

W ramach wstępu proszę o przypomnienie niektórym Waszym kolegom/koleżankom o oddawanie/przesyłanie prac. Wysyłacie je ZAWSZE na tego samego maila- [dawidkoch.szkoła@gmail.com](mailto:dawidkoch.szkoła@gmail.com) W razie problemów czy innych spraw, które uniemożliwiają Wam oddanie pracy, proszę o kontakt przez tego maila. Gdy prace nie będą oddawane w terminie (i tak staram się zadawać stosunkowo mało, by Was nie przeładować obowiązkami) o sytuacji będą informowani Wasi wychowawcy.

27.05.2020r.

## Podstawy pracy w środowisku Python

**Z tego tematu nie ma pracy domowej**

Linki do tematu:

<https://www.python.org/> Tutaj w sekcji downloads macie możliwość ściągnięcia najnowszej wersji środowiska Python w wersji 3.8.3

<https://ideone.com/> Tutaj jest serwis, który służy do programowania online. Polecam założenie tam konta. Możecie programować w ponad 40 różnych językach.

Na platformie Udemy.com znajduje się bardzo duża liczba kursów programowania Python. W tym momencie są **one płatne**, ale bardzo często są promocje/kody i wtedy szkolenia są darmowe. Polecam śledzić.

Ciekawą propozycją dla osób, które nigdy nie miały do czynienia z tym językiem programowania jest- <https://ksplus.komputerswiat.pl/sklep#python-kurs-programowania-na-prostych-przykladach-biblioteczka-3-2019> To oczywiście jeżeli jesteście zainteresowani nauką tego języka również później i chcecie rozwinąć umiejętności, które zdobędziecie na lekcjach. Absolutnie nie ma konieczności kupowania jakichkolwiek książek czy szkoleń.

<https://www.learnpython.org/pl/> Tylko, że tutaj działają na wersji 2.7



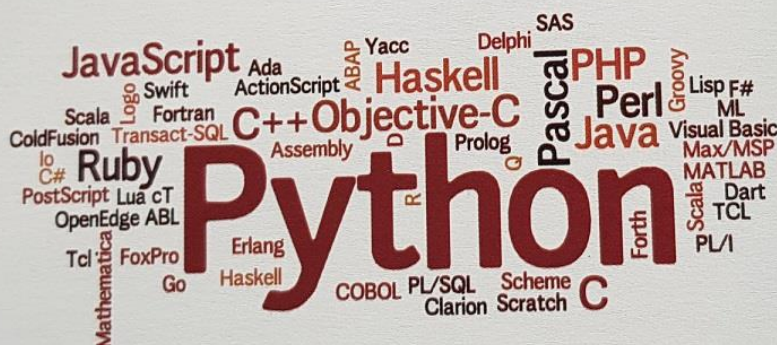
## 14. Podstawy pracy w środowisku Python

- Wprowadzenie do programowania w języku Python
- Praca w trybie bezpośrednim i w pliku
- Definiowanie prostych funkcji

### ZADANIE NA START

#### Programowanie w Pythonie

Python to język interpretowany, którego używają zarówno początkujący programiści, jak i profesjonaliści. Doskonale sprawdza się jako narzędzie do tworzenia bardziej lub mniej złożonych aplikacji desktopowych, pracujących pod kontrolą systemów operacyjnych Windows, Linux czy OS, dynamicznych stron internetowych oraz zaawansowanych aplikacji webowych. Na przykład Google wykorzystuje Pythona w wyszukiwarce, amerykańska National Security Agency – w kryptografii oraz analizach wywiadowczych, firmy Intel, Hewlett-Packard czy IBM – w testowaniu urządzeń. Znajdź trzy aplikacje napisane w języku Python.



### ABC JĘZYKA PYTHON

W systemie Windows pracę w środowisku Python należy rozpocząć od zainstalowania tego środowiska na komputerze. W Ubuntu Linuksie język ten stanowi standardowy element systemu. Środowisko Python w wersji 3.\* odpowiedniej dla danego systemu należy pobrać z serwisu <https://python.org>. Oprócz rekomendowanej opcji instalacji programu dla wszystkich użytkowników warto zaznaczyć opcję umożliwiającą uruchamianie skryptów w wierszu polecenia z dowolnej lokalizacji.

### POSZUKAJ W INTERNECIE

Poszukaj w internecie wywiadów z twórcą Pythona Guidem van Rossumem, aby dowiedzieć się więcej o nim i jego filozofii oraz o pochodzeniu nazwy języka.



