[**http://wmii.uwm.edu.pl/~cezary/WD/cw9.pdf**](http://wmii.uwm.edu.pl/~cezary/WD/cw9.pdf)

[**https://www.w3schools.com/python/matplotlib\_labels.asp**](https://www.w3schools.com/python/matplotlib_labels.asp)

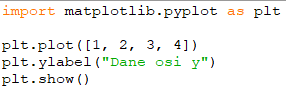
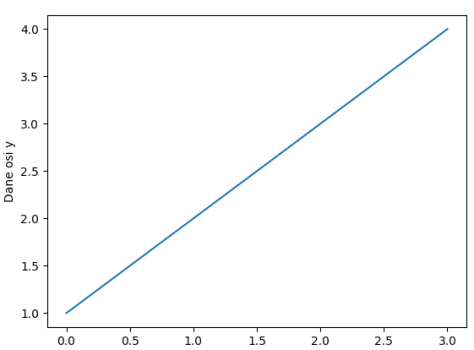
[**https://inetum.pl/biblioteka-matplotlib/**](https://inetum.pl/biblioteka-matplotlib/)

**https://python101.readthedocs.io/pl/latest/pylab/**

**Wykresy w Pythonie**

[**matplotlib.pyplot**](https://matplotlib.org/stable/api/pyplot_summary.html#module-matplotlib.pyplot)to zbiór funkcji, dzięki którym Matplotlib działa jak MATLAB. Każda pyplotfunkcja powoduje pewne zmiany w figurze: np. tworzy figurę, tworzy obszar kreślenia na figurze, kreśli niektóre linie w obszarze kreślenia, ozdabia wykres etykietami.

Wizualizacja wykresu za pomocą pyplot.



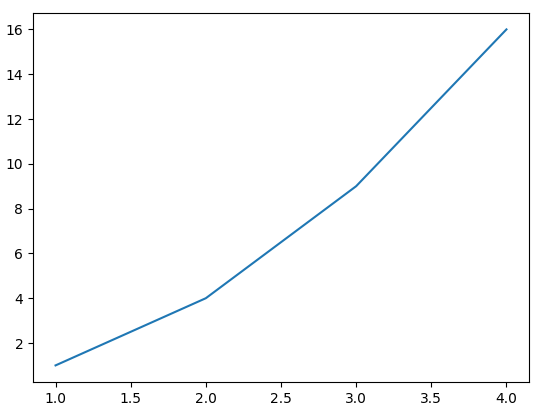
Jest to sekwencja wartości y i automatycznie generuje wartości x

[**plot**](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html#matplotlib.pyplot.plot) jest funkcją wszechstronną i przyjmuje dowolną liczbę argumentów. Aby wykreślić wykresy punktowe, gdy znaczniki mają identyczny rozmiar i kolor

**ylabel -** Tekst etykiety

**show()** – wyświetlenie okna

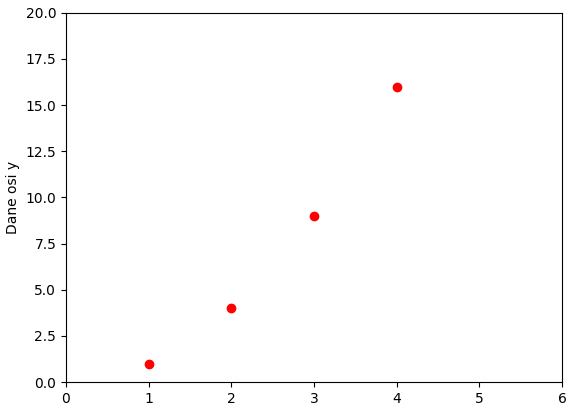
Zmieńmy

**Formatowanie stylu działki**

Dla każdej pary argumentów x, y istnieje opcjonalny trzeci argument, który jest ciągiem formatującym wskazującym kolor i rodzaj linii wykresu. Litery i symbole ciągu formatującego pochodzą z MATLAB-a, a ciąg kolorów łączy się z ciągiem o stylu linii. Domyślnym ciągiem formatującym jest „b-”, czyli ciągła niebieska linia. Na przykład, aby narysować powyższe za pomocą czerwonych kółek, dodałbyś

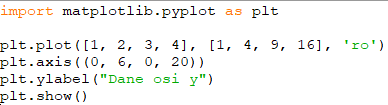


Funkcja **[axis](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.axis.html" \l "matplotlib.pyplot.axis" \o "oś matplotlib.pyplot)** z powyższego przykładu pobiera listę i określa rzutnię osi. 

