Temat: Obsługa wykresów - Matplotlib.

Tworzenie wykresów za pomocą biblioteki Matplotlib

- większość narzędzi biblioteki `matplotlib` znajduje się w module `pyplot`, której nazwę najczęściej importujemy jako alias `plt`

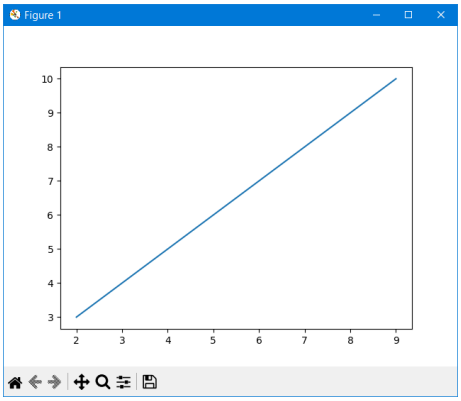
```
import matplotlib.pyplot as plt
```

A. Proste użycie funkcji plot()

a. Przykład. Kreślenie linii za pomocą współrzędnych x i y dwóch punktów.

- Funkcja `plot()` służy do rysowania punktów (znaczników) i linii na diagramie.
- Domyślnie funkcja `plot()` rysuje linię od punktu do punktu.
- Funkcja pobiera parametry do określania punktów na diagramie.
- Parametr 1 jest tablicą zawierającą punkty na osi x.
- Parametr 2 jest tablicą zawierającą punkty na osi y.
- Aby wykreślić linie od (2, 3) do (9, 10), musimy do funkcji przekazać dwie tablice: [2, 9] i [3, 10].
- Uwaga:
  Oś x jest osią poziomą.
  Oś y jest osią pionową.

```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([2, 9])
ypoints = np.array([3, 10])
plt.plot(xpoints, ypoints)
plt.show()
```

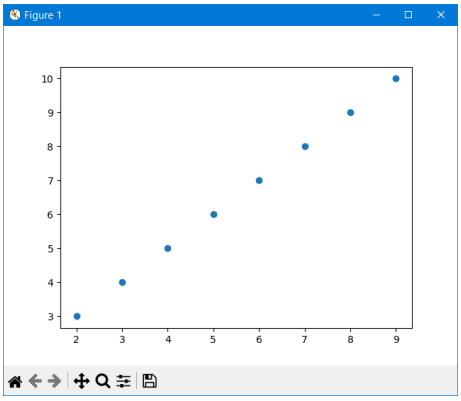


b. Przykład. Wykreślanie tylko punktów (bez linii) we współrzędnych x i y

- Umieść na wykresie osiem punktów o współrzędnych (2,3); (3,4); (4,5); (5,6); (6,7); (7,8); (8,9); (9,10).
- Aby wykreślić tylko znaczniki, możesz użyć parametru *shortcut string notation* "o", co oznacza "rings".

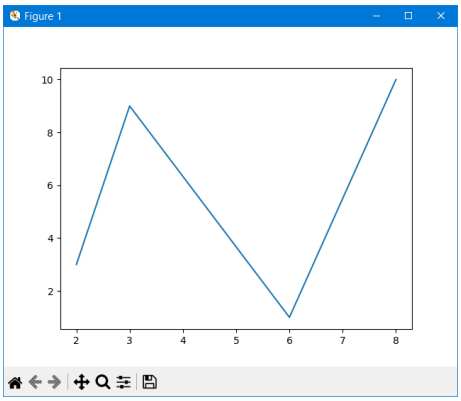
```
xpoints = np.array([2, 3, 4, 5, 6, 7, 8,  9])
ypoints = np.array([3, 4, 5, 6, 7, 8, 9, 10])
```

```
plt.plot(xpoints, ypoints, 'o')
plt.show()
```



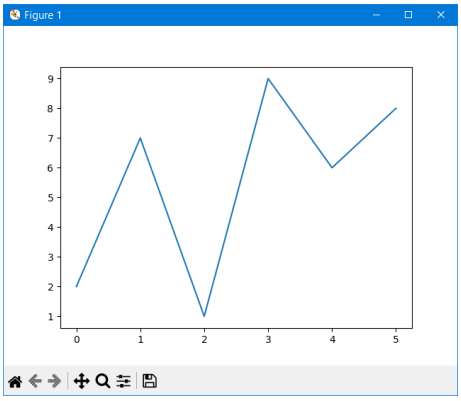
c. Przykład. Wykreślanie linii za pomocą współrzędnych x i y wielu punktów.

```
xpoints = np.array([1, 2, 6, 8])
ypoints = np.array([3, 8, 1, 10])
plt.plot(xpoints, ypoints)
plt.show()
```



d. Przykład. Wykreślanie linii za pomocą tylko współrzędnych y.

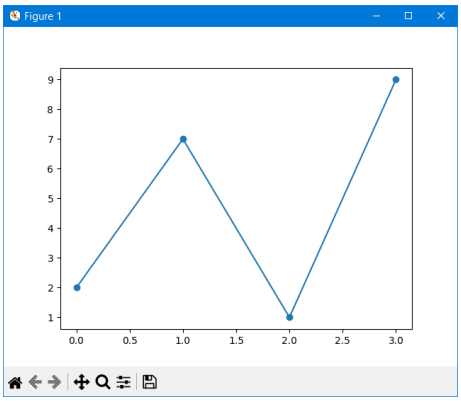
```
ypoints = np.array([2, 7, 1, 9, 6, 8])
plt.plot(ypoints)
plt.show()
```



B. Charakterystyka znaczników na wykresie.

a. Przykład. Wykreślanie linii wraz ze znacznikami punktów (`marker`).

```
ypoints = np.array([2, 7, 1, 9])
plt.plot(ypoints, marker = 'o')
plt.show()
```



Wypróbuj inne markery z dostępnych:

Marker	Description
'o'	Circle
'*'	Star
'.'	Point
','	Pixel
'x'	X
'X'	X (filled)
'+'	Plus
'P'	Plus (filled)
's'	Square
'D'	Diamond
'd'	Diamond (thin)
'p'	Pentagon
'H'	Hexagon
'h'	Hexagon
'v'	Triangle Down
'^'	Triangle Up
'<'	Triangle Left
'>'	Triangle Right
'1'	Tri Down
'2'	Tri Up
'3'	Tri Left
'4'	Tri Right
'|'	Vline
'_'	Hline

b. Przykład. Użycie parametru formatowania `fmt`.

parametr `fmt` ma składnię: *`marker|line|color`*

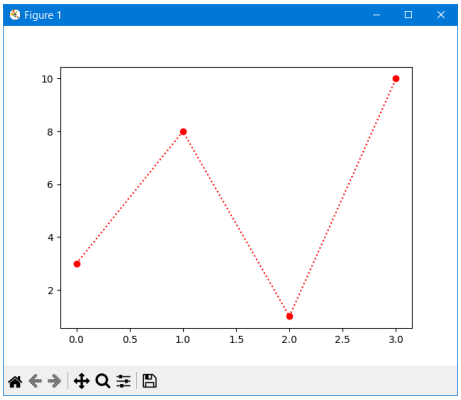
wartości dla opcji `line` można wybrać z poniższych:

```
Line Syntax     Description
'-'             Solid line
':'             Dotted line
'--'            Dashed line
'-.'            Dashed/dotted line
```

wartości dla opcji `color` można wybrać z poniższych:

