

## I Dział instalowanie pycharm Zad 2

Krok 1. Pobranie instalatora pycharm ze strony  
<https://www.jetbrains.com/pycharm/download/#section=windows> oraz wybranie darmowej opcji

### Download PyCharm

Windows

macOS

Linux

#### Professional

Full-featured IDE  
for Python & Web  
development

DOWNLOAD

Free trial

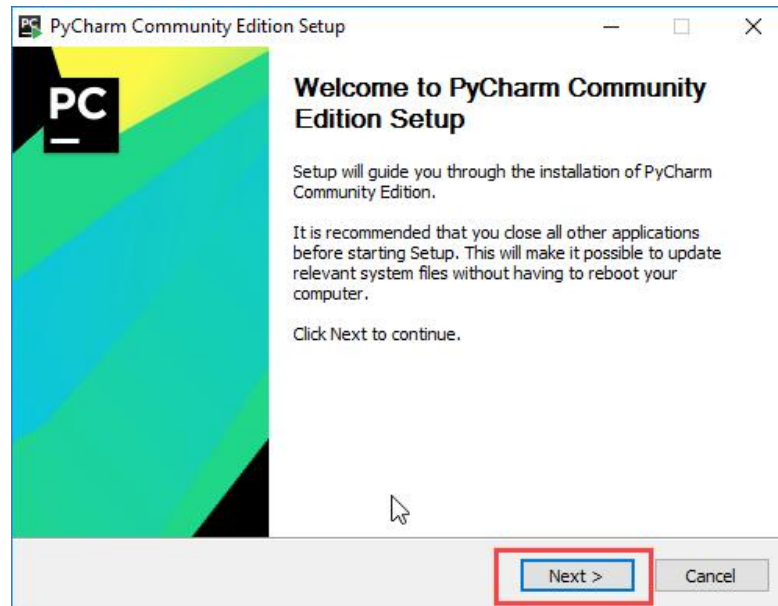
#### Community

Lightweight IDE  
for Python & Scientific  
development

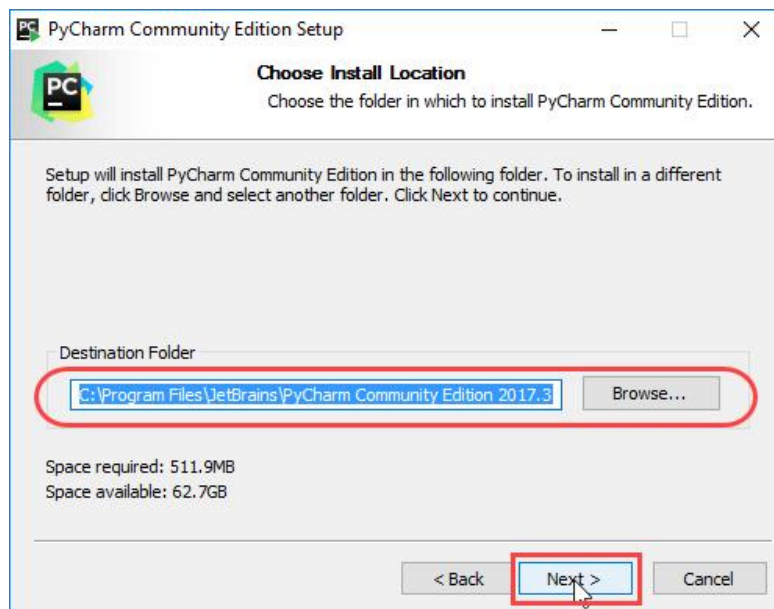
DOWNLOAD

Free, open-source

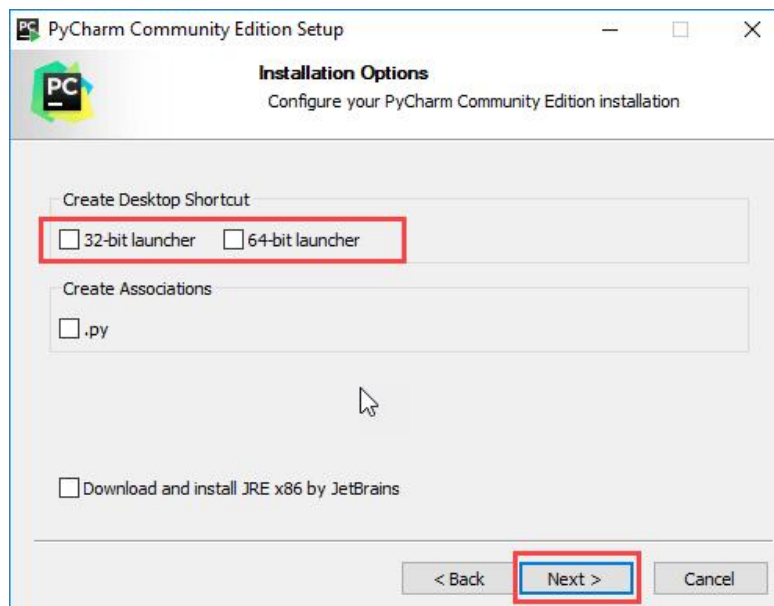
krok 2. Uruchomienie instalatora oraz postepowanie zgodnie ze zdjeciami