**Ustalenie rozrzutu na osiach:**

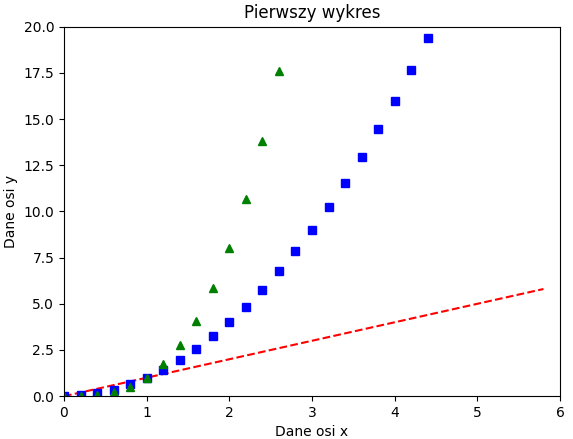
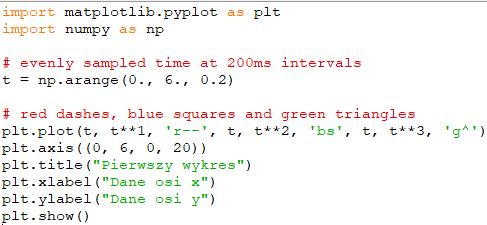
plt.xticks(np.arange(0,15,1))

plt.yticks(np.arange(0,15,1))



****

Możesz dodać tytuł do całej grupy figur za pomocą suptitle()





Dzięki tej subplot()funkcji możesz narysować wiele wykresów na jednym rysunku.

Funkcja subplot()przyjmuje trzy argumenty opisujące układ figury.

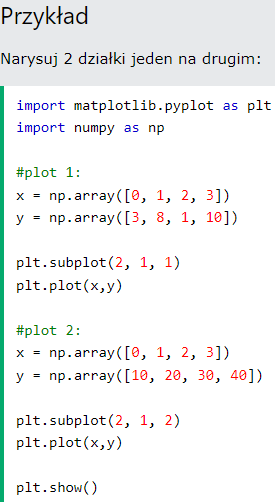
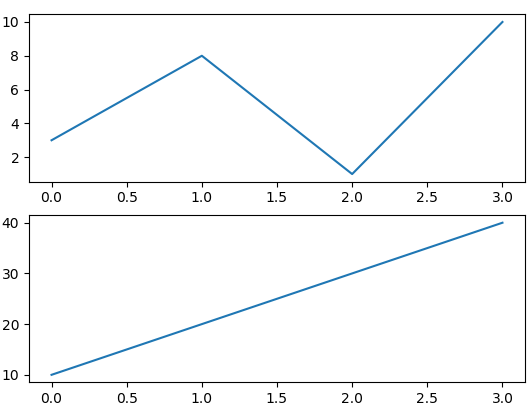
Układ jest zorganizowany w wiersze i kolumny, które są reprezentowane przez pierwszy i drugi argument.

Trzeci argument reprezentuje indeks bieżącego wykresu.

Dodawanie wspólnego tytułu dla wykresów **plt.suptitle()**

# Podstawy analizy danych w języku Python - Wiele Wykresów

https://www.youtube.com/watch?app=desktop&v=DbMhok1uHEE

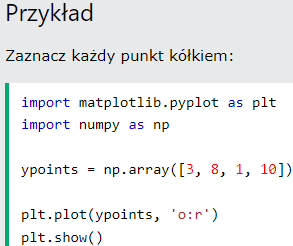
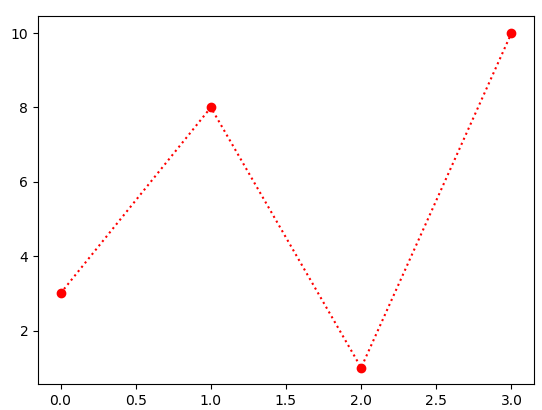
**Formatuj ciągi**

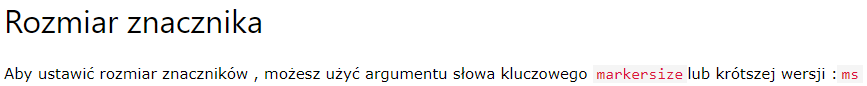
Ciąg formatujący składa się z części odpowiadającej kolorowi, znacznikowi i linii:

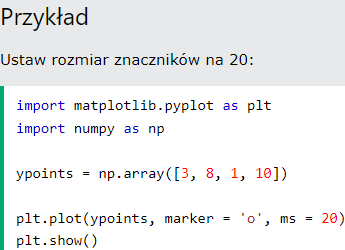
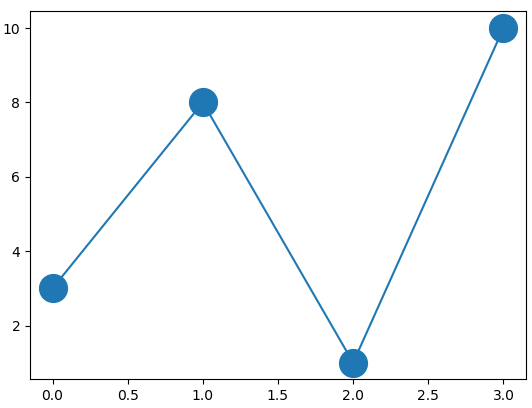