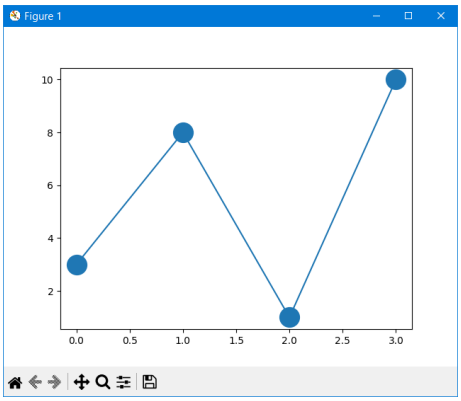
```
Color Syntax    Description
'r'             Red
'g'             Green
'b'             Blue
'c'             Cyan
'm'             Magenta
'y'             Yellow
'k'             Black
'w'             White
```

```
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, 'o:r')
plt.show()
```



c. Przykład. Można użyć parametru `markersize` (lub krócej ms) do określenia wielkości znacznika

```
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, marker = 'o', ms = 20)
plt.show()
```



d. Przykład. Można użyć parametrów `markeredgecolor` oraz `markerfacecolor` (lub krócej `mec` i `mfc`) do określenia koloru krawędzi i wypełnienia znacznika.

`markeredgecolor`

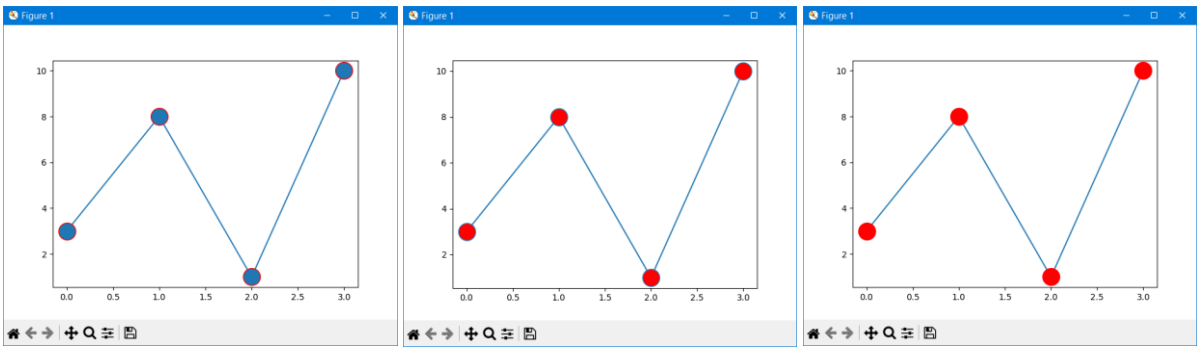
```
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, marker = 'o', ms = 20, mec = 'r')
plt.show()
```

`markerfacecolor`

```
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, marker = 'o', ms = 20, mfc = 'r')
plt.show()
```

`markeredgecolor` + `markerfacecolor`

```
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, marker = 'o', ms = 20, mec = 'r', mfc = 'r')
plt.show()
```



można także użyć nazw kolorów lub ich wartości heksadecymalnych z puli 140 kolorów

AliceBlue #F0F8FF
AntiqueWhite #FAEBD7
Aqua #00FFFF
Aquamarine #7FFFD4
Azure #F0FFFF
Beige #F5F5DC
Bisque #FFE4C4
Black #000000
BlanchedAlmond #FFEBCD
BlueViolet #8A2BE2
Brown #A52A2A
BurlyWood #DEB887
CadetBlue #5F9EA0
Chartreuse #7FFF00
Chocolate #D2691E
Coral #FF7F50
CornflowerBlue #6495ED
Cornsilk #FFF8DC

Crimson #DC143C
Cyan #00FFFF
DarkCyan #008B8B
DarkGoldenRod #B8860B
DarkGray #A9A9A9
DarkGrey #A9A9A9
DarkKhaki #BDB76B
DarkOliveGreen #556B2F
DarkOrange #FF8C00
DarkOrchid #9932CC
DarkSalmon #E9967A
DarkSeaGreen #8FBC8F
DarkSlateGray #2F4F4F
DarkSlateGrey #2F4F4F

DarkTurquoise #00CED1
DarkViolet #9400D3
DeepPink #FF1493
DeepSkyBlue #00BFFF
DimGray #696969
DimGrey #696969
DodgerBlue #1E90FF
FireBrick #B22222
FloralWhite #FFFAF0
ForestGreen #228B22
Fuchsia #FF00FF
Gainsboro #DCDCDC
GhostWhite #F8F8FF
Gold #FFD700
GoldenRod #DAA520
Gray #808080
Grey #808080
GreenYellow #ADFF2F

HoneyDew #F0FFF0
HotPink #FF69B4
IndianRed #CD5C5C
Ivory #FFFFF0
Khaki #F0E68C
Lavender #E6E6FA
LavenderBlush #FFF0F5
LawnGreen #7CFC00
LemonChiffon #FFFACD
LightBlue #ADD8E6
LightCoral #F08080
LightCyan #E0FFFF
LightGoldenRodYellow #FAFAD2
LightGray #D3D3D3
LightGrey #D3D3D3
LightGreen #90EE90
LightPink #FFB6C1
LightSalmon #FFA07A

LightSeaGreen #20B2AA
LightSkyBlue #87CEFA
LightSlateGray #778899
LightSlateGrey #778899
LightSteelBlue #B0C4DE
LightYellow #FFFFE0
Lime #00FF00
LimeGreen #32CD32
Linen #FAF0E6
Magenta #FF00FF
Maroon #800000
MediumAquaMarine #66CDAA
MediumBlue #0000CD
MediumOrchid #BA55D3
MediumPurple #9370DB
MediumSeaGreen #3CB371
MediumSlateBlue #7B68EE
MediumSpringGreen #00FA9A

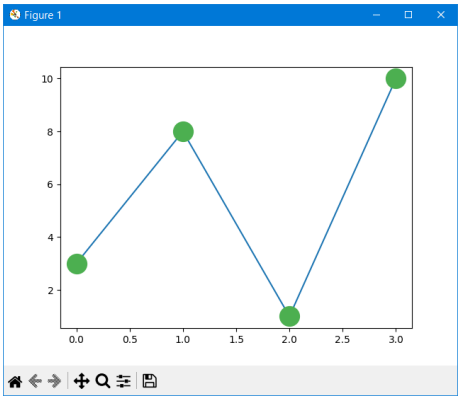
MediumTurquoise #48D1CC
MediumVioletRed #C71585
MidnightBlue #191970
MintCream #F5FFFA
MistyRose #FFE4E1
Moccasin #FFE4B5
NavajoWhite #FFDEAD
Navy #000080
OldLace #FDF5E6
Olive #808000
OliveDrab #6B8E23
Orange #FFA500
OrangeRed #FF4500
Orchid #DA70D6
PaleGoldenRod #EEE8AA
PaleGreen #98FB98
PaleTurquoise #AFEEEE
PaleVioletRed #DB7093

