

## IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

## KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

## TWÓJ KOSZYK

DODAJ DO KOSZYKA

## CENNIK I INFORMACJE

ZAMÓW INFORMACJE  
O NOWOŚCIACH

ZAMÓW CENNIK

## CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

# VBA dla Excela 2007 PL. XXX praktyczne przykłady

Autor: Piotr Czarny  
ISBN: 83-246-1218-1  
Format: B5, stron: 224  
[Przykłady na ftp: 732 kB](#)



### Zautomatyzuj działanie arkusza kalkulacyjnego

- Poznaj język Visual Basic for Applications i zasady programowania obiektowego
- Zaprojektuj interfejsy użytkownika dla aplikacji
- Zoptymalizuj działanie programu w VBA

Visual Basic for Applications, w skrócie VBA, to język programowania wykorzystywany do tworzenia programów wykonywanych wewnątrz innych aplikacji i korzystających z ich funkcji. Najprostszym programem w VBA jest makropolecenie – zarejestrowany przez użytkownika ciąg operacji, który można potem wielokrotnie powtórzyć. Makropolecenia mogą znacznie przyspieszyć i usprawnić Twoją pracę.

Oczywiście na tym nie kończą się możliwości VBA. Wiele aplikacji w tym języku powstaje na potrzeby użytkowników arkusza kalkulacyjnego Excel. Za pomocą VBA tworzone są wyspecjalizowane narzędzia umożliwiające wprowadzanie danych, ich analizę, przetwarzanie i prezentację bez potrzeby żmudnego poszukiwania odpowiedniej funkcji w menu.

Książka „VBA dla Excela 2007 PL. 88 praktycznych przykładów” to podręcznik opisujący zasady automatyzowania i usprawniania pracy z tym arkuszem kalkulacyjnym za pomocą aplikacji VBA. Dzięki lekturze tego podręcznika poznasz język VBA i reguły programowania obiektowego. Nauczysz się korzystać z edytora VBA dostępnego w pakiecie Microsoft Office, pisać wydajny kod źródłowy, dzielić programy na moduły i projektować interfejsy użytkownika. Dowiesz się, jak wyszukiwać i usuwać błędy z aplikacji oraz rozwiązywać najczęściej spotykane problemy z działaniem programów.

- Rejestrowanie i wykorzystywanie makropoleceń
- VBA i programowanie obiektowe
- Edytor VBA
- Zmienne i typy danych
- Komunikacja z użytkownikiem aplikacji
- Przypisywanie wartości komórkom
- Operacje na arkuszu kalkulacyjnym
- Podział programu na moduły funkcjonalne
- Usuwanie błędów z kodu

**Poznaj ogrom możliwości VBA i napisz własne aplikacje dla Excela**



# Spis treści

<b>Rozdział 1. Co to jest VBA .....</b>	<b>7</b>
Basic .....	7
Visual Basic .....	8
Visual Basic for Applications .....	8
Czy VBA jest przeżytkiem .....	9
VBA i Office 2007 .....	10
Zmieniony wygląd okna programu .....	10
Makropolecenia w Office 2007 .....	10
Bezpieczeństwo w Office 2007 .....	11
VBA i Macintosh .....	11
Jaką rolę może pełnić VBA .....	11
Jak napisać najprostszy program w VBA .....	12
<b>Rozdział 2. Makropolecenia .....</b>	<b>13</b>
Office 2003 i 2007 .....	14
Office 2003 .....	14
Office 2007 .....	22
<b>Rozdział 3. Jak uruchomić edytor Visual Basic .....</b>	<b>31</b>
Czy makropolecenia są groźne .....	35
<b>Rozdział 4. Okno edytora Visual Basic .....</b>	<b>39</b>
Okno Project .....	40
Okno Properties .....	40
Okno Code .....	40
Pasek menu .....	42
Pasek narzędziowy .....	43
Pierwszy program .....	43
Strzelanie z armaty do komara .....	52
<b>Rozdział 5. Zmienne .....</b>	<b>53</b>
Nazwy zmiennych w VBA .....	53
Pułapki systemu komunikatów .....	56
Typy danych .....	56
Zasięg deklaracji .....	60
Zmienne lokalne .....	60
Zmienne obowiązujące wewnątrz całego modułu .....	61
Zmienne globalne .....	61

