

01. Podstawowe funkcje Jupyter Notebooka

Podstawowe funkcje Jupyter Notebooka¶

Jupyter Notebook pozwala tworzyć dokumenty, które łączą ze sobą:

- sformatowany tekst i ilustracje
- kod źródłowy
- wyniki wykonywania kodu

Dzięki tym funkcjom możemy tworzyć w Jupyterze różnego rodzaju dokumenty - samouczki, raporty z badań, "pseudo" aplikacje.

Przestrzeń pracy w Notebookach dzieli się na komórki *cells*. Każda komórka może mieć typ "Code" lub "Markdown". W komórkach pierwszego typu wpisujemy kod Pythona (lub innego obsługiwanego języka) i wykonujemy go, w komórkach drugiego typu możemy umieścić tekst (i "skompilować" go do sformatowanej postaci).

Podstawowe skróty klawiszowe (bez których ani rusz, więcej w "Help" na górze)¶

Te komendy można zastosować tylko po wejściu do trybu poleceń naciskając klawisz **Escape**

- A(bove) - utwórz nową komórkę **powyżej** tej, która jest aktualnie zaznaczona
- B(elow) - utwórz nową komórkę **poniżej** tej, która jest aktualnie zaznaczona
- D(elete) - usuń aktualnie zaznaczoną komórkę
- Alt|Shift|Control + ENTER - wykonaj kod w aktualnie zaznaczonej komórce i utwórz komórkę poniżej/przejdź do komórki poniżej/nic nie rób
- P - otwórz paletę z poleceniami

Formatowanie tekstu¶

Test sformatować możemy za pomocą składni Markdown. Poniżej znajduje się bardzo zwięzłe podsumowanie dostępnych możliwości:

Tak tworzymy nagłówek trzeciego poziomu (liczba # = poziom nagłówka)

Tekst możemy **pochylić**, ****pogrubić**** lub *****pochylić i pogrubić*****.

```
`print('Kod możemy wyświetlić czcionką o stałej szerokości)`
```

```
> Możemy zacytować jakiś fragment tekstu
```

```
#### Listy numerowane
```

1. Łatwo tworzymy listy numerowane
3. Nie musimy przejmować się numeracją!

```
#### Listy nienumerowane
```

- Listy nienumerowane możemy tworząc używając myślników,
* gwiazdek bądź
+ plusów (i możemy je zagnieżdżać)

```
[Łatwo dodać hiperłącza do innych stron](http://bartoszmackiewicz.pl)
```

```
![Logo Jupyter](https://jupyter.org/assets/main-logo.svg "Możemy wyświetlać również obrazki")
```

Tak tworzymy nagłówek trzeciego poziomu (liczba # = poziom nagłówka)¶

Tekst możemy *pochylić*, **pogrubić** lub ***pochylić i pogrubić***.

```
print('Kod możemy wyświetlić czcionką o stałej szerokości)
```

```
    Możemy zacytować jakiś fragment tekstu
```

Listy numerowane¶

1. Łatwo tworzymy listy numerowane
2. Nie musimy przejmować się numeracją!

Listy nienumerowane¶

- Listy nienumerowane możemy tworząc używając myślników,
• gwiazdek bądź
– plusów (i możemy je zagnieżdżać)

Łatwo dodać hiperłącza do innych stron



Wykonywanie kodu¶

Ogólnie rzecz ujmując *output* Notebooków działa tak jak wiersz poleceń Pythona:

In [1]:

```
print('Rezultaty wykonywania kodu widać w polu `output` każdej komórki')
print('Co do zasady możemy wyświetlać tekst drukując go standardowym poleceniem `print`')
```

Rezultaty wykonywania kodu widać w polu `output` każdej komórki
Co do zasady możemy wyświetlać tekst drukując go standardowym poleceniem `print`

Niektóre obiekty są jednak przy wyświetlaniu specjalnie formatowane przez Notebooki (np. obiekt `DataFrame` z pakietu `pandas`)

In [2]:

```
from pandas import DataFrame
data = DataFrame({'imię' : ['Bartosz', 'Kasia', 'Marysia'],
                  'ocena' : [4,3,5]})
```

data

Out[2]:

	imię	ocena
0	Bartosz	4
1	Kasia	3
2	Marysia	5

Formatowanie to działa jednak tylko wówczas, gdy wywołamy dany obiekt przez nazwę zmiennej. Jeśli użyjemy funkcji `print`, dostaniemy standardowy *output*.

In [3]:

```
print(data)
```

```
      imię  ocena
0  Bartosz     4
1   Kasia     3
2  Marysia     5
```

Jeśli korzystamy z jakiegoś pakietu rysującego wykresy, to efekt tego rysowania również możemy bezpośrednio wyświetlić w Notebooku:

In [4]:

```
from matplotlib import pyplot as plt
from scipy.stats import norm
# bez tego polecenia nie zobaczymy wykresów, dostępne są inne niż inline opcje - odsyłam do
%matplotlib inline

plt.hist(norm.rvs(size = 1000)); # losujemy 1000 wartości z rozkładu normalnego i rysujemy i
```