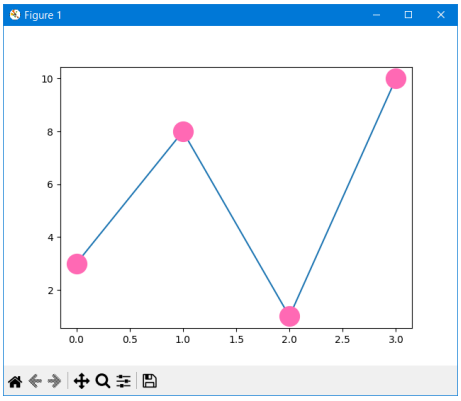
PapayaWhip #FFEFD5
PeachPuff #FFDAB9
Peru #CD853F
Pink #FFC0CB
Plum #DDA0DD
PowderBlue #B0E0E6
Purple #800080
RebeccaPurple #663399
Red #FF0000
RosyBrown #BC8F8F
RoyalBlue #4169E1
SaddleBrown #8B4513
Salmon #FA8072
SandyBrown #F4A460
SeaGreen #2E8B57
SeaShell #FFF5EE
Sienna #A0522D
Silver #C0C0C0

SkyBlue #87CEEB
SlateBlue #6A5ACD
SlateGray #708090
SlateGrey #708090
Snow #FFFAFA
SpringGreen #00FF7F
SteelBlue #4682B4
Tan #D2B48C
Teal #008080
Thistle #D8BFD8
Tomato #FF6347
Turquoise #40E0D0
Violet #EE82EE
Wheat #F5DEB3
White #FFFFFF
WhiteSmoke #F5F5F5
Yellow #FFFF00
YellowGreen #9ACD32

```
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, marker = 'o', ms = 20, mec = '#4CAF50', mfc = '#4CAF50')
plt.show()
```

```
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, marker = 'o', ms = 20, mec = 'hotpink', mfc = 'hotpink')
plt.show()
```





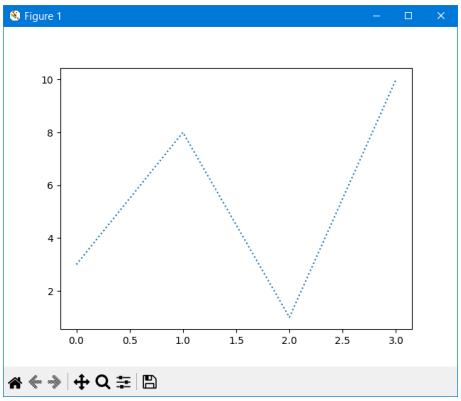
C. Określenie stylu linii.

a. Przykład. Użycie słowa kluczowego `linestyle` lub `ls`.

Rodzaj linii można wybrać na dwa sposoby

```
Style                   Or
'solid' (default)       '-'
'dotted'                ':'
'dashed'                '--'
'dashdot'               '-.'
'None'                  '' or ' '
```

```
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, linestyle = 'dotted')
plt.show()
```



b. Przykład. Użycie słowa kluczowego `color` lub skrótu `c` do określenia koloru linii

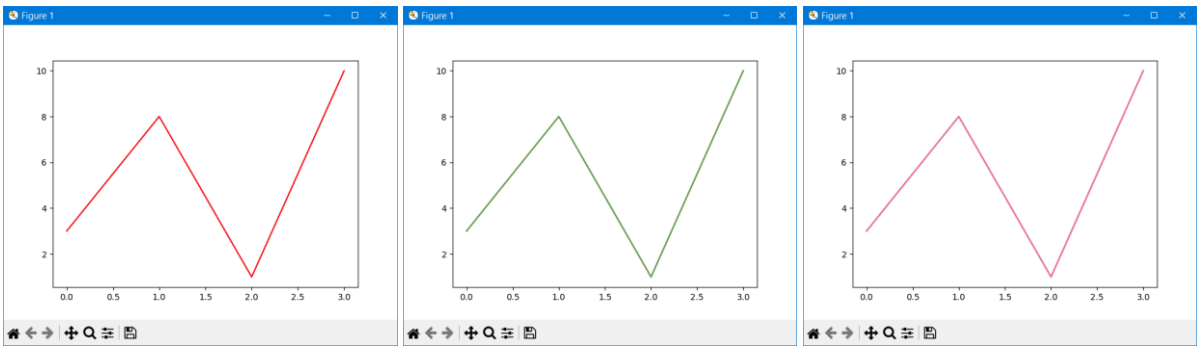
```
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, color = 'r')
plt.show()
```

druga wersja

```
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, c = '#4CAF50')
plt.show()
```

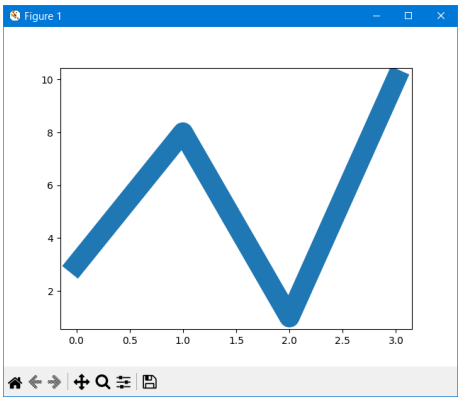
trzecia wersja

```
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, c = 'hotpink')
plt.show()
```



c. Przykład. Użycie słowa kluczowego `linewidth` lub `lw`.

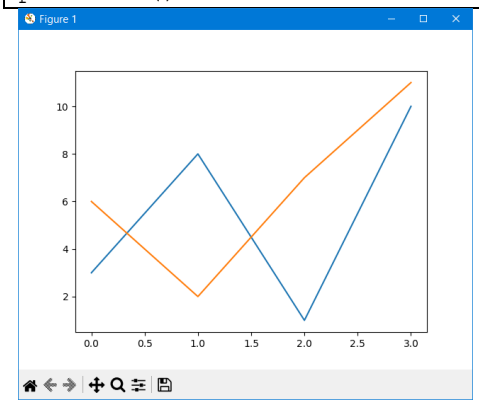
```
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, linewidth = '20.5')
plt.show()
```



d. Przykład. Rysowanie wielu linii poprzez użycie kilku wystąpień `plt.plot()`.

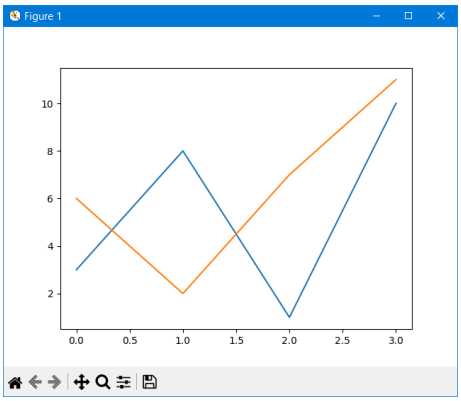
Wariant z podaniem tylko współrzędnych y (współrzędne x wyznaczane są domyślnie).

```
y1 = np.array([3, 8, 1, 10])
y2 = np.array([6, 2, 7, 11])

plt.plot(y1)
plt.plot(y2)

plt.show()
```



Wariant z podaniem obu współrzędnych x i y. Wartości te są sparowane.