Deklaracja typu i instrukcja przypisania .....	62
Komórka arkusza jako zmienna .....	65
Tekst jako wartość zmiennej .....	66
<b>Rozdział 6. Komunikacja z użytkownikiem .....</b>	<b>67</b>
Wprowadzanie danych .....	67
Wyświetlanie komunikatów .....	70
Poprawność wprowadzanych danych .....	79
<b>Rozdział 7. Korzystanie z obiektów .....</b>	<b>85</b>
Metody .....	85
Zaznaczanie komórki .....	85
Usuwanie formatowania .....	87
Właściwości .....	89
Przypisanie wartości komórce .....	89
Kopiowanie zawartości komórek .....	90
Nadawanie komórce koloru z użyciem nazwy koloru .....	92
Nadawanie komórce koloru z użyciem kodu koloru .....	94
Przesuwanie wyboru komórki .....	97
<b>Rozdział 8. Instrukcje warunkowe .....</b>	<b>99</b>
Porównywanie .....	100
Sterowanie wykonywaniem procedur .....	100
Skok do etykiety .....	100
Podejmowanie decyzji .....	102
Wybór jednej z trzech lub więcej opcji .....	104
Wykonanie grupy instrukcji określoną liczbę razy .....	106
Pętle zagnieżdżone .....	108
Wykonywanie pętli, gdy warunek jest spełniony .....	110
Wykonywanie pętli, gdy warunek nie jest spełniony .....	111
<b>Rozdział 9. Elementy sterujące arkusza .....</b>	<b>115</b>
Pole listy .....	115
Pole kombi .....	123
Pasek Toolbox i elementy sterujące arkusza .....	125
Właściwości .....	127
<b>Rozdział 10. Zdarzenia .....</b>	<b>131</b>
Lista zdarzeń dla skoroszytu .....	132
Lista zdarzeń dla arkusza .....	137
Lista zdarzeń dla aplikacji .....	139
Komunikacja z programem .....	141
Lista zdarzeń dla formularzy .....	144
<b>Rozdział 11. Metody i właściwości dla zakresu .....</b>	<b>147</b>
Kopiowanie zakresu komórek .....	147
Sortowanie zakresu komórek .....	149
Filtrowanie zakresu komórek .....	152
Wyszukiwanie informacji .....	155
<b>Rozdział 12. Podprogramy .....</b>	<b>159</b>
Śledzenie pracy programu .....	164
Procedury zagnieżdżone .....	166
Procedury zapętłone .....	168

---

<b>Rozdział 13. Ściągawka z VBA .....</b>	<b>171</b>
Metody .....	171
Funkcje .....	178
Instrukcje .....	188
Operatory .....	192
Operatory arytmetyczne .....	192
Operatory porównywania .....	194
Operator konkatencji .....	196
Operatory logiczne .....	197
<b>Rozdział 14. FAQ .....</b>	<b>199</b>
Co zrobić, aby makropolecenia były dostępne we wszystkich arkuszach? .....	199
Czy w VBA istnieje operator kontynuacji wiersza? .....	201
Co oznaczają komunikaty błędów VBA? .....	203
Jak wyświetlić listę metod i właściwości obiektu? .....	210
Jak wyświetlić komunikat na pasku stanu? .....	211
<b>Zakończenie .....</b>	<b>213</b>
<b>Skorowidz .....</b>	<b>215</b>

## Rozdział 7.

# Korzystanie z obiektów

Obiektem jest element składowy aplikacji, np. komórka, zakres komórek.

Obiekty mają właściwości. Właściwość jest cechą obiektu. Właściwościami obiektu Range są: zawartość komórki, styl czcionki itp.

Metoda jest czynnością, np.: uaktywnienie komórki lub zakresu komórek, wyczyszczenie komórki lub zakresu komórek.

## Metody

Metoda jest sposobem postępowania, który prowadzi do określonego rezultatu. W przykładach 35. i 36. pokazano, jaką rolę w poleceniu pełnią metody.

## Zaznaczanie komórki

Do zaznaczania komórek można wykorzystać metodę `Select`.

### Przykład 35.

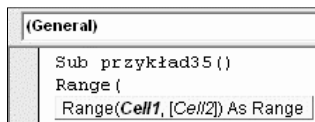
Napisz program, który będzie zaznaczał komórkę B2.

1. Uruchom arkusz kalkulacyjny Excel.
2. Naciśnij klawisze `Alt+F11`. Spowoduje to załadowanie edytora Visual Basic.
3. Naciśnij klawisz `F7`. Spowoduje to wyświetlenie okna `Code`.
4. Wpisz instrukcję `Sub przykład35()`.
5. Naciśnij klawisz `Enter`.
6. Edytor dopisał linię `End Sub`.
7. Umieść znak wstawiania na końcu pierwszej linii.