Do określenia znacznika można także użyć parametru *zapisu ciągu skrótu .*

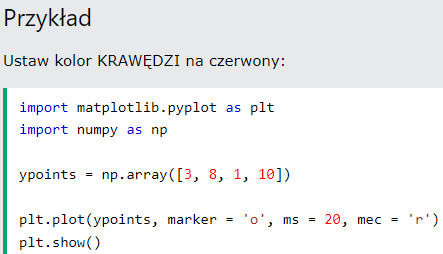
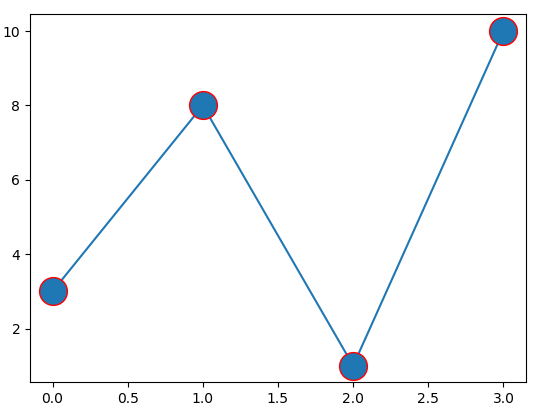
Ten parametr jest również nazywany fmt i jest zapisywany przy użyciu następującej składni: 



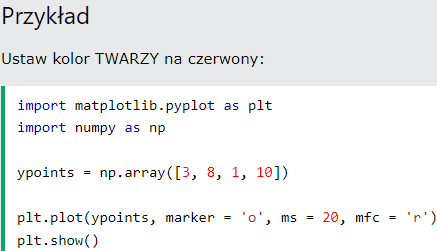
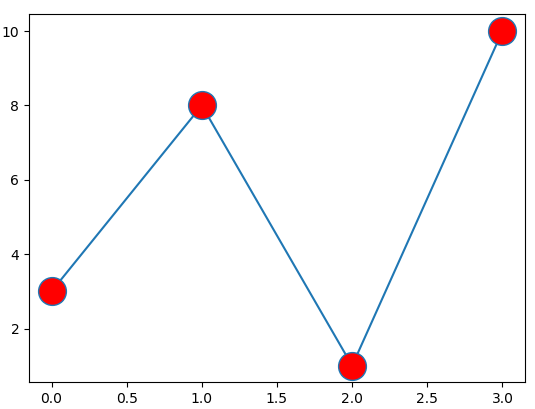
 



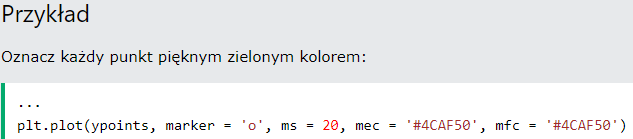
 

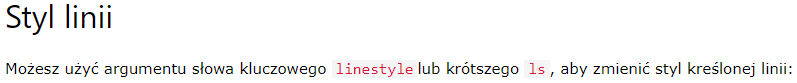


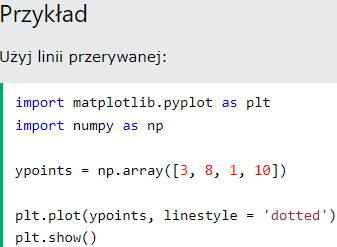
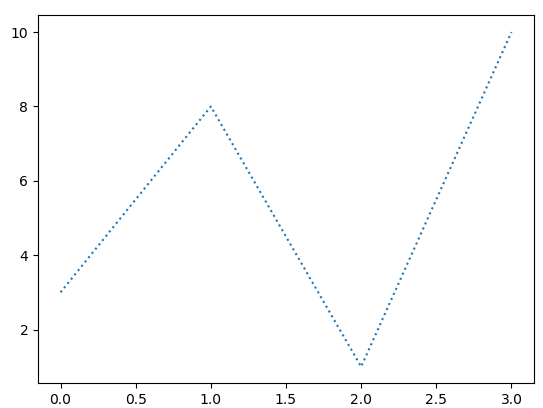
markeredgewidth grubość krawędzi markera, skrót **mew**

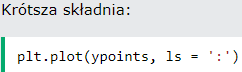
 

Możesz także użyć [szesnastkowych wartości kolorów](https://www.w3schools.com/colors/colors_hexadecimal.asp) :