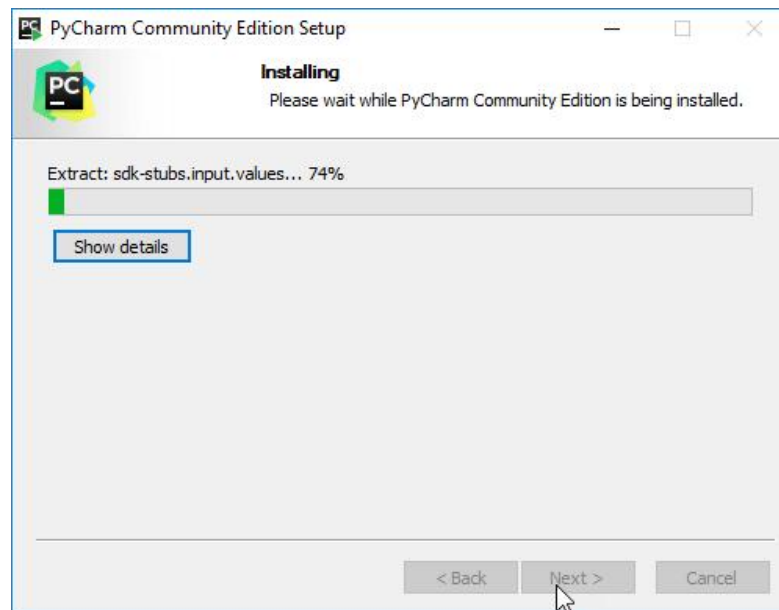
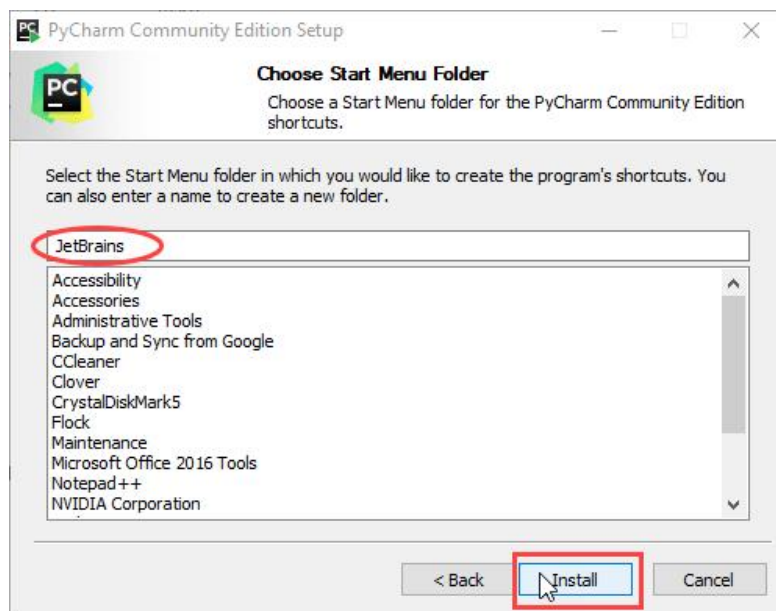


Folder instalacji pycharm moze zostac w domyslej lokacji

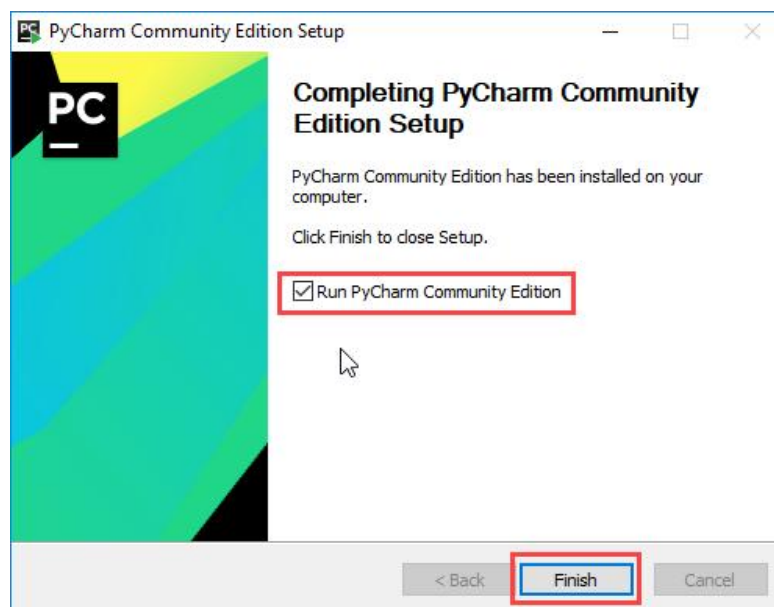


W tym okienku należy wybrać wersję 64bit lub 32bit zależnie od systemu oraz można zaznaczyć "create associations .py" dzięki czemu każdy program z końcówką .py zostanie uruchomiony w pycharmie