8. Naciśnij klawisz *Enter*.
9. Podaj nazwę obiektu. Wpisz `Range` (. W oknie edytora pojawiła się podpowiedź (rysunek 7.1).

**Rysunek 7.1.**

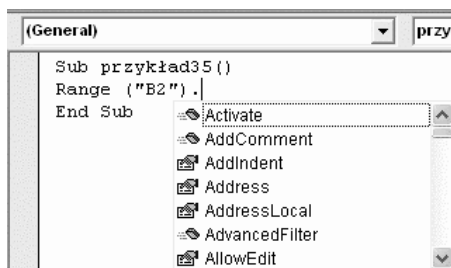
*Edytor podpowiada, jak można zdefiniować zakres*



10. Wpisz `"B2")`..
11. W oknie edytora pojawiła się podpowiedź (rysunek 7.2).

**Rysunek 7.2.**

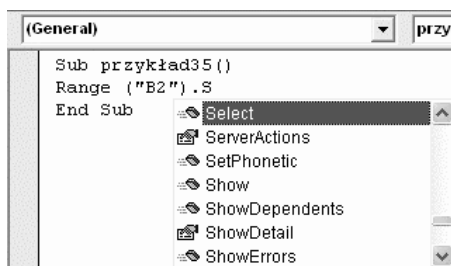
*Lista metod i właściwości*



12. Metody i właściwości uporządkowane są alfabetycznie. Zaznaczenie obiektu powoduje komenda `Select`. Wpisz literę `S`.
13. Wyświetlona została lista metod i właściwości o nazwach rozpoczynających się od litery `S`. Zaznacz metodę `Select` (rysunek 7.3).

**Rysunek 7.3.**

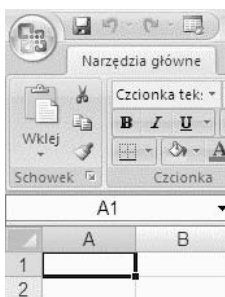
*Lista metod i właściwości o nazwach rozpoczynających się od litery S*



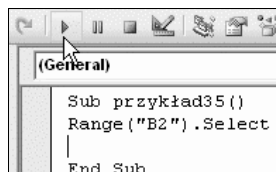
14. Potwierdź wybór naciśnięciem klawisza *Enter*.
15. Do programu wstawiona została metoda `Select`.
16. Wyświetl okno arkusza Excel (rysunek 7.4). Domyślnie wybrana jest komórka `A1`.
17. Wyświetl okno edytora.
18. Uruchom program kliknięciem ikony (rysunek 7.5).
19. Program został wykonany.
20. Wyświetl okno arkusza Excel (rysunek 7.6). Wybrana jest komórka `B2`.

**Rysunek 7.4.**

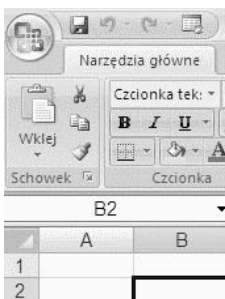
Arkusz z wybraną komórką A1

**Rysunek 7.5.**

Kliknięcie ikony spowoduje uruchomienie programu

**Rysunek 7.6.**

Arkusz z wybraną komórką B2



## Usuwanie formatowania

Usunięcie formatowania z komórek wymaga zaznaczenia zakresu komórek i posłużenia się metodą `Clear`.

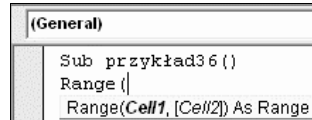
### Przykład 36.

Napisz program, który będzie czyścił i usuwał formatowanie komórek w zakresie od A1 do C3.

1. Uruchom arkusz kalkulacyjny Excel.
2. Naciśnij klawisze: `Alt+F11`. Spowoduje to załadowanie edytora Visual Basic.
3. Naciśnij klawisz `F7`. Spowoduje to wyświetlenie okna `Code`.
4. Wpisz instrukcję `Sub przyklad36()`.
5. Naciśnij klawisz `Enter`.
6. Edytor dopisał linię `End Sub`.
7. Umieść znak wstawiania na końcu pierwszej linii.