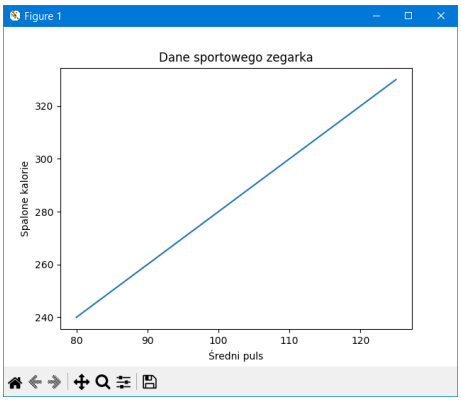
```
x1 = np.array([0, 1, 2, 3])
y1 = np.array([3, 8, 1, 10])
x2 = np.array([0, 1, 2, 3])
y2 = np.array([6, 2, 7, 11])

plt.plot(x1, y1, x2, y2)
plt.show()
```



D. Charakterystyka tytułu wykresu i etykiet.

a. Przykład. Użycie funkcji `xlabel()` `ylabel()` do ustawienia etykiet dla osi x i y.

```
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])
plt.plot(x, y)
plt.title("Dane sportowego zegarka")
plt.xlabel("Średni puls")
plt.ylabel("Spalone kalorie")
plt.show()
```

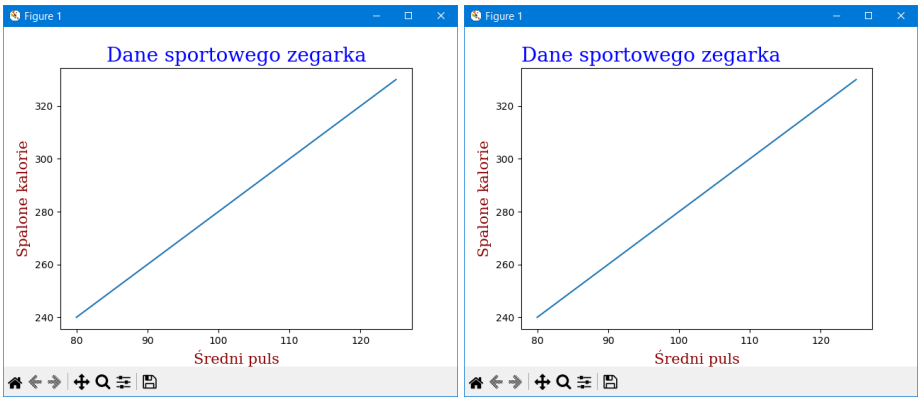


b. Przykład. Ustawienie czcionki dla tytułu i etykiet, parametr `fontdict`.

```
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

font1 = {'family':'serif','color':'blue','size':20}
font2 = {'family':'serif','color':'darkred','size':15}

plt.plot(x, y)

plt.title("Dane sportowego zegarka", fontdict = font1)
plt.xlabel("Średni puls", fontdict = font2)
plt.ylabel("Spalone kalorie", fontdict = font2)

plt.show()
```

Ponadto, w parametrze `title` można użyć słowa `loc` do określenia lokalizacji.

```
…
plt.title("Dane sportowego zegarka", fontdict = font1, loc="left")
…
```



E. Dodawanie linii `Grid` do `Plot`.

a. Przykład. Użycie funkcji `grid()` w celu dodania siatki.

```
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

font1 = {'family':'serif','color':'blue','size':20}
font2 = {'family':'serif','color':'darkred','size':15}

plt.plot(x, y)

plt.title("Dane sportowego zegarka")
plt.xlabel("Średni puls", fontdict = font2)
plt.ylabel("Spalone kalorie", fontdict = font2)

plt.grid(color = 'green', linestyle = '--', linewidth = 0.5)

plt.show()
```

inne warianty:

```
…
plt.grid(axis = 'x')
…
```

```
…
plt.grid(axis = 'y')
…
```

F. Tworzenie wykresów podrzędnych. Funkcja `subplot()`

- Funkcja `subplot()` przyjmuje trzy argumenty, które opisują układ wykresu.
- Układ jest zorganizowany w wiersze i kolumny, które są reprezentowane przez *pierwszy* i *drugi* argument.
- Trzeci argument reprezentuje indeks bieżącego podwykresu.

np. poniższe polecenie tworzy 1 wiersz i dwie kolumny, a następnie umieszcza pierwszy wykres w pierwszym polu:
`plt.subplot(1, 2, 1)`

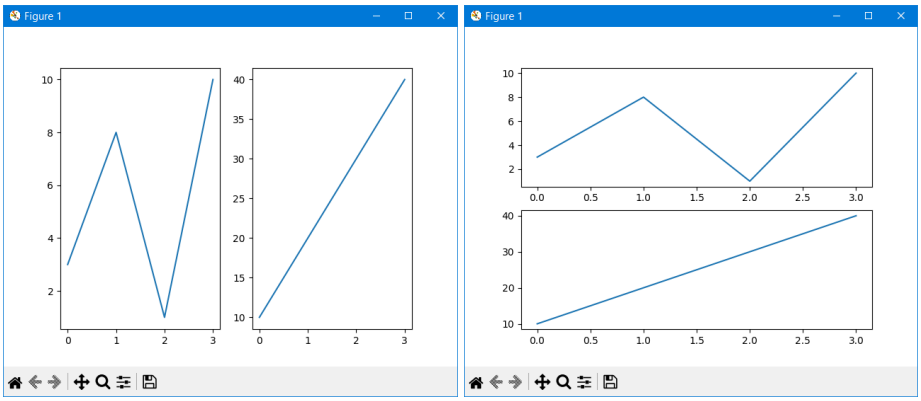
poniższe polecenie umieszcza drugi wykres w drugim polu:
`plt.subplot(1, 2, 2)`

a. Przykład. Użycie funkcji `subplot()` do wykreślenia dwóch wykresów podrzędnych.

```
#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(1, 2, 1)
plt.plot(x,y)

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)
plt.plot(x,y)

plt.show()
```

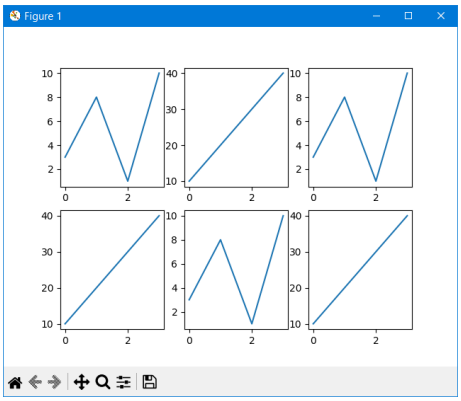
Jeśli chcemy mieć rysunek z 2 wierszami i 1 kolumną (co oznacza, że dwa wykresy będą wyświetlane jeden na drugim, a nie obok siebie), składnię polecenia możemy zapisać w następujący sposób:

```
#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(2, 1, 1)
plt.plot(x,y)

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(2, 1, 2)
plt.plot(x,y)

plt.show()
```



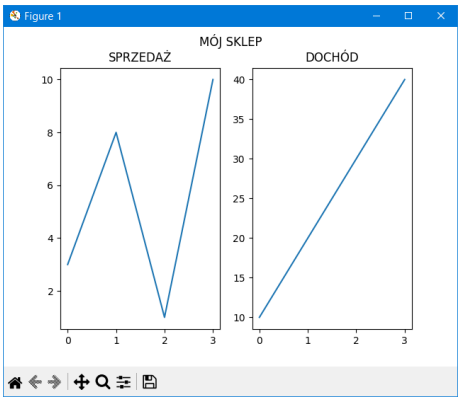
Bardziej rozbudowany przykład:

```
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(2, 3, 1)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(2, 3, 2)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(2, 3, 3)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(2, 3, 4)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(2, 3, 5)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(2, 3, 6)
plt.plot(x,y)

plt.show()
```



b. Do wykresu oraz podwykresów można dodać tytuły.

```
#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(1, 2, 1)
plt.plot(x,y)
plt.title("SPRZEDAŻ")

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)
plt.plot(x,y)
plt.title("DOCHÓD")

plt.suptitle("MÓJ SKLEP")
plt.show()
```



