy wiersz listy zawiera etykiety poszczególnych kolumn. Adres można podawać jako zakres komórek lub jako nazwę odpowiadającą zakresowi zawierającemu listę.
- **pole**

wskazuje, którą kolumnę funkcja ma obejmować swoim działaniem. Kolumny danych muszą mieć etykietę identyfikacyjną w pierwszym wierszu. Argument pole można podać jako tekst, ujmując etykietę kolumny w cudzysłów lub jako numer odpowiadający pozycji kolumny na liście: 1 dla kolumny pierwszej, 2 dla drugiej itd.

- **kryteria**

adres zakresu komórek zawierających warunki funkcji. Funkcja zwraca tylko te informacje z listy, które spełniają warunki podane w zakresie kryteriów. Zakres kryteriów zawiera kopię etykiety kolumny, która ma zostać przez funkcję podsumowana. Adres kryteriów można wprowadzić jako zakres komórek (np. A1:F2) lub jako nazwę przypisaną do zakresu

## **Wskazówki**

- Możliwe jest użycie dowolnego zakresu jako argumentu kryterium, o ile tylko zawiera on co najmniej jedną etykietę kolumny i co najmniej jedną komórkę poniżej etykiety, służącą do określania warunku.
- Należy upewnić się, że zakres kryteriów nie zachodzi na zakres z listy.
- Aby przeprowadzić operację na całej kolumnie w bazie danych, należy wprowadzić pusty wiersz poniżej etykiet kolumn w zakresie zawierającym kryterium.

**BD.ILE.REKORDÓW** - zlicza komórki w kolumnie bazy danych, zawierające liczby, pasujące do podanych kryteriów. Argument pole jest opcjonalny. Jeśli argument pole został pominięty, funkcja BD.ILE.REKORDÓW zlicza wszystkie rekordy w bazie danych, które spełniają kryteria.

Składnia: **BD.ILE.REKORDÓW(baza\_danych ; pole ; kryteria)**

**BD.ILOCZYN** - mnoży wartości w kolumnie bazy danych, spełniające podane warunki.

Składnia: **BD.ILOCZYN(baza\_danych ; pole ; kryteria)**

**BD.MAX** - podaje wartość największej liczby w kolumnie bazy danych, spełniającej podane warunki.

Składnia: **BD.MAX(baza\_danych ; pole ; kryteria)**

**BD.MIN** - podaje wartość najmniejszej liczby w kolumnie bazy danych, spełniającej podane warunki.

Składnia: **BD.MIN(baza\_danych ; pole ; kryteria)**

**BD.SUMA** - dodaje do siebie liczby w kolumnie bazy danych, spełniające podane warunki.

Składnia: **BD.SUMA(baza\_danych ; pole ; kryteria)**



**BD.ŚREDNIA** - uśrednia wartości w kolumnie bazy danych, spełniające podane kryteria.

Składnia: **BD.ŚREDNIA(baza\_danych ; pole ; kryteria)**

## Funkcje finansowe

Funkcje finansowe służą do wykonywania często stosowanych obliczeń finansowych takich, jak ustalanie wysokości spłat pożyczki, przyszłej wartości lub bieżącej wartości inwestycji oraz wartości obligacji lub papierów wartościowych. Do najczęściej stosowanych argumentów funkcji finansowych należą:

- **Wartość przyszła (fv)** - wartość inwestycji lub pożyczki po spłaceniu wszystkich wpłat.
- **Liczba okresów (nper)** - całkowita liczba wpłat lub okresów inwestycji.
- **Wpłata (pmt)** - kwota wpłacana okresowo na poczet inwestycji lub pożyczki.
- **Wartość obecna (pv)** - wartość inwestycji lub pożyczki na początku okresu inwestycji. Np.: obecna wartość pożyczki jest podstawową kwotą, która jest pożyczana.
- **Oprocentowanie (rate)** - wysokość oprocentowania pożyczki lub inwestycji.

- **Typ (type)** - odstępy czasu, w jakich są dokonywane wpłaty w czasie okresu spłat takie, jak początek lub koniec miesiąca.

## Funkcje informacyjne

Informacyjne funkcje są przeznaczone do ustalania typu danych przechowywanych w komórce. Do funkcji informacyjnych należą funkcje stanowiące grupę funkcji znanych jako funkcje **CZY**. Funkcje te zwracają wartość **PRAWDA**, jeśli komórka spełnia warunek. Jeśli np. komórka zawiera wartość parzystą, funkcja arkusza roboczego **ISEVEN** zwraca wartość **PRAWDA**. Jeśli trzeba ustalić, czy w zakresie komórek są komórki puste, można zastosować funkcję arkusza roboczego **LICZ.PUSTE** do policzenia liczby pustych komórek w zakresie albo można zastosować funkcję **CZY.PUSTA** w celu ustalenia, czy w zakresie jest przynajmniej jedna pusta komórka.