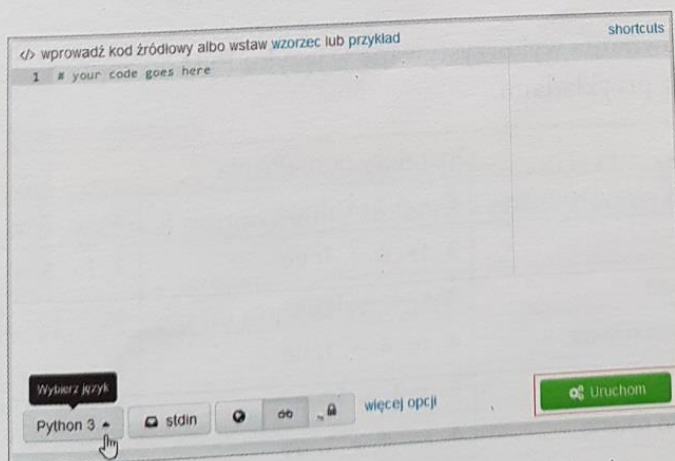
Skrypty Pythona można zapisywać w dowolnym edytorze tekstu, ale wygodniej jest używać zintegrowanego środowiska deweloperskiego (IDE), zawierającego różne funkcje, np. kolorowania składni. Osobom początkującym wystarczy **IDLE** – prosty edytor stanowiący część domyślnego pakietu instalacyjnego.

Po uruchomieniu IDLE otworzy się okno **Shell** – powłoka systemowa działająca jako interpreter, który wypisuje wynik działania skryptu zapisanego w nim w trybie bezpośrednim albo programu utworzonego wcześniej w edytorze. Trzy nawiasy ostrokatne, tzw. znak zachęty, oznaczają, że interpreter czeka na wprowadzenie polecenia.

```
Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16)
[MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.
>>>
```

Rys. 1. Okno Shell edytora IDLE

W ostatnich latach coraz więcej osób pracuje w chmurze. Programowanie online umożliwia m.in. serwis **ideone.com** stworzony przez polską firmę z Trójmiasta. Serwis ten obsługuje ponad 40 języków programowania. Warto założyć w nim konto i ustawić domyślny język programowania, język interfejsu oraz włączyć podświetlanie składni, a także używać notatek do opisu programu oraz etykiet do organizacji swoich kodów. Wszystkie skrypty programów zapisywane są w chmurze, ale można je pobrać na dysk i skorzystać z nich ponownie.



Rys. 2. W serwisie **ideone.com** kursor wskazuje wybór języka, a czerwona ramka – uruchomienie programu

**OPERATORY**

Python obsługuje operatory arytmetyczne i porównania. Do siedmiu podstawowych **operatorów arytmetycznych** należą: dodawanie (+), odejmowanie (-), mnożenie (\*), potęgowanie (\*\*), dzielenie rzeczywiste (/), dzielenie całkowite (//) oraz reszta z dzielenia całkowitego (%). W przypadku dzielenia rzeczywistego iloraz jest liczbą rzeczywistą, dlatego wynikiem dzielenia 4 przez 2 jest dla Pythona liczba 2.0 (część całkowitą w tym języku zawsze oddziela od części dziesiętnej kropka). W przypadku dzielenia całkowitego wynikiem może być tylko liczba całkowita – iloraz całkowity dwóch liczb bez reszty.

```

Python 3.7.1 Shell
File Edit Shell Debug
>>> 34 / 2
17.0
>>> 35 // 2
17
  
```

Rys. 3. Dzielenie rzeczywiste i całkowite w trybie bezpośrednim IDLE

Zastanów się, jak za pomocą potęgowania obliczyć pierwiastek kwadratowy z liczby.

**Ćwiczenie 1. Kupowanie obcej waluty**

Oblicz, ile musisz zapłacić w złotychkach za 126 euro (aktualny kurs wymiany sprawdź na stronie NBP).

- Zapisz odpowiedni wzór: przemnoż daną kwotę przez aktualny kurs wymiany (pamiętaj o zastąpieniu przecinka kropką) – np. dla 1 euro równego 4,20 zł kod przyjmuje poniższą postać:

$$126 * 4.2$$

**Operatory porównania** wykorzystywane w języku Python zaprezentowano w poniższej tabeli wraz z przykładami.

Operatory porównania			
==	równe	3 == 3 → True	4 == 3 → False
!=	różne	3 != 2 → True	3 != 3 → False
<	mniejsze	3 < 4 → True	4 < 3 → False
<=	mniejsze równe	4 <= 4 → True	4 <= 3 → False
>	większe	4 > 3 → True	3 > 4 → False
>=	większe równe	4 >= 4 → True	4 >= 5 → False



Wynikami porównań są wartości logiczne **True** (prawda) lub **False** (fałsz). Należy pamiętać, że w porównaniu znak równości w Pythonie jest podwójny. Pojedynczy znak `=` to operator przypisania, za pomocą którego przypisuje się zmiennej pewną wartość.