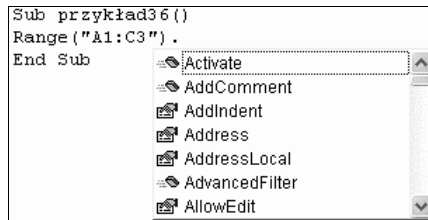
8. Naciśnij klawisz *Enter*.
9. Podaj nazwę obiektu. Wpisz `Range(`. W oknie edytora pojawiła się podpowieź (rysunek 7.7).

**Rysunek 7.7.**  
Operować można na pojedynczych komórkach lub ich zakresie



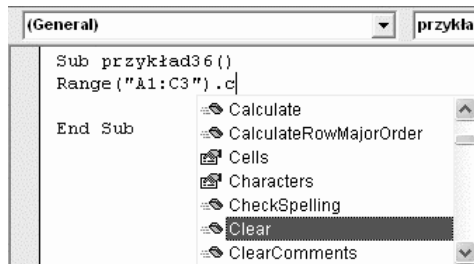
10. Wpisz `"A1:C3")`.
11. W oknie edytora pojawiła się podpowieź (rysunek 7.8).

**Rysunek 7.8.**  
Lista metod i właściwości



12. Metody i właściwości uporządkowane są alfabetycznie. Wyczyszczenie komórek zakresu powoduje polecenie `Clear`. Wpisz literę `C`.
13. Wyświetlona została lista metod i właściwości o nazwach rozpoczynających się od litery `C`. Zaznacz metodę `Clear` (rysunek 7.9).

**Rysunek 7.9.**  
Lista metod i właściwości o nazwach rozpoczynających się od litery `C`



14. Potwierdź wybór naciśnięciem klawisza *Enter*.
15. Do programu wstawiona została metoda `Clear`.
16. Wyświetl okno arkusza Excel. Domyślnie wszystkie komórki są puste. Wpisz dane do komórek (rysunek 7.10).

**Rysunek 7.10.**  
Arkusz z danymi

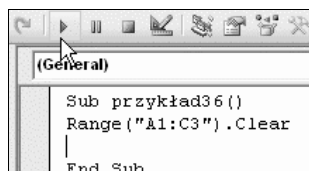
	A	B	C	D	E
1	a	a	a	a	
2	a	a	a	a	
3	a	a	a	a	
4	a	a	a	a	
5					



17. Wyświetl okno edytora.
18. Uruchom program kliknięciem ikony (rysunek 7.11).

**Rysunek 7.11.**

Kliknięcie ikony spowoduje uruchomienie programu



19. Program został wykonany.
20. Wyświetl okno arkusza Excel (rysunek 7.12). Komórki z zakresu A1:C3 zostały wyczyszczone.

**Rysunek 7.12.**

Komórki z zakresu A1:C3 zostały wyczyszczone

	A	B	C	D	E
1				a	
2				a	
3				a	
4	a	a	a	a	
5					

## Właściwości

Właściwością jest cecha obiektu. Może nią być np.: liczba wpisana w komórkę, kolor komórki itp.

## Przypisanie wartości komórce

Wartość, która jest przypisana komórce, zależy od właściwości Value.

**Przykład 37.**

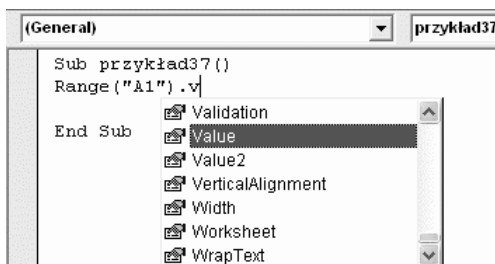
Napisz program, który będzie wpisywał liczbę 1 do komórki A1.

1. Uruchom arkusz kalkulacyjny Excel.
2. Naciśnij klawisze: *Alt+F11*. Spowoduje to załadowanie edytora Visual Basic.
3. Naciśnij klawisz *F7*. Spowoduje to wyświetlenie okna *Code*.
4. Wpisz instrukcję `Sub przykład37()`.
5. Naciśnij klawisz *Enter*.
6. Edytor dopisał linię `End Sub`.
7. Umieść znak wstawiania na końcu pierwszej linii.

8. Naciśnij klawisz *Enter*.
9. Podaj nazwę obiektu. Wpisz `Range(`. W oknie edytora pojawiła się podpowiedź.
10. Wpisz `"A1")`.
11. W oknie edytora pojawiła się podpowiedź.
12. Metody i właściwości uporządkowane są alfabetycznie. Przypisanie wartości komórce powoduje polecenie `Value`. Wpisz literę `v`.
13. Wyświetlona została lista metod i właściwości o nazwach rozpoczynających się od litery `V`. Zaznacz metodę `Value` (rysunek 7.13).

