



# PODSTAWY PROGRAMOWANIA W JĘZYKU PYTHON

## PO 1. ZAJĘCIACH

1. Omawiane zagadnienia:
  - a. Pycharm
    - i. Projekty
    - ii. Pliki
    - iii. ustawienia, wygląd, okienka
    - iv. podpowiedzi (okienko podpowiedzi, podkreślenia)
  - b. Git
    - i. Tworzenie/kopiowanie repozytorium
    - ii. Kurs tworzenia branchy:  
<https://learngitbranching.js.org/>
2. Przydatna dokumentacja:
  - a. [https://www.wikiwand.com/en/File\\_format#/Identifying\\_file\\_type](https://www.wikiwand.com/en/File_format#/Identifying_file_type)
  - b. [https://www.wikiwand.com/en/Integrated\\_development\\_environment](https://www.wikiwand.com/en/Integrated_development_environment)

## PO 2. ZAJĘCIACH

1. Omawiane zagadnienia:
  - a. pamięć komputera
    - i. stos, sarta
    - ii. zarządzanie pamięcią – ręczne (język C),  
automagiczne (Python)
  - b. Zmienne i typy danych w Python
    - i. int, float, string (niezmienna!), bool
    - ii. problemy z typem float
  - c. Działania na typach danych:

- i. funkcje wbudowane typów np. str.capitalize()  
float.is\_integer()
  - ii. formatowanie stringu w print()
  - iii. przyjmowanie danych od użytkownika input()
  - iv. operatory działań logicznych and, or, not
  - v. operator przypisania =, operatory porównania ==, <, >, <=, >=, !=
  - vi. operatory arytmetyczne += itd
  - vii. instrukcje warunkowe if, elif, else
- d. Funkcje dir() oraz help() – uzyskanie dokumentacji offline
- e. Code style – PEP8
- 2. Przydatna dokumentacja:
  - a. <https://docs.python.org/3/library/stdtypes.html#boolean-values>
  - b. <https://docs.python.org/3/library/stdtypes.html#comparisons>
  - c. <https://docs.python.org/3/library/stdtypes.html#boolean-operations-and-or-not>
  - d. <https://docs.python.org/3/library/stdtypes.html#numeric-types-int-float-complex>
  - e. <https://docs.python.org/3/library/stdtypes.html#textseq>
  - f. <https://docs.python.org/3/tutorial/introduction.html#strings>
  - g. <https://docs.python.org/3/library/string.html#formatstrings>
  - h. <https://docs.python.org/3/tutorial/controlflow.html#if-statements>
  - i. [https://docs.python.org/3/reference/compound\\_stmts.html#if](https://docs.python.org/3/reference/compound_stmts.html#if)
  - j. <https://www.python.org/dev/peps/pep-0008/>

## PO 3. ZAJĘCIACH

1. Omawiane zagadnienia:
  - a. Schemat blokowy
  - b. Wizualizacja przebiegu programu w pythontutor.com:  
<http://pythontutor.com/visualize.html#mode=edit>
  - c. Pętla while
  - d. Listy
  - e. Zakres (range)
  - f. Slicing:
    - i. Stringów
    - ii. list
  - g. Pętla for
2. Przydatna dokumentacja:
  - a. <https://docs.python.org/3/library/stdtypes.html#sequence-types-list-tuple-range>
  - b. <https://docs.python.org/3/library/stdtypes.html#range>
  - c. <https://docs.python.org/3/tutorial/controlflow.html#the-range-function>
  - d. [https://docs.python.org/3/reference/compound\\_stmts.html#for](https://docs.python.org/3/reference/compound_stmts.html#for)
  - e. <https://docs.python.org/3/tutorial/controlflow.html#for-statements>
  - f. <https://docs.python.org/3/tutorial/introduction.html#first-steps-towards-programming>
  - g. <https://docs.python.org/3/tutorial/controlflow.html#break-and-continue-statements-and-else-clauses-on-loops>
  - h. <https://docs.python.org/3/tutorial/datastructures.html#looping-techniques>

## OBEJRZYJ SOBIE

1. <https://youtu.be/PZRlIfStY0>
2. <https://youtu.be/MijmeoH9LT4>
3. <https://youtu.be/m2Ux2PnJe6E>
4. <https://youtu.be/UvI-AMAttrvE>
5. <https://youtu.be/1S0aBV-Waao>