### Ćwiczenie 2. Porównanie cen wyrażonych w różnych walutach

Porównaj ceny tych samych butów, które w Grecji kosztują 49 euro, a w Polsce 200 zł (aktualny kurs wymiany sprawdź na stronie NBP).

- **Sposób 1** – przelicz cenę butów w Grecji na złotówki, a potem porównaj obie wartości.
- **Sposób 2** – zapisz oba działania w jednym wierszu z wykorzystaniem operatorów arytmetycznych i porównania. Gdy 1 euro = 4,20 zł, kod w IDLE może przyjąć poniższą postać:

```
>>> 49 * 4.2 >= 200
True
```

### ZMIENNE

Zmienne przechowują wartości określonego typu, m.in. liczby całkowite, liczby rzeczywiste, wartości logiczne i ciągi znaków. Każda zmienna w Pythonie powinna mieć krótką i znaczącą nazwę (bez polskich liter i spacji), np. `n` albo **liczba**. Należy przy tym pamiętać, że **wielkość liter jest istotna** – `N` i `n` to różne zmienne. Aby nadać zmiennej wartość, należy zastosować operator przypisania `=`. W każdej chwili można z tej wartości skorzystać lub ją modyfikować.

```
>>> x = 34
>>> y = 66
>>> x + y
100
```

Rys. 4. Obliczanie sumy zmiennych całkowitych `x` oraz `y`

### Ćwiczenie 3. Średni dystans

W czasie wakacji uczniowie na obozie wędrownym przeszli w ciągu kolejnych trzech dni 12, 14 i 18 km. Zdefiniuj odpowiednie zmienne i oblicz średnią długość pokonywanych tras.

- Zdefiniuj zmienne i przypisz im podane wartości.

```
a = 12
b = 14
c = 18
```

- Zapisz odpowiedni wzór, za pomocą którego wyznaczysz średnią podanych wartości.