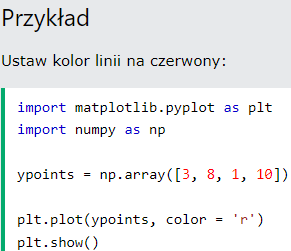
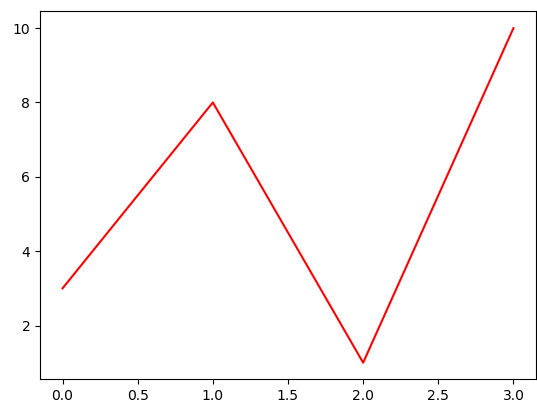




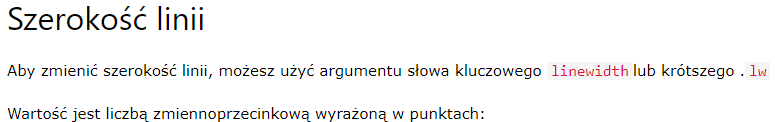
 

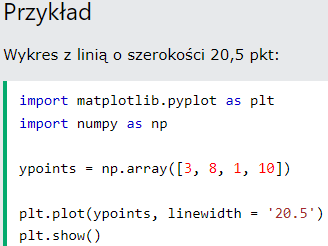
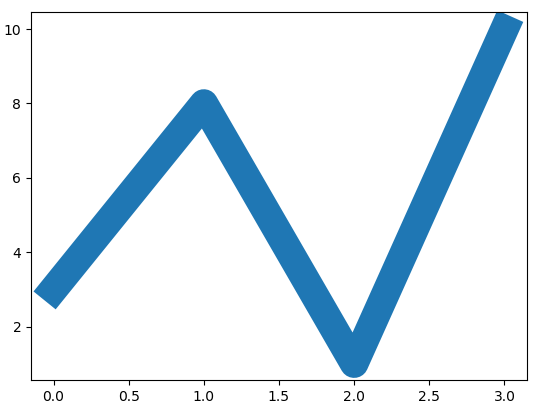


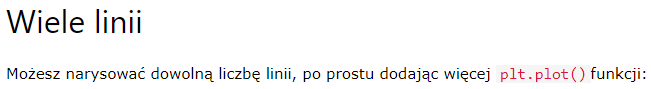


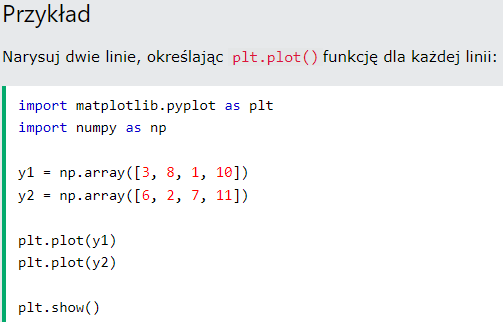
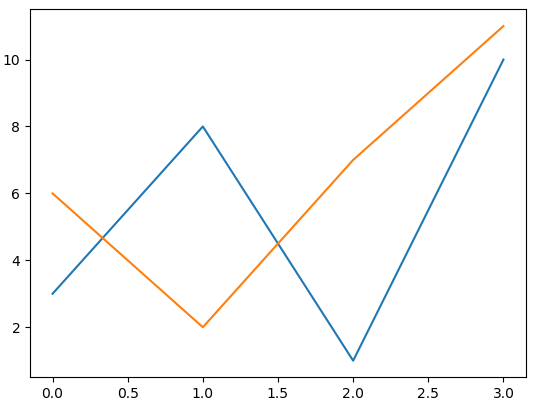
 

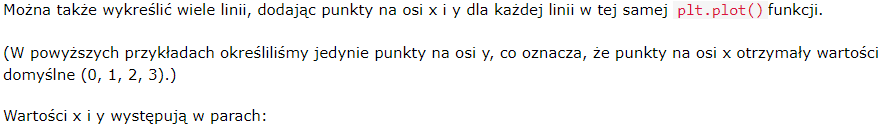
Możesz także użyć [szesnastkowych wartości kolorów](https://www.w3schools.com/colors/colors_hexadecimal.asp) lub dowolna ze [140 obsługiwanych nazw kolorów](https://www.w3schools.com/colors/colors_names.asp)