**Rysunek 7.13.**

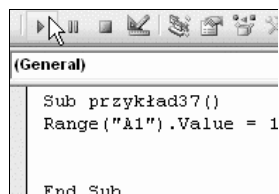
Zaznaczona  
właściwość



14. Potwierdź wybór naciśnięciem klawisza *Enter*.
15. Za wartością `Value` wpisz `=1`.
16. Uruchom program kliknięciem ikony *Run Sub/UserForm* (rysunek 7.14).

**Rysunek 7.14.**

Program można  
uruchomić  
kliknięciem ikony



17. Wyświetl arkusz Excel.
18. W komórce `A1` wpisana została wartość (rysunek 7.15).

**Rysunek 7.15.**

W komórce `A1`  
wpisana została  
wartość `1`

		A1		
		A	B	C
1		1		

## Kopiowanie zawartości komórek

Kopiowanie zawartości komórek polega na przypisaniu właściwości `Value` obszaru z którego kopiowane są dane, właściwości `Value` obszaru, do którego dane mają zostać skopiowane. Innymi słowy, kopiowana jest właściwość `Value`.

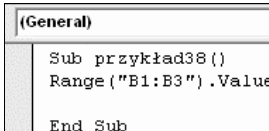
**Przykład 38.**

Napisz program, który będzie kopiował zawartość obszaru *A1:A3* do obszaru *B1:B3*.

1. Uruchom arkusz kalkulacyjny Excel.
2. Naciśnij klawisze *Alt+F11*. Spowoduje to załadowanie edytora Visual Basic.
3. Naciśnij klawisz *F7*. Spowoduje to wyświetlenie okna *Code*.
4. Wpisz instrukcję `Sub przykład38()`.
5. Naciśnij klawisz *Enter*.
6. Edytor dopisał linię `End Sub`.
7. Umieść znak wstawiania na końcu pierwszej linii.
8. Naciśnij klawisz *Enter*.
9. Podaj nazwę obiektu. Wpisz `Range(`. W oknie edytora pojawiła się podpowiedź.
10. Wpisz `"B1:B3")`..
11. W oknie edytora pojawiła się podpowiedź.
12. Metody i właściwości uporządkowane są alfabetycznie. Przypisanie wartości komórce powoduje polecenie `Value`. Wpisz literę *v*.
13. Wyświetlona została lista metod i właściwości o nazwach rozpoczynających się od litery *V*. Zaznacz metodę `Value`.
14. Potwierdź wybór naciśnięciem klawisza *Enter*.
15. Umieść znak wstawiania za metodą `Value` (rysunek 7.16).

**Rysunek 7.16.**

Zakresowi komórek  
od *B1* do *B3*  
przypisana zostanie  
wartość



```

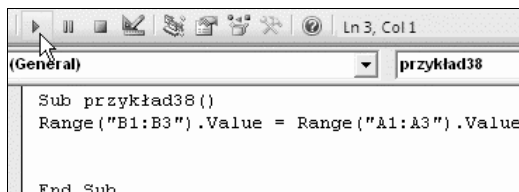
(General)
Sub przykład38()
Range("B1:B3").Value
End Sub

```

16. Wpisz z klawiatury znak równości.
17. Podaj nazwę obiektu, którego wartość zostanie przypisana obszarowi od *B1* do *B3*. Wpisz `Range(`. W oknie edytora pojawiła się podpowiedź.
18. Wpisz `"A1:A3")`..
19. W oknie edytora pojawiła się podpowiedź.
20. Metody i właściwości uporządkowane są alfabetycznie. Przypisanie wartości komórce powoduje polecenie `Value`. Wpisz literę *v*.
21. Wyświetlona została lista metod i właściwości o nazwach rozpoczynających się od litery *V*. Zaznacz metodę `Value`.

22. Potwierdź wybór naciśnięciem klawisza *Enter*.
23. Wyświetl arkusz Excel.
24. W komórki od *A1* do *A3* wpisz odpowiednio: 1, 2, 3.
25. Wyświetl okno edytora Visual Basic.
26. Uruchom program kliknięciem ikony *Run Sub/UserForm* (rysunek 7.17).

**Rysunek 7.17.**  
Program można uruchomić kliknięciem ikony



27. Wyświetl arkusz Excel.
28. W komórkach od *B1* do *B3* skopiowana została zawartość komórek od *A1* do *A3* (rysunek 7.18).