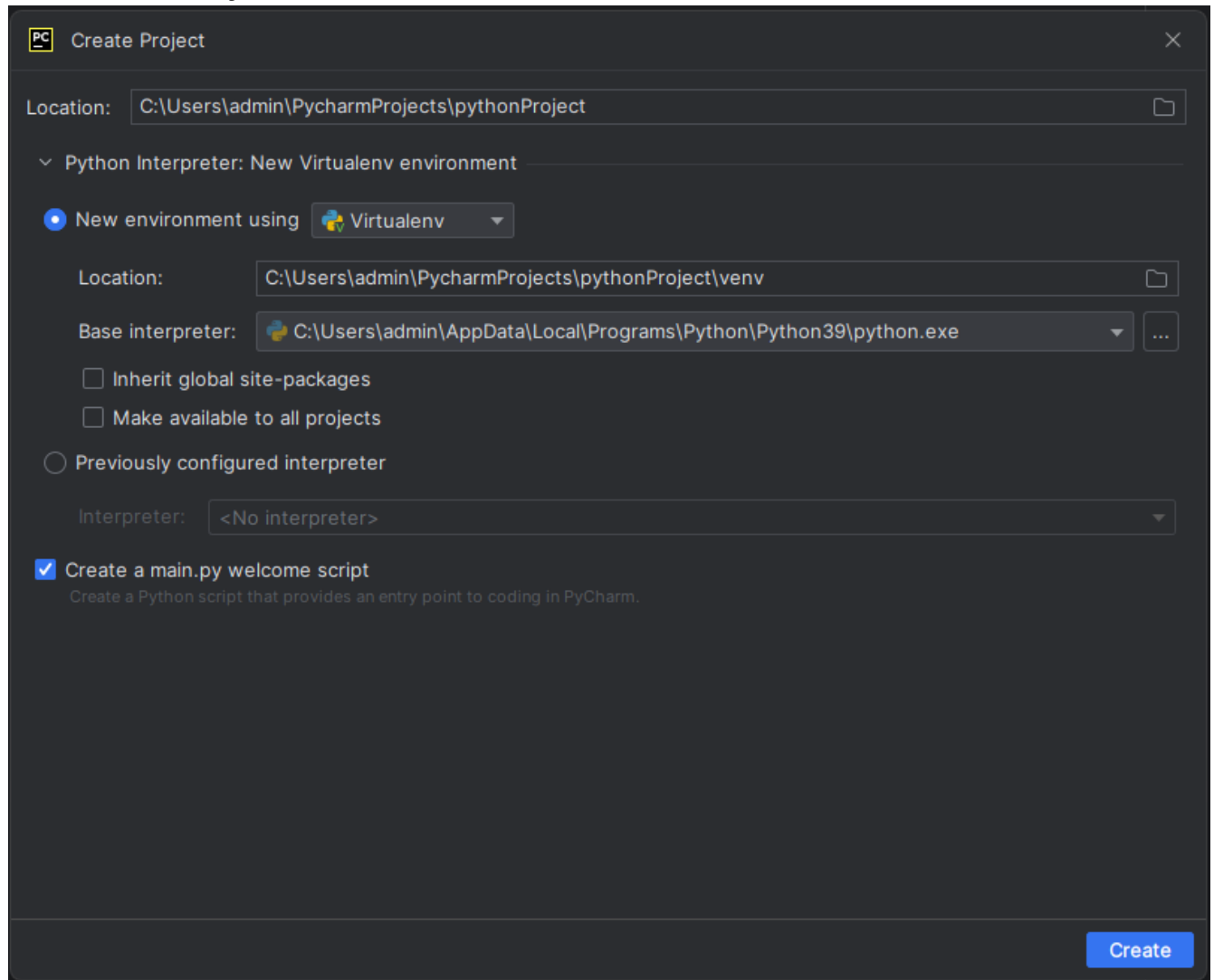




w tym momencie kończymy instalację pycharma  
klikając Finish



W tym miejscu możemy wybrać folder z naszym projektem i naciskamy Create



## Zad 1

Narzędzia programistyczne to oprogramowanie, które ułatwia tworzenie, testowanie i debugowanie kodu. Przykłady narzędzi to edytory tekstu, narzędzia do kontroli wersji, narzędzia do tworzenia dokumentacji, narzędzia do testowania kodu itp.