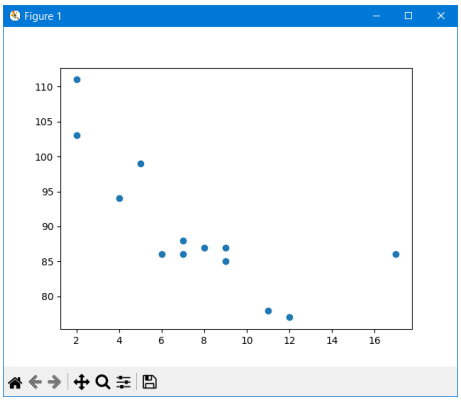
G. Tworzenie wykresów punktowych.

a. Przykład. Proste użycie funkcji `scatter()`.

- Funkcja `scatter()` wykreśla jeden punkt dla każdej obserwacji i potrzebuje dwóch tablic o tej samej długości: dla wartości na oś x i dla wartości na osi y.

```
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

plt.scatter(x, y)
```

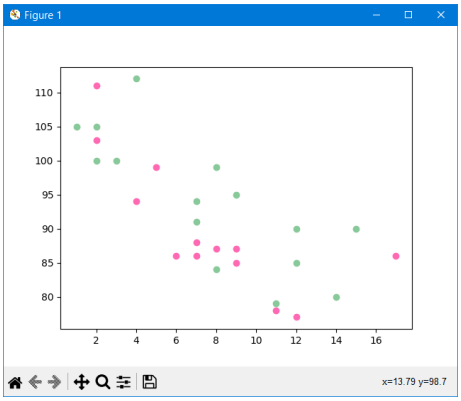
```
plt.show()
```



b. Przykład. Dwie serie punktów na jednym wykresie.

- Kolory poszczególnych serii punktów można określić za pomocą parametru color lub c.

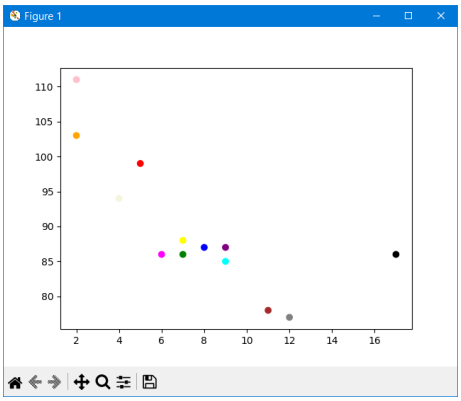
```
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
plt.scatter(x, y, color = 'hotpink')

x = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])
y = np.array([100,105,84,105,90,99,90,95,94,100,79,112,91,80,85])
plt.scatter(x, y, color = '#88c999')

plt.show()
```



c. Przykład. Określenie koloru dla każdego punktu danych.

- W tym przypadku nie można użyć parametru `color`, tylko `c`.

```
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
colors =
np.array(["red","green","blue","yellow","pink","black","orange","purple","b
eige","brown","gray","cyan","magenta"])

plt.scatter(x, y, c=colors)

plt.show()
```

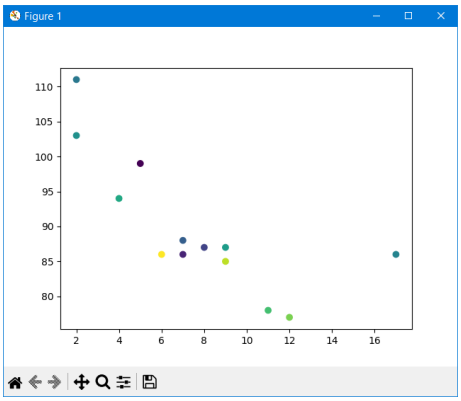


d. Przykład. Użycie mapy kolorów.

- Biblioteka matplotlib posiada wiele dostępnych map kolorów.
- Mapa kolorów jest jak lista kolorów, gdzie każdy kolor ma wartość z zakresu od 0 do 100.
- Oto przykład mapy kolorów "`viridis`" i jak widać zakres kolorów waha się od 0 jako kolor fioletowy do 100 który jest kolorem żółtym:

- Jak korzystać z `ColorMap`?
    - Mapę kolorów można określić za pomocą argumentu słowa kluczowego `cmap` z wartością mapy kolorów, w tym przypadku `viridis`, który jest jednym z wbudowanych map kolorów dostępnych w `Matplotlib`.
- Dodatkowo należy utworzyć tablicę z wartościami (od 0 do 100), po jednej wartości dla każdego punktu na wykresie punktowym:

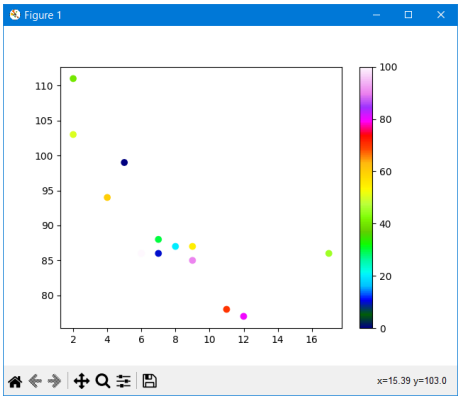
```
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
colors = np.array([0, 10, 20, 30, 40, 45, 50, 55, 60, 70, 80, 90, 100])
plt.scatter(x, y, c=colors, cmap='viridis')
plt.show()
```



e. Przykład. Umieszczenie mapy kolorów na wykresie.

- Parametr `plt.colorbar()`

```
…
plt.scatter(x, y, c=colors, cmap='gist_ncar')
plt.colorbar()
plt.show()
```



- Lista dostępnych map kolorów:

Accent	Accent_r	
Blues	Blues_r	
BrBG	BrBG_r	
BuGn	BuGn_r	
BuPu	BuPu_r	
CMRmap	CMRmap_r	
Dark2	Dark2_r	
GnBu	GnBu_r	
Greens	Greens_r	
Greys	Greys_r	
OrRd	OrRd_r	
Oranges	Oranges_r	
PRGn	PRGn_r	
Paired	Paired_r	
Pastel1	Pastel1_r	
Pastel2	Pastel2_r	
PiYG	PiYG_r	
PuBu	PuBu_r	
PuBuGn	PuBuGn_r	
PuOr	PuOr_r	
PuRd	PuRd_r	

Purples	Purples_r	
RdBu	RdBu_r	
RdGy	RdGy_r	
RdPu	RdPu_r	
RdYlBu	RdYlBu_r	
RdYlGn	RdYlGn_r	
Reds	Reds_r	
Set1	Set1_r	
Set2	Set2_r	
Set3	Set3_r	
Spectral	Spectral_r	
Wistia	Wistia_r	
YlGn	YlGn_r	
YlGnBu	YlGnBu_r	
YlOrBr	YlOrBr_r	
YlOrRd	YlOrRd_r	
afmhot	afmhot_r	
autumn	autumn_r	
binary	binary_r	
bone	bone_r	
brg	brg_r	