**Rysunek 7.18.**  
W komórkach *B1:B3* wpisana została zawartość komórek *A1:A3*

	A	B	C
1	1	1	
2	2	2	
3	3	3	

## Nadawanie komórce koloru z użyciem nazwy koloru

Aby nadać komórce kolor, wykorzystując nazwę koloru, należy posłużyć się właściwością `Interior.Color`.

### Przykład 39.

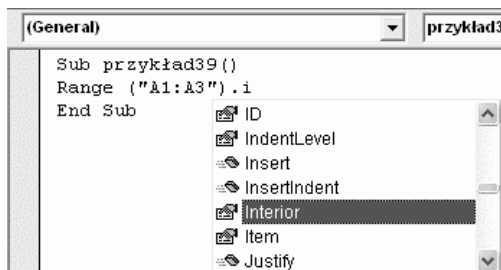
Napisz program, który będzie nadawał komórkom z obszaru *A1:A3* kolor żółty.

1. Uruchom arkusz kalkulacyjny Excel.
2. Naciśnij klawisze: *Alt+F11*. Spowoduje to załadowanie edytora Visual Basic.
3. Naciśnij klawisz *F7*. Spowoduje to wyświetlenie okna *Code*.
4. Wpisz instrukcję `Sub przykład39()`.
5. Naciśnij klawisz *Enter*.
6. Edytor dopisał linię `End Sub`.
7. Umieść znak wstawiania na końcu pierwszej linii.
8. Naciśnij klawisz *Enter*.

9. Podaj nazwę obiektu. Wpisz Range(. W oknie edytora pojawiła się podpowiedź.
10. Wpisz "A1:A3")..
11. W oknie edytora pojawiła się podpowiedź.
12. Za zmianę domyślnego koloru komórki odpowiada właściwość Interior. Wpisz z klawiatury literę i.
13. Zaznacz właściwość Interior (rysunek 7.19).

**Rysunek 7.19.**

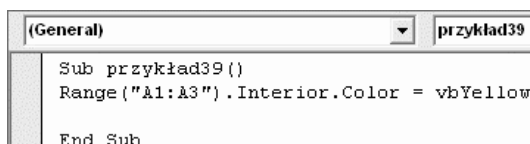
*Fragment programu z otwartym menu podręcznym*



14. Umieść kursor za właściwością Interior.
15. Korzystając z menu, które zostanie wyświetlone po wpisaniu kropki, wstaw do programu właściwość Color.
16. Przypisz właściwości Interior wartość vbYellow (rysunek 7.20).

**Rysunek 7.20.**

*Gotowy kod programu*



17. Uruchom program.
18. Wyświetl okno arkusza Excel.
19. Kolor komórek od A1 do A3 został zmieniony (rysunek 7.21).

**Rysunek 7.21.**

*Komórki, których kolor został zmieniony*

	A1	
	A	B
1		
2		
3		
4		



**Uwaga**

W tabeli 7.1 zebrano stałe, których przypisanie właściwości Interior powoduje wyświetlenie tła komórki w określonym kolorze.

**Tabela 7.1.** Stałe odpowiadające najczęściej używanym kolorom

Stała	Kolor
vbBlack	czarny
vbRed	czerwony
vbGreen	zielony
vbYellow	żółty
vbBlue	niebieski
vbMagenta	fioletowy
vbCyan	zielononiebieski
vbWhite	biały

## Nadawanie komórce koloru z użyciem kodu koloru

Aby nadać komórce kolor, wykorzystując kod koloru, należy posłużyć się właściwością `ColorIndex`.

### Przykład 40.

Napisz program, który będzie nadawał komórkom z obszaru `A1:A56` kolory z palety barw dostępnej za pośrednictwem właściwości `ColorIndex`.

1. Uruchom arkusz kalkulacyjny Excel.
2. Naciśnij klawisze: `Alt+F11`. Spowoduje to załadowanie edytora Visual Basic.
3. Naciśnij klawisz `F7`. Spowoduje to wyświetlenie okna `Code`.
4. Wpisz instrukcję `Sub przykład40()`.
5. Naciśnij klawisz `Enter`.
6. Edytor dopisał linię `End Sub`.
7. Umieść znak wstawiania na końcu pierwszej linii.
8. Naciśnij klawisz `Enter`.
9. Podaj nazwę obiektu. Wpisz `Range(`. W oknie edytora pojawiła się podpowiedź.
10. Wpisz `"A1")`.
11. W oknie edytora pojawiła się podpowiedź.
12. Za zmianę domyślnego koloru komórki odpowiada właściwość `Interior`. Wpisz z klawiatury literę `i`.
13. Zaznacz właściwość `Interior` (rysunek 7.19).
14. Umieść kursor za właściwością `Interior`.
15. Korzystając z menu, które zostanie wyświetlone po wpisaniu kropki, wstaw do programu właściwość `ColorIndex`.