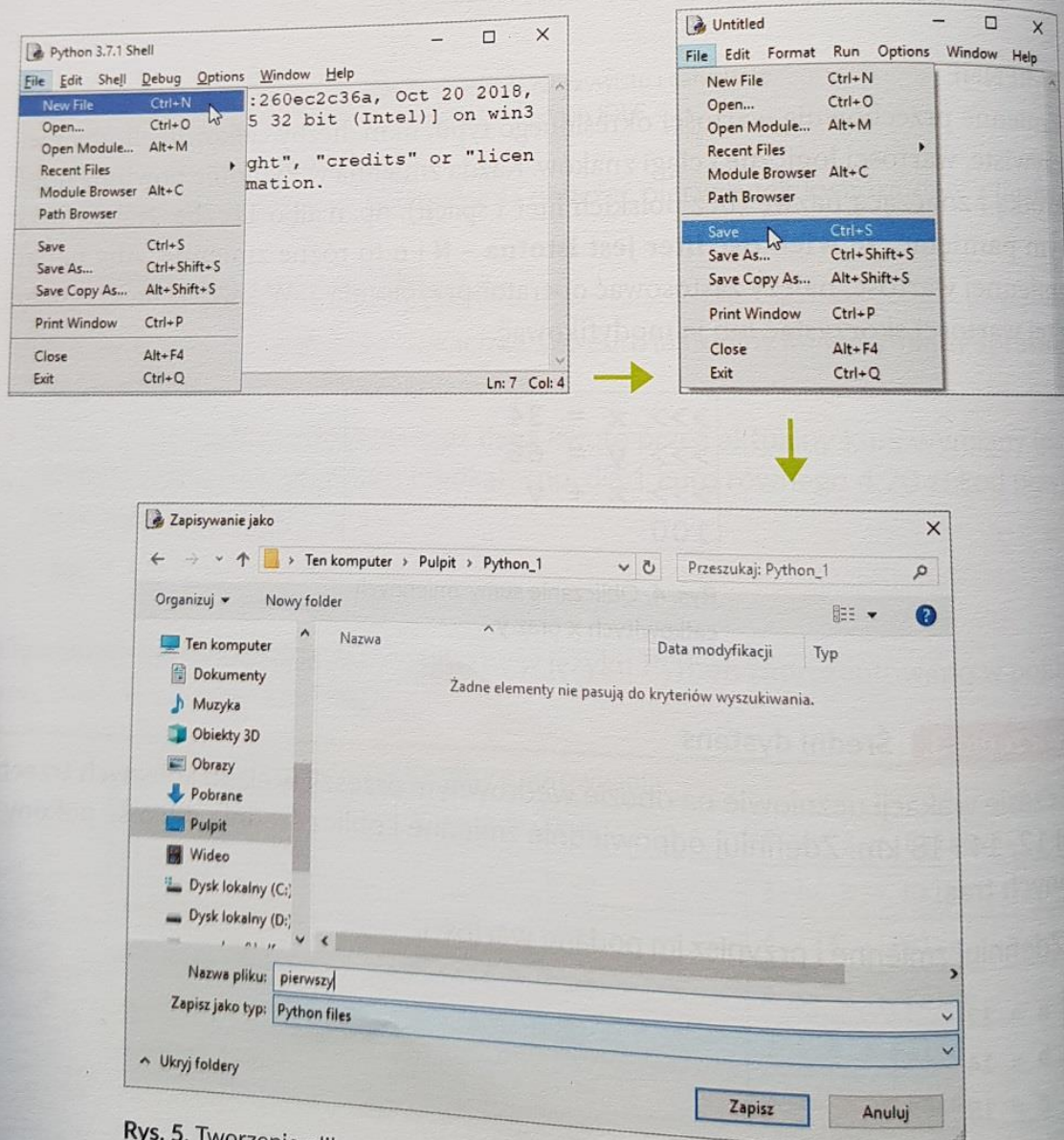
$$(a + b + c)/3$$

- Sprawdź odpowiedź.

### PRACA W EDYTORZE

Kodowanie polega na pisaniu poleceń zgodnie z regułami obowiązującymi dla danego języka. Te polecenia w dalszej kolejności są analizowane przez interpreter, a następnie wykonywane przez komputer.

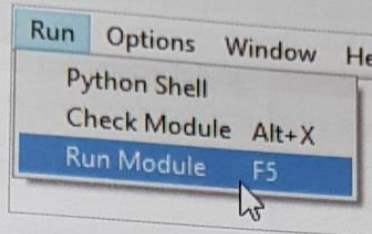
Aby rozpocząć pracę w IDLE, należy kliknąć **File** na pasku menu i wybrać opcję **New File**. Otworzy się niezatytułowane okno – edytor. Następnie w oknie edytora należy wybrać **File** i opcję **Save** lub **Save as**, nazwać plik i zapisać go w preferowanej lokalizacji. W ten sposób powstanie plik z rozszerzeniem PY.



Rys. 5. Tworzenie pliku o nazwie pierwszy w IDLE

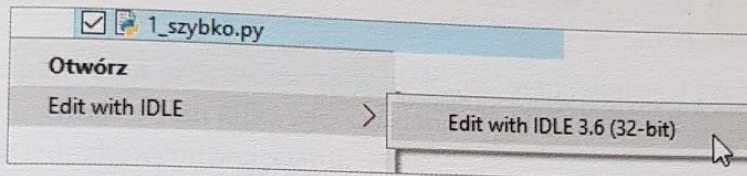


Aby uruchomić program, należy wybrać na pasku menu **Run** → **Run Module**. Gdy pojawi się okno z pytaniem, czy zapisać zmiany, trzeba kliknąć **OK**. Warto pamiętać, że każde uruchomienie pliku Pythona powoduje jego zapisanie.



Rys. 6. Uruchamianie programu w IDLE

Aby ponownie edytować utworzony plik, trzeba go zaznaczyć, a następnie z menu podręcznego wybrać polecenie **Edit with IDLE**.



Rys. 7. Otwieranie pliku PY do edycji w IDLE

Aby rozpocząć pracę w edytorze w serwisie **ideone.com**, wystarczy w przeglądarce wpisać adres serwisu i po wgraniu strony można zacząć pracę. Aby uruchomić program, należy wybrać opcję **Uruchom**. Wszystkie zapisane programy znajdują się w sekcji **Moje kody**. Każdy z nich można ponownie otworzyć, duplikować lub pobrać na dysk.

#### WIEM WIĘCEJ

Podczas pisania programu należy dbać o odpowiednie standardy kodowania. Z regułami i zaleceniami dotyczącymi formatowania kodu w Pythonie (*Style Guide for Python Code*) można się zapoznać na stronie [www.python.org/dev/peps/pep-0008](http://www.python.org/dev/peps/pep-0008).

#### POLECENIE PRINT

Instrukcja **print** służy do pisania znaków. Możesz wpisać tekst (ciąg znaków), liczbę albo zmienną.

#### ZAPAMIĘTAJ!