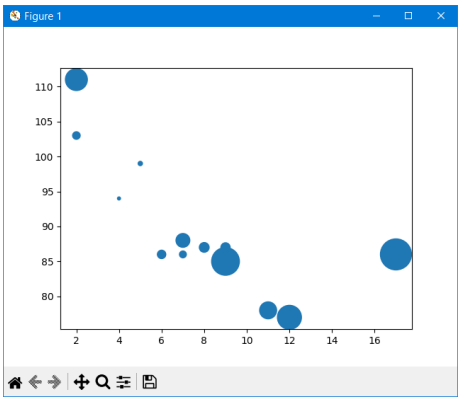
bwr	bwr_r	
cividis	cividis_r	
cool	cool_r	
coolwarm	coolwarm_r	
copper	copper_r	
cubehelix	cubehelix_r	
flag	flag_r	
gist_earth	gist_earth_r	
gist_gray	gist_gray_r	
gist_heat	gist_heat_r	
gist_ncar	gist_ncar_r	
gist_rainbow	gist_rainbow_r	
gist_stern	gist_stern_r	
gist_yarg	gist_yarg_r	
gnuplot	gnuplot_r	
gnuplot2	gnuplot2_r	
gray	gray_r	
hot	hot_r	
hsv	hsv_r	
inferno	inferno_r	
jet	jet_r	

magma	magma_r
nipy_spectral	nipy_spectral_r
ocean	ocean_r
pink	pink_r
plasma	plasma_r
prism	prism_r
rainbow	rainbow_r
seismic	seismic_r
spring	spring_r
summer	summer_r
tab10	tab10_r
tab20	tab20_r
tab20b	tab20b_r
tab20c	tab20c_r
terrain	terrain_r
twilight	twilight_r
twilight_shifted	twilight_shifted_r
viridis	viridis_r
winter	winter_r

f. Przykład. Zmiana rozmiaru punktów, parametr `s`.

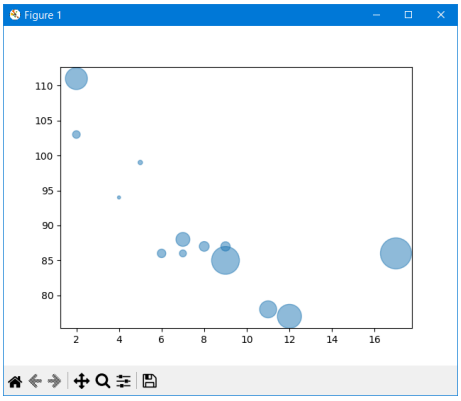
- Podobnie jak w przypadku kolorów, należy upewnić się, że tablica rozmiarów ma taką samą długość jak tablice dla osi x i y:

```
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
sizes = np.array([20,50,100,200,500,1000,60,90,10,300,600,800,75])

plt.scatter(x, y, s=sizes)
plt.show()
```



g. Przykład. Ustawienie przezroczystości punktów na wykresie. Parametr `alpha`.

```
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
sizes = np.array([20,50,100,200,500,1000,60,90,10,300,600,800,75])

plt.scatter(x, y, s=sizes, alpha=0.5)

plt.show()
```

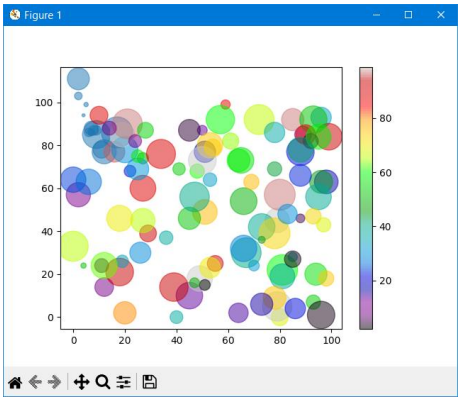


h. Przykład. Połączenie koloru, rozmiaru i przezroczystości.

- Utwórz losowe tablice ze 100 wartościami dla punktów x, punktów y, kolorów i rozmiarów.

```
x = np.random.randint(100, size=(100))
y = np.random.randint(100, size=(100))
colors = np.random.randint(100, size=(100))
sizes = 10 * np.random.randint(100, size=(100))
```

```
plt.scatter(x, y, c=colors, s=sizes, alpha=0.5, cmap='nipy_spectral')
plt.colorbar()

plt.show()
```



H. Tworzenie wykresów słupkowych.

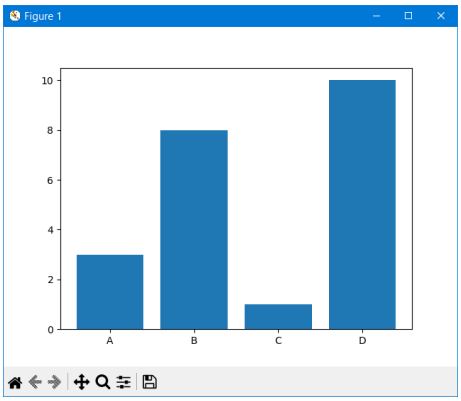
a. Przykład. Użycie funkcji `bar()` i `barh()` do rysowania wykresów słupkowych.

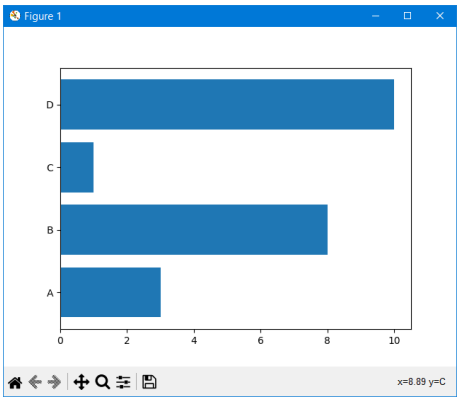
Słupki pionowe – `bar()`

```
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x,y)
plt.show()
```

Słupki poziome – `barh()`

```
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.barh(x, y)
plt.show()
```





b. Przykład. Określenie koloru słupków. Parametr `color`.

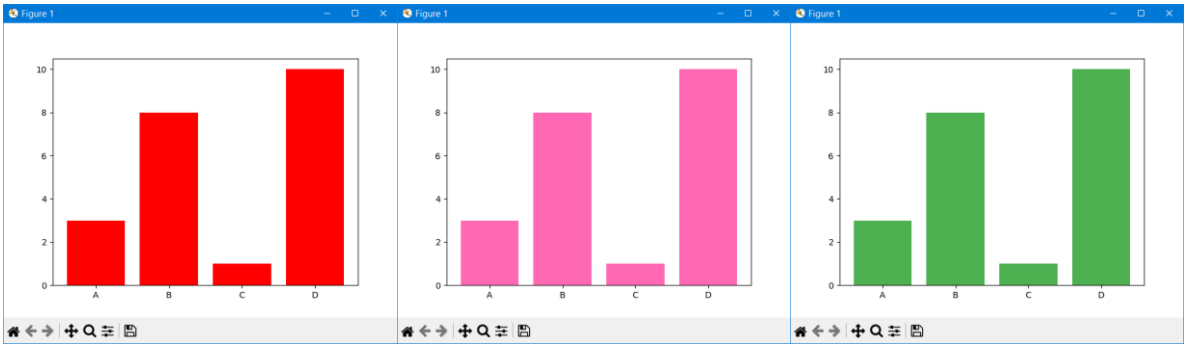
```
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x, y, color = "red")
plt.show()
```