16. Przypisz właściwości Interior wartość 1 (rysunek 7.22).

**Rysunek 7.22.**

*Fragment programu*

```
Sub przykład40()
Range("A1").Interior.ColorIndex = 1
End Sub
```

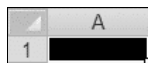
17. Uruchom program.

18. Wyświetl okno arkusza Excel.

19. Kolor komórki A1 został zmieniony na czarny (rysunek 7.23).

**Rysunek 7.23.**

*Kolor komórki A1 został zmieniony na czarny*



20. Wyświetl okno edytora Visual Basic.

21. Dopisz do programu linie `Range("An").Interior.ColorIndex = n`, gdzie  $n = 1, 2, 3, \dots, 56$ :

```
Sub przykład40()
Range("A1").Interior.ColorIndex = 1
Range("A2").Interior.ColorIndex = 2
Range("A3").Interior.ColorIndex = 3
Range("A4").Interior.ColorIndex = 4
Range("A5").Interior.ColorIndex = 5
Range("A6").Interior.ColorIndex = 6
Range("A7").Interior.ColorIndex = 7
Range("A8").Interior.ColorIndex = 8
Range("A9").Interior.ColorIndex = 9
Range("A10").Interior.ColorIndex = 10
Range("A11").Interior.ColorIndex = 11
Range("A12").Interior.ColorIndex = 12
Range("A13").Interior.ColorIndex = 13
Range("A14").Interior.ColorIndex = 14
Range("A15").Interior.ColorIndex = 15
Range("A16").Interior.ColorIndex = 16
Range("A17").Interior.ColorIndex = 17
Range("A18").Interior.ColorIndex = 18
Range("A19").Interior.ColorIndex = 19
Range("A20").Interior.ColorIndex = 20
Range("A21").Interior.ColorIndex = 21
Range("A22").Interior.ColorIndex = 22
Range("A23").Interior.ColorIndex = 23
Range("A24").Interior.ColorIndex = 24
Range("A25").Interior.ColorIndex = 25
Range("A26").Interior.ColorIndex = 26
Range("A27").Interior.ColorIndex = 27
Range("A28").Interior.ColorIndex = 28
Range("A29").Interior.ColorIndex = 29
Range("A30").Interior.ColorIndex = 30
Range("A31").Interior.ColorIndex = 31
Range("A32").Interior.ColorIndex = 32
Range("A33").Interior.ColorIndex = 33
Range("A34").Interior.ColorIndex = 34
Range("A35").Interior.ColorIndex = 35
```

```

Range("A36").Interior.ColorIndex = 36
Range("A37").Interior.ColorIndex = 37
Range("A38").Interior.ColorIndex = 38
Range("A39").Interior.ColorIndex = 39
Range("A40").Interior.ColorIndex = 40
Range("A41").Interior.ColorIndex = 41
Range("A42").Interior.ColorIndex = 42
Range("A43").Interior.ColorIndex = 43
Range("A44").Interior.ColorIndex = 44
Range("A45").Interior.ColorIndex = 45
Range("A46").Interior.ColorIndex = 46
Range("A47").Interior.ColorIndex = 47
Range("A48").Interior.ColorIndex = 48
Range("A49").Interior.ColorIndex = 49
Range("A50").Interior.ColorIndex = 50
Range("A51").Interior.ColorIndex = 51
Range("A52").Interior.ColorIndex = 52
Range("A53").Interior.ColorIndex = 53
Range("A54").Interior.ColorIndex = 54
Range("A55").Interior.ColorIndex = 55
Range("A56").Interior.ColorIndex = 56

```

```
End Sub
```

## 22. Uruchom program.

### 1. Wyświetl okno arkusza Excel.

### 24. W komórkach od A1 do A56 wyświetlone zostały próbki kolorów (rysunek 7.24).

#### Rysunek 7.24.

Numer wiersza  
odpowiada wartości  
parametru *ColorIndex*

	A
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	



Uwaga

Jeżeli w programie z przykładu 40. właściwość *Interior* zostanie zastąpiona przez właściwość *Border*, zmienione zostaną kolory obramowań komórek.