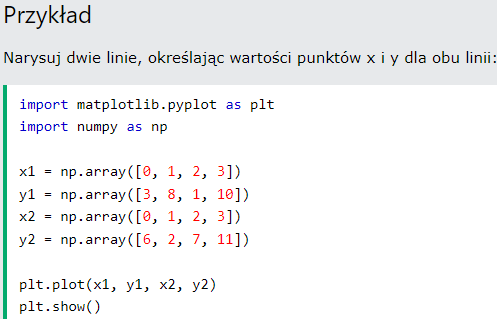
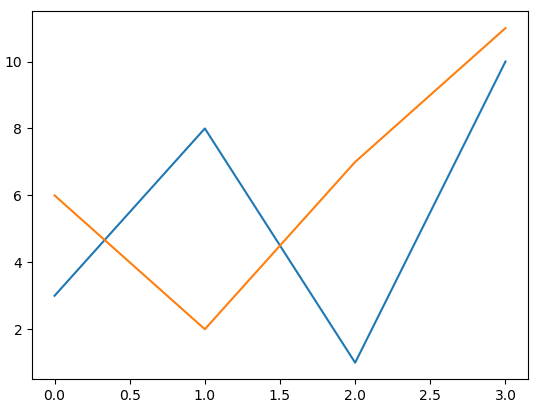






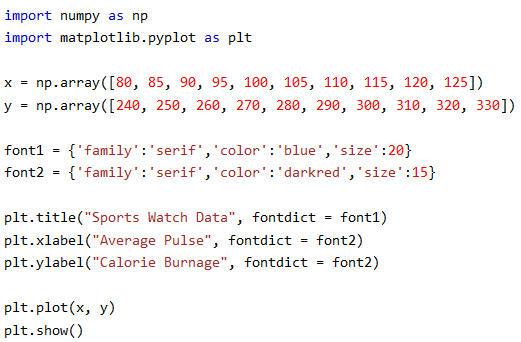
 

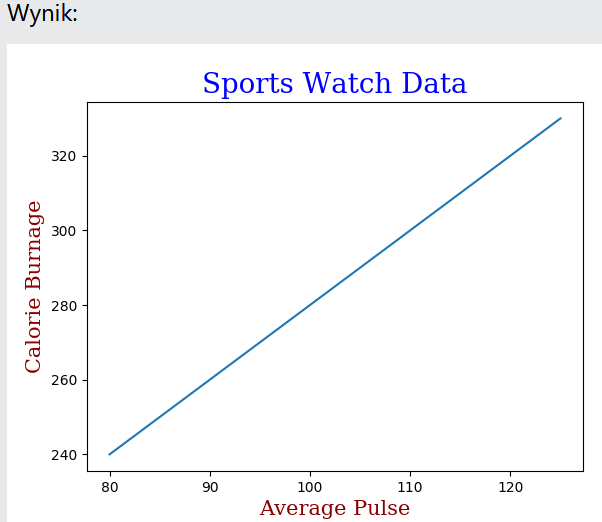
**Tekst na wykresie**

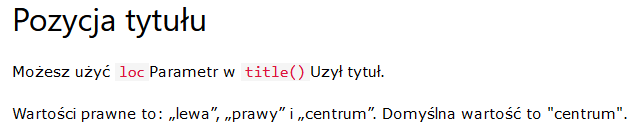
Współrzędne x,y

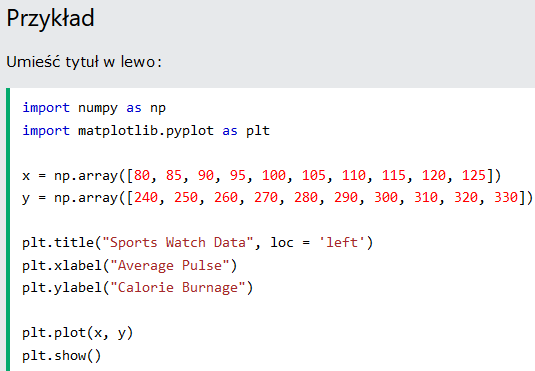
## Ustaw właściwości czcionki dla tytułu i etykiet