**Funkcje CZY**: grupa 9 funkcji wykorzystywanych do testowania rodzaju wartości lub adresu. Każda z tych funkcji sprawdza rodzaj wartości i zwraca wartość **PRAWDA** lub **FAŁSZ** w zależności od wyniku. Funkcje **CZY** są przydatne w formułach do testowania wyniku obliczeń. W połączeniu z funkcją **JEŻELI**, pozwalają na wyszukiwanie błędów w formułach.

Składnia: **CZY.funkcja(wartość)**

**wartość** wartość poddawaną sprawdzeniu. Wartość może nie istnieć (pusta komórka), może zawierać poddawany testowi błąd, wartość logiczną, tekst, liczbę, adres lub nazwę odnoszącą się do którejkolwiek z powyższych wartości.

<b>Funkcja:</b>	<b>Wyświetla wartość logiczną PRAWDA jeśli:</b>
<b>CZY.PUSTA</b>	wartość odnosi się do pustej komórki
<b>CZY.BŁ</b>	wartość odnosi się do każdej wartości błędnej z wyjątkiem #N/D!
<b>CZY.BŁĄD</b>	wartość odnosi się do każdej wartości błędnej (#N/D!, #ARG!, #ADR!, #DZIEL/0!, #LICZBA!, #NAZWA? lub #ZERO!)
<b>CZY.LOGICZNA</b>	wartość odnosi się do wartości logicznej
<b>CZY.BRAK</b>	wartość odnosi się do wartości błędnej #N/D
<b>CZY.NIE.TEKST</b>	wartość odnosi się do każdej zawartości komórki innej niż tekst (również dla pustej komórki)
<b>CZY.LICZBA</b>	wartość odnosi się do liczby
<b>CZY.ADR</b>	wartość odnosi się do adresu
<b>CZY.TEKST</b>	wartość odnosi się do tekstu

*Przykład:*

Jeżeli dla zakresu A1:A4 chcemy obliczyć średnią, ale nie mamy pewności czy komórki zawierają liczby (gdy zakres nie zawiera żadnych liczb, formuła ŚREDNIA(...) wyświetla wartość błędu #DZIEL/0!):

**JEŻELI(CZY.BŁĄD(ŚREDNIA(A1:A4)));"Brak Liczb";ŚREDNIA(A1:A4))**

## Funkcje wyszukiwania i adresu

Szukając wartości na listach lub w tabelach albo odwołania do komórki można stosować funkcje wyszukiwania i odwołania. Aby np. znaleźć wartość w tabeli przez dopasowanie wartości z pierwszej kolumny tabeli, należy zastosować funkcję **WYSZUKAJ.PIONOWO**. Aby ustalić położenie wartości na liście, należy zastosować funkcję **PODAJ.POZYCJĘ**.

**NR.KOLUMNY** - podaje w wyniku numer kolumny przy danym adresie

Składnia: **NR.KOLUMNY(adres)**

**adres**                      komórka, której numer kolumny należy podać.

Jeśli argument adres zostanie pominięty, to zakłada się, że jest adresem komórki, w której pojawia się funkcja NR.KOLUMNY.

*Przykłady*

**NR.KOLUMNY(A3)** jest równe 1

Jeśli NR.KOLUMNY zostanie wprowadzona do C5, wtedy:

**NR.KOLUMNY()** jest równe 3

## WIERSZ - podaje numer wiersza adresu

Składnia: **WIERSZ(adres)**

**adres**

komórka, której numer kolumny należy podać.

komórka, dla której należy określić numer wiersza.

Jeśli argument adres zostanie pominięty, to zakłada się, że jest adresem komórki, w której pojawia się funkcja WIERSZ.

### *Przykłady*

**WIERSZ(A3)** jest równe 3

Jeśli WIERSZ wprowadzone jest w C5, wtedy:

**WIERSZ()** jest równe 5

### Funkcje definiowane przez użytkownika

Stosując szczególnie złożone obliczenia w wielu formułach lub obliczenia wymagające wielu formuł ponieważ istniejące funkcje arkusza roboczego są niewystarczające, można utworzyć funkcje niestandardowe. Funkcje te znane jako funkcje definiowane przez użytkownika są tworzone za pomocą języka **Visual Basic for Application**.