Uwaga

Jeżeli w programie z przykładu 40. właściwość *Interior* zostanie zastąpiona przez właściwość *Font*, zmienione zostaną kolory czcionki.



## Przesuwanie wyboru komórki

W arkuszu kalkulacyjnym musi być wybrana jedna komórka. W niej będą np. pojawiały się znaki wpisywane z klawiatury. Wpisując lub odczytując dane, przesuwamy się do innej komórki. Do tego celu najwygodniej wykorzystać właściwość `Offset`.

Ma ona następującą składnię:

```
Offset(x,y)
```

gdzie:

*x* — przesunięcie w poziomie,

*y* — przesunięcie w pionie.

Wartości ujemne powodują odpowiednio przesunięcie w lewo lub w górę. Wartości dodatnie powodują odpowiednio przesunięcie w prawo lub w dół.

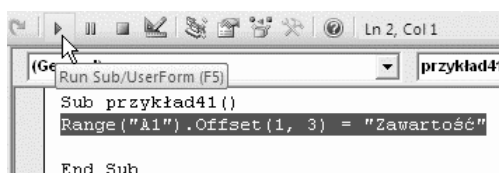
### Przykład 41.

Napisz program, który będzie przesuwał wybraną komórkę o jedną komórkę w dół i trzy komórki w prawo, a następnie do zaznaczonej komórki wstawi tekst Zawartość.

1. Uruchom arkusz kalkulacyjny Excel.
2. Naciśnij klawisze: *Alt+F11*. Spowoduje to załadowanie edytora Visual Basic.
3. Naciśnij klawisz *F7*. Spowoduje to wyświetlenie okna *Code*.
4. Wpisz instrukcję `Sub przykład41()`.
5. Naciśnij klawisz *Enter*.
6. Edytor dopisał linię `End Sub`.
7. Umieść znak wstawiania na końcu pierwszej linii.
8. Naciśnij klawisz *Enter*.
9. Wpisz z klawiatury kod `Range("A1").Offset(1, 3) = "Zawartość"`. Punktem odniesienia jest komórka A1. Właściwość `Offset` powoduje przesunięcie zaznaczenia o jedną komórkę w dół i trzy komórki w prawo. Ciąg znaków widoczny po znaku równości zostanie wstawiony do zaznaczonej komórki.
10. Uruchom program (rysunek 7.25).

#### Rysunek 7.25.

Kliknięcie ikony spowoduje uruchomienie programu



11. Wyświetl okno arkusza Excel (rysunek 7.26).

	A	B	C	D
1				
2				Zawartość

**Rysunek 7.26.** Zaznaczona jest komórka A1. Treść została wpisana do komórki odległej o zadane przesunięcie

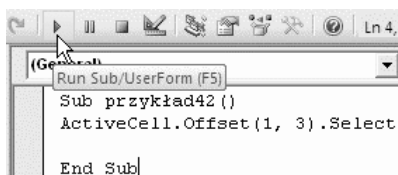
### Przykład 42.

Napisz program, który będzie przesuwał zaznaczanie komórki o jedną komórkę w dół i trzy komórki w prawo.

1. Uruchom arkusz kalkulacyjny Excel.
2. Naciśnij klawisze: *Alt+F11*. Spowoduje to załadowanie edytora Visual Basic.
3. Naciśnij klawisz *F7*. Spowoduje to wyświetlenie okna *Code*.
4. Wpisz instrukcję `Sub przykład42()`.
5. Naciśnij klawisz *Enter*.
6. Edytor dopisał linię `End Sub`.
7. Umieść znak wstawiania na końcu pierwszej linii.
8. Naciśnij klawisz *Enter*.
9. Wpisz z klawiatury kod `ActiveCell.Offset(1, 3).Select`. Punktem odniesienia jest komórka aktywna. Właściwość `Offset` powoduje przesunięcie zaznaczenia o jedną komórkę w dół i trzy komórki w prawo. Metoda `Select` powoduje wybranie komórki.
10. Wyświetl arkusz Excel. Domyślnie zaznaczona jest komórka *A1*.
11. Uruchom program (rysunek 7.27).

### Rysunek 7.27.

Kliknięcie ikony spowoduje uruchomienie programu



12. Wyświetl okno arkusza Excel (rysunek 7.28).

### Rysunek 7.28.

Zaznaczona jest komórka D2

	A	B	C	D
1				
2				