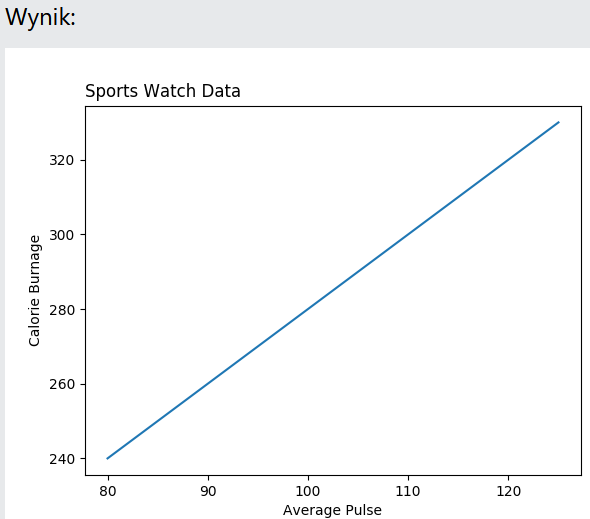












# Podpis Matplotlib

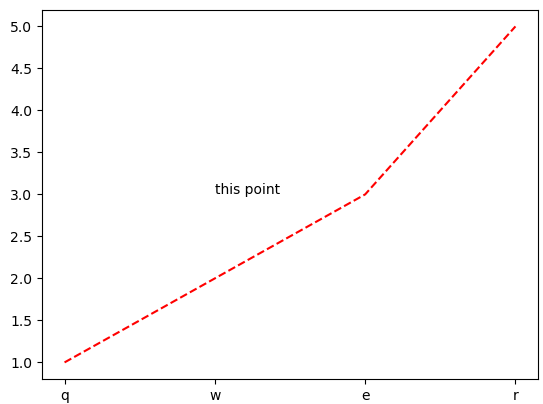
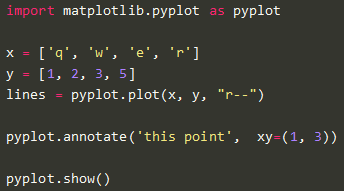
https://www.w3schools.com/python/matplotlib\_subplot.asp

# Matplotlib Dodawanie linii siatki

https://www.w3schools.com/python/matplotlib\_grid.asp

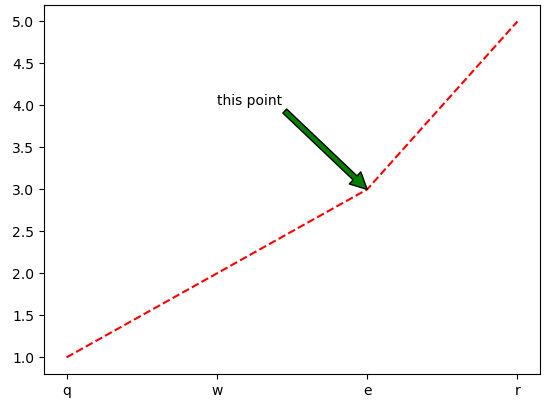
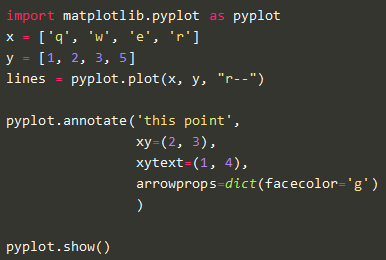
**Dodatkowy tekst na wykresie**

Aby oznaczyć kluczowe informacje, posługując się metodą **annotate()**, możemy wygenerować na przykład tekst na wykresie. Jako pierwszy argument funkcji podajemy opis, który pojawi się na wykresie. Następnie argument **xy**, który jest **krotką**dwóch wartości, współrzędnych, gdzie ma znaleźć się wcześniej wpisany tekst.



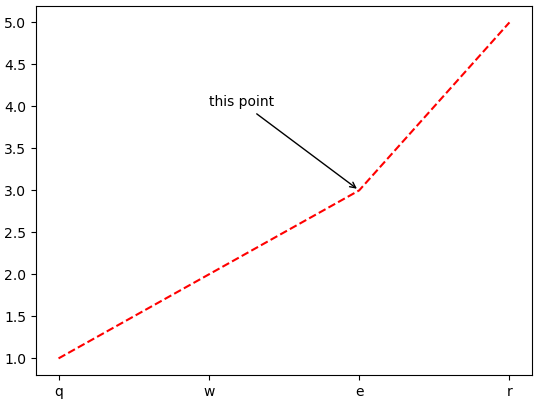
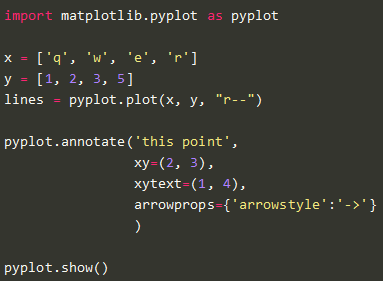
**Oznaczenie strzałką ważnych informacji na wykresie**

Metoda **annotate()** oprócz opisu, dodatkowo umożliwia generowanie strzałki wskazującej konkretny punkt. Zaczynamy podobnie jak w poprzednim przykładzie, jako pierwszy argument podajemy **opis punktu**, który pojawi się na wykresie. Kolejnym **xytext** podajemy jako współrzędne w formie krotki, gdzie ma się rozpoczynać strzałka i jednocześnie pojawić opis wskazywanego punktu. Argument **xy**różni się od standardowego użycia bez strzałki, gdyż w tym wypadku **xy** jest krotką dwóch wartości, współrzędnych punktu, na który ma wskazać strzałka. Ostatnim wymaganym argumentem jest **arrowprops**. Bez niego pojawi się tylko opis, z pominięciem strzałki. Przyjmuje on słownika z parametrami określającymi wygląd strzałki. Słownik może być pusty – wtedy strzałka będzie się składać z samych domyślnych wartości. W poniższym przykładzie zmieniam kolor wypełniania strzałki na zielony.