lub jeden ze 140 zdefiniowanych kolorów

```
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x, y, color = "hotpink")
plt.show()
```

lub za pomocą kodu szesnastkowego

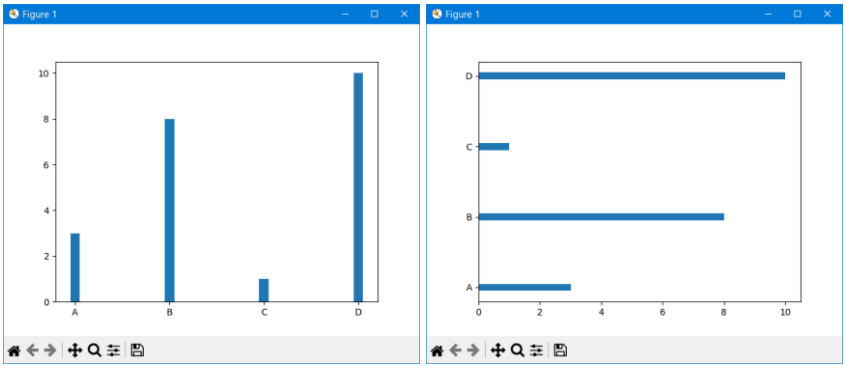
```
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x, y, color = "#4CAF50")
plt.show()
```



c. Przykład. Ustawienie szerokości i wysokości słupka, parametry `width` i `height`.

```
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x, y, width = 0.1)
plt.show()
```

```
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.barh(x, y, height = 0.1)
plt.show()
```



I. Tworzenie histogramów.

- Histogram to wykres przedstawiający rozkład częstości zmiennej, czyli liczbę obserwacji w każdym przedziale histogramu.

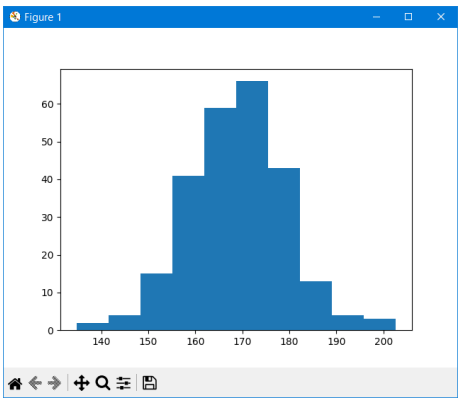
a. Przykład. Użycie funkcji `hist()`.

- Funkcja `hist()` używa tablicy liczb do utworzenia histogramu, tablica jest wysyłana do funkcji jako argument.
- Dla uproszczenia użyjemy NumPy do losowego generowania tablicy z 250 wartościami, gdzie wartości będą koncentrować się wokół 170, a odchylenie standardowe wynosi 10.

```
import numpy as np

x = np.random.normal(170, 10, 250)
print(x)
```

```
[167.37978318 184.03102243 176.98975443 175.38443473 179.23477447
 171.4853175  168.78233792 178.32218066 177.91087311 160.94589194
 173.96188329 172.40749508 175.11628254 177.01180945 155.32219678
 160.94578082 158.40797654 163.71461862 155.65680144 157.9165313
 170.53710961 165.5155922  163.02009454 171.62969932 166.40143163
 168.81398644 154.56208727 175.7821266  148.31200596 164.77188081
 172.56666268 162.4473838  150.42075648 178.66570848 151.67049027
 175.76179317 178.86748225 166.24461428 168.05589144 152.22719281
 179.29450119 167.65524271 183.46390083 186.38974031 176.49234976
 179.61993975 175.43264747 161.92148073 173.42423936 175.35698073
 181.4650684  168.61512567 160.84790584 169.95553612 163.00972383
 169.33612451 172.86666145 152.79494457 173.81587897 172.83266764
 164.79284613 177.97005852 176.04750762 164.87884133 178.06520593
 176.44209842 187.76229382 169.2793046  174.94295787 153.45589545
 172.04969174 180.86253741 197.13465749 154.84796563 171.1988829
 170.78629994 164.15369984 180.00058322 175.27173184 187.67597404
 174.24537135 166.38069254 162.47580523 166.26740177 153.95562061
 159.63565432 162.41608316 171.14759467 173.21240213 161.77157698
 164.88852769 192.41861745 174.02822527 158.5889086  159.74075994
 163.11369444 174.30060007 172.2068869  158.61368752 161.16521228
 162.77115648 183.99850908 187.24857063 191.85243747 162.04098127
 160.06832184 179.70823211 181.46337302 176.69674892 182.16995139
 171.1570972  172.41828105 165.65768253 173.12448479 166.48721884
 167.18966018 175.22796556 191.36740449 177.55118769 177.25936443
 155.5177305  189.03842349 173.60247042 175.34966775 186.428631
 176.49943956 153.65196709 175.84772292 173.46868285 176.94190949
 167.5784394  171.93425234 182.10347975 161.56508905 170.63221372
 163.37956656 164.35203165 160.28195206 190.17748051 168.58029932
 177.11588303 167.95058232 159.89649109 157.1886813  176.51240902
 185.53677694 179.98816149 176.23208851 173.84944737 169.10655985
 172.11995363 171.62343949 164.54716398 171.70610351 164.09192525
 171.05835423 180.93306202 177.65756402 172.1864204  166.89003115
 180.52079348 153.81840546 162.86135871 158.63966015 174.9495508
 157.36632013 162.42522522 188.58917464 159.11537371 168.56270123
 172.85032748 178.68970737 158.43581582 166.88532773 171.07568212
 165.86176619 184.79314351 181.77242653 199.55782582 167.52734661
 173.18722117 185.69166165 194.82314955 170.04832894 180.18375472
 174.75941016 161.71223408 166.97954981 177.42116972 169.08730704
 176.83746056 178.39033974 170.48507995 159.49376493 161.66388454
 180.93165285 164.85029029 160.56541804 155.87371767 166.38284698
 170.5422743  173.52994897 179.73967731 166.37549609 157.7689237
 173.33669382 187.97153344 165.69607775 163.71706906 177.23256271
 163.57669174 163.22821432 176.60399976 151.20061483 177.17880129
 178.64196152 179.89571601 159.84820849 168.38617089 172.35800554
 150.67552472 169.96053525 177.24772458 167.24690268 167.55897043
 179.99418697 162.68658963 167.7966882  176.18487762 161.23509507
 162.1063315  157.36446732 189.06572127 173.19780187 152.31581112
 185.03815656 181.19730161 171.09215733 183.79525262 152.29990982
 168.22586084 169.68139276 179.8973007  175.13835353 185.05141517
 172.74486214 181.35008202 170.62227704 180.20872736 156.42917718]
```

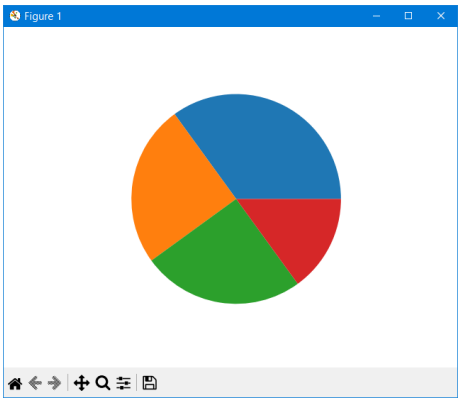
```
x = np.random.normal(170, 10, 250)

plt.hist(x)
plt.show()
```