Środowisko programistyczne (ang. Integrated Development Environment, IDE) to program, który zawiera zestaw narzędzi programistycznych, takich jak edytor kodu, narzędzie do debugowania, narzędzie do tworzenia dokumentacji, narzędzie do kontroli wersji itp. IDE pozwala na stworzenie kompleksowego i zintegrowanego środowiska do pracy nad projektem.

W zależności od wielkości projektu, różne narzędzia i środowiska programistyczne mogą być bardziej lub mniej odpowiednie. Przykładowo, dla małych projektów, edytor tekstu i linia poleceń mogą być wystarczające, natomiast dla dużych projektów, bardziej zaawansowane

programisty. Popularne narzędzia i środowiska programistyczne to Visual Studio, Eclipse, PyCharm, IntelliJ IDEA, Visual Studio Code, Sublime Text. Nie ma jednoznacznej odpowiedzi, które narzędzie/środowisko programistyczne jest najciekawsze/najbardziej wartościowe dla rozwoju projektów programistycznych, bo to zależy od indywidualnych potrzeb i preferencji.

## DZIAŁ 2

### ZAD 1

Typ danych to określenie rodzaju danych, które mogą być przechowywane i przetwarzane przez komputer. Środowisko wywoławcze Python przypisuje i przechowuje typy danych automatycznie, na podstawie wartości, które są przypisane do zmiennej.

Istnieje możliwość zamiany jednego typu danych na inny, za pomocą odpowiednich funkcji lub operatorów.

Proste typy danych zaimplementowane w języku Python to:

liczby całkowite (int) - wartości całkowite, np. -5, 0, 100, itd. Ich rozmiar zajmuje 4 bajty w pamięci operacyjnej

liczby zmiennoprzecinkowe (float) - wartości zmiennoprzecinkowe, np. 1.23, -3.14, itd. Ich rozmiar zajmuje 8 bajtów w pamięci operacyjnej

ciąg znaków (str) - ciąg znaków, np. "hello", "abc123", itd. Ich rozmiar zależy od długości ciągu znaków

wartości logiczne (bool) - wartości true lub false. Ich rozmiar zajmuje 1 bajt w pamięci operacyjnej

listy (list) - posortowana sekwencja obiektów, np. [1, 2, 3, 4]

tuple (tuple) - niezmienna sekwencja obiektów, np. (1, 2, 3, 4)

słowniki (dict) - kolekcja par klucz-wartość

zbiory (set) - kolekcja unikalnych elementów

Rozmiar zajmowany przez te typy danych jest zmienny i zależy od ilości i wartości danych.

Link do kodu z zadania

[https://github.com/frozuu/TEB\\_PD/blob/main/main.py](https://github.com/frozuu/TEB_PD/blob/main/main.py)

```
PC File Edit View Navigate Code Refactor Run Tools VCS Window Help
DK Version control
main.py x
1
2 x = float(input("Podaj pierwszą liczbę: "))
3 y = float(input("Podaj drugą liczbę: "))
4
5 operacja = input("Wybierz operację arytmetyczną (+, -, *, /, **, %): ")
6
7
8 if operacja == "+":
9     wynik = x + y
10 elif operacja == "-":
11     wynik = x - y
12 elif operacja == "*":
13     wynik = x * y
14 elif operacja == "/":
15     wynik = x / y
16 elif operacja == "**":
17     wynik = x ** y
18 elif operacja == "%":
19     wynik = x % y
20 else:
21     wynik = "Nieprawidłowa operacja"
22
23 print(x, operacja, y, "=", wynik)
```

```
elif operacja == "**"
Run
C:\Users\admin\PycharmProjects\DK\venv\Scripts\python.exe C:\Users\admin\PycharmProjects\DK\main.py
Podaj pierwszą liczbę: 2.21
Podaj drugą liczbę: 3.7
Wybierz operację arytmetyczną (+, -, *, /, **, %): **
2.21 ** 3.7 = 18.804050663645274
Process finished with exit code 0
|
```

## ZAD 1

W języku Python istnieją różne rodzaje rozwiązań decyzyjnych, w tym:

Instrukcje warunkowe: if, elif, else

Pętle: for i while

Instrukcje skoku: break, continue, return

Te rozwiązania pozwalają na wykonywanie różnych operacji w zależności od określonych warunków lub na powtarzanie operacji w pętli.

W języku Python nie ma natomiast natywnej implementacji rozwiązań takich jak case/switch lub goto. Mimo to, rozwiązania te mogłyby być zaimplementowane za pomocą innych instrukcji decyzyjnych, takich jak instrukcje warunkowe i pętle.