**Wygląd strzałki**

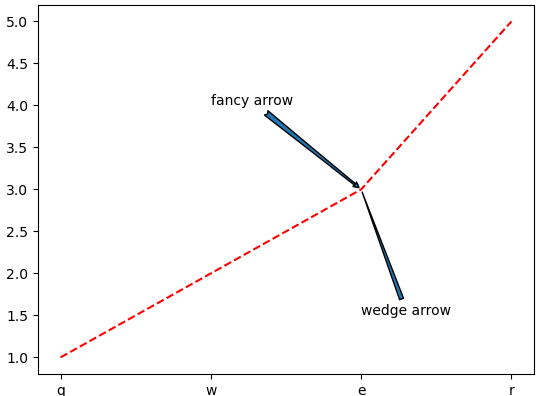
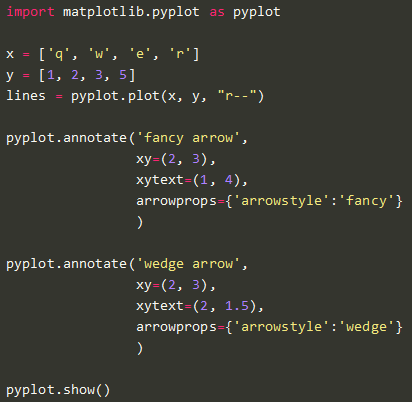
Strzałka,o prócz koloru, może także przyjmować różne style. Aby zmienić wygląd strzałki, użyj klucza **arrowstyle** w słowniku **arrowprops**. Klucz ten może przyjmować różne wartości typu **string.** Jedną z nich może by **‚->’**i oznacza ona cienką strzałkę:



Strzałka posiada wiele możliwych wyglądów:

* „->” wąska strzałka
* „-” linia bez strzałki
* „-[” zbiór punktów, przedział
* „|-|” odcinek
* „<-” strzałka zwrócona w przeciwnym kierunku (na przykład w stronę napisu)
* „<->” strzałka obustronna
* „wedge” strzałka „igła”
* „fancy” strzałka stylizowana

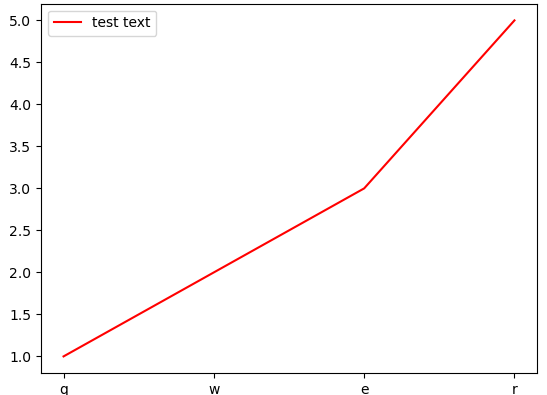
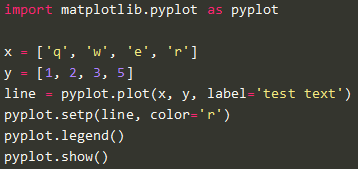
Ostatnie dwa style strzałek (**wedge** i **fancy**) mogą być nie zrozumiałe, więc zaprezentuję je poniżej. Mogę to zrobić jednocześnie, gdyż na jednym wykresie może znajdować się wiele dodatkowych oznaczeń jednocześnie.



**Automatyczne generowanie legendy**

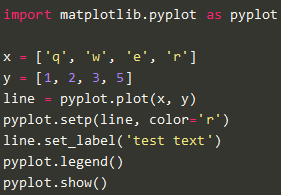
**https://qabrio.pl/python-matplotlib-pyplot-jak-generowac-wykresy-4/**

Najprostszym sposobem na automatyczne wygenerowanie legendy jest po prostu użycie metody **legend**bez żadnych dodatkowych argumentów. Jednak aby legenda mogła się wygenerować, w metodzie**plot()** musi znaleźć się argument **label**, w którym w formie **string’a** podajemy dowolną nazwę wykresu. Dane takie jak opis, kolor czy wygląd wykresu, użyte w przy tworzeniu wykresu zostaną automatycznie umieszczone w legendzie.