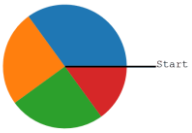
J. Tworzenie wykresów kołowych.

a. Przykład. Użycie funkcji `pie()`.

```
y = np.array([35, 25, 25, 15])

plt.pie(y)
plt.show()
```



- Jak widać, wykres kołowy rysuje jeden element (zwany klinem) dla każdej wartości w tablicy (w tym przypadku [35, 25, 25, 15]).
- Domyślnie kreślenie pierwszego klina rozpoczyna się od osi x i przesuwa się *w kierunku przeciwnym do ruchu wskazówek zegara*:

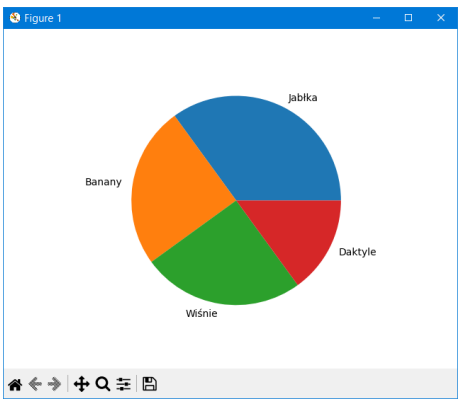


- Rozmiar każdego klina określa się, porównując wartość ze wszystkimi innymi wartościami za pomocą następującego wzoru: $x/sum(x)$ - wartość podzielona przez sumę wszystkich wartości.

b. Przykład. Używanie etykiet, parametr `labels`.

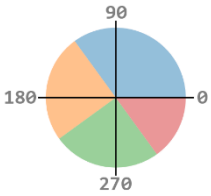
- Parametr `labels` musi być tablicą z jedną etykietą dla każdego klina.

```
y = np.array([35, 25, 25, 15])
mylabels = ["Jabłka", "Banany", "Wiśnie", "Daktyle"]

plt.pie(y, labels = mylabels)
plt.show()
```



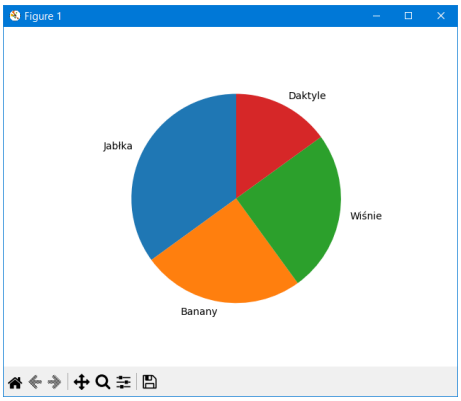
c. Przykład. Ustalenie kąta początkowego, parametr `startangle`.

- Jak wspomniano, domyślny kąt początkowy znajduje się na osi x, ale można go zmienić, określając parametr `startangle`.
- Parametr `startangle` jest zdefiniowany za pomocą kąta w stopniach, domyślny kąt to 0



d. Przykład. Rozpoczęcie pierwszego klina od 90 stopni.

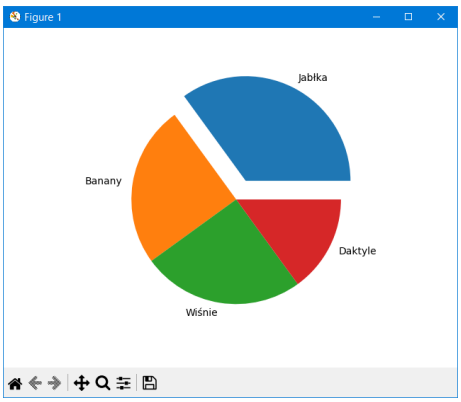
```
y = np.array([35, 25, 25, 15])
mylabels = ["Jabłka", "Banany", "Wiśnie", "Daktyle"]

plt.pie(y, labels = mylabels, startangle = 90)
plt.show()
```



e. Przykład. Wyróżnienie jednego klina przez wysunięcie, parametr `explode`.

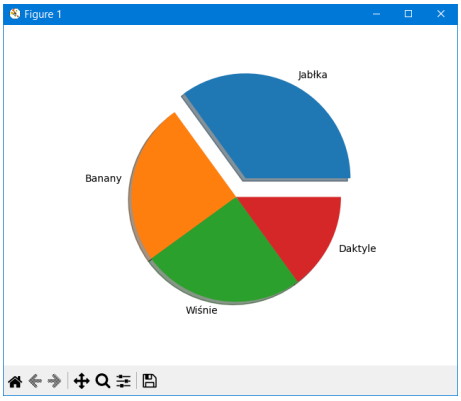
- Jeżeli parametr `explode` został wyspecyfikowany, to musi być tablicą z jedną wartością dla każdego klina.
- Każda z wartości określa jak daleko od środka są wysunięte poszczególne kliny.

```
y = np.array([35, 25, 25, 15])
mylabels = ["Jabłka", "Banany", "Wiśnie", "Daktyle"]
myexplode = [0.2, 0, 0, 0]

plt.pie(y, labels = mylabels, explode = myexplode)
plt.show()
```



f. Przykład. Dodanie cienia do wykresu, parametr `shadow`.

```
y = np.array([35, 25, 25, 15])
mylabels = ["Jabłka", "Banany", "Wiśnie", "Daktyle"]
myexplode = [0.2, 0, 0, 0]

plt.pie(y, labels = mylabels, explode = myexplode, shadow = True)
plt.show()
```

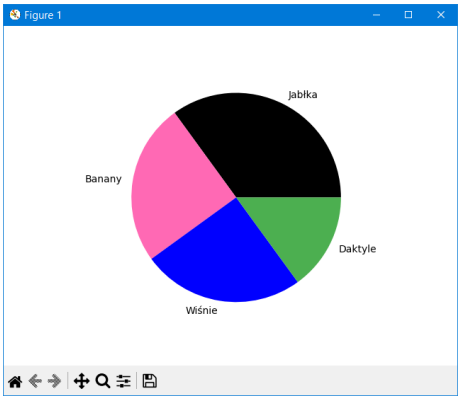


g. Przykład. Ustawienie koloru dla każdego klina, parametr `color`.

- można użyć szesnastokowych wartości kolorów, dowolnej nazwy ze 140 wspieranych kolorów lub jednego z poniższych skrótów:

'r' - Red
'g' - Green
'b' - Blue
'c' - Cyan
'm' - Magenta
'y' - Yellow
'k' - Black
'w' - White

```
y = np.array([35, 25, 25, 15])
mylabels = ["Jabłka", "Banany", "Wiśnie", "Daktyle"]
mycolors = ["black", "hotpink", "b", "#4CAF50"]

plt.pie(y, labels = mylabels, colors = mycolors)
plt.show()
```



h. Przykład. Użycie legendy, funkcja `legend()`.

- do legendy można dołączyć tytuł, używając opcji `title`.

```
y = np.array([35, 25, 25, 15])
mylabels = ["Jabłka", "Banany", "Wiśnie", "Daktyle"]
plt.pie(y, labels = mylabels)

plt.legend(title = "Cztery owoce:")
plt.show()
```