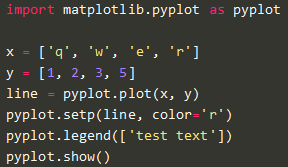


**Opisy w legendzie istniejących wykresów**

Opisy wykresów w legendzie można dodać nie tylko jako argument metody **plot**. Kolejnym sposobem jest użycie metody **set\_label(**) na obiekcie stworzonego wykresu. Wygląd w stosunku do poprzedniego wykresu się nie zmieni, ale warto wiedzieć, że posiada dwie możliwości wprowadzenia opisu:

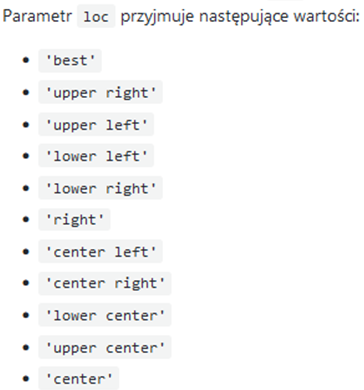


Ostatni ze sposobów labelowania wykresów odbywa się bezpośrednio za pośrednictwem metody **legend**. W liście wpisujemy kolejno nazwy wykresów tworzonych metodami **plot()**:



**Parametry legend**

Metoda **legend** posiada szereg parametrów (argumentów), dzięki którym zmienimy jej wygląd lub miejsce położenia. Poniżej przestawię kilka najbardziej użytecznych:

* loc – (string) – umiejscowienie legendy. Opisać można jednym lub dwoma słowami, opisującymi ustawienie w pionie i/lub w poziomie. Na przykład: right (po prawej i domyślnie u góry), lower center (na dole i po środku)
* 
* fontsize – (string) – wielkość czcionki od najmniejszej do największej: 'xx-small’, 'x-small’, 'small’, 'medium’, 'large’, 'x-large’, 'xx-large’
* title – (string) – tytuł legendy
* shadow – (bool) – Cień ramki
* style – styl linii np. style=”italic”

import matplotlib.pyplot as pyplot

x = ['q', 'w', 'e', 'r']

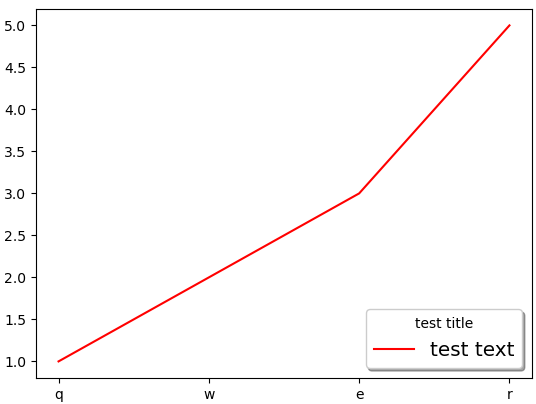
y = [1, 2, 3, 5]

line, = pyplot.plot(x, y)

pyplot.setp(line, color='r')

pyplot.legend(['test text'], loc='lower right', shadow=True, fontsize='x-large', title='test title')

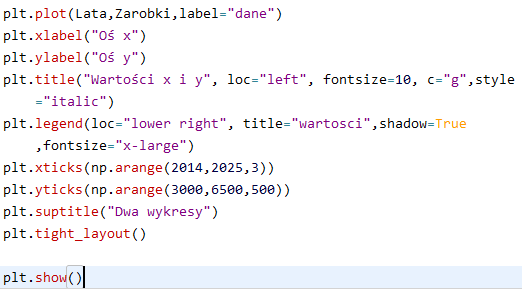
pyplot.show()

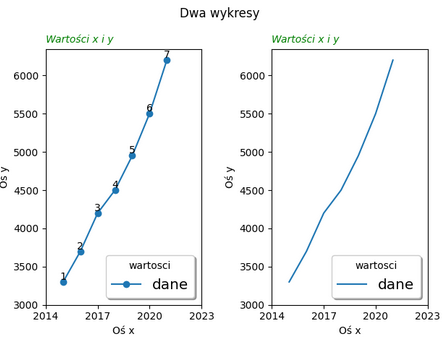


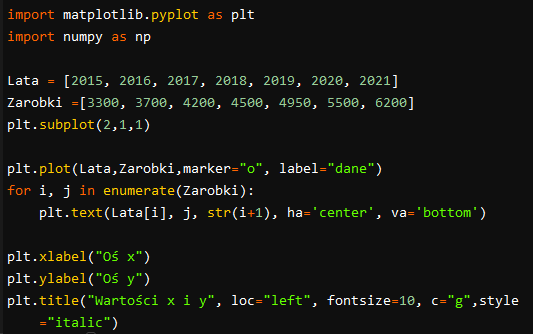
**Dodawanie numerów do markerów na wykresie**

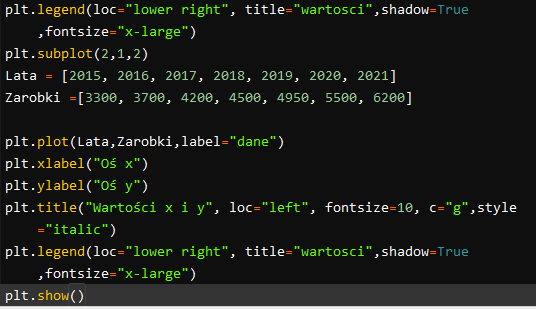


**Zadanie**







**plt.tight\_layout()**

**Funkcja do dopasowania wykresów z opisami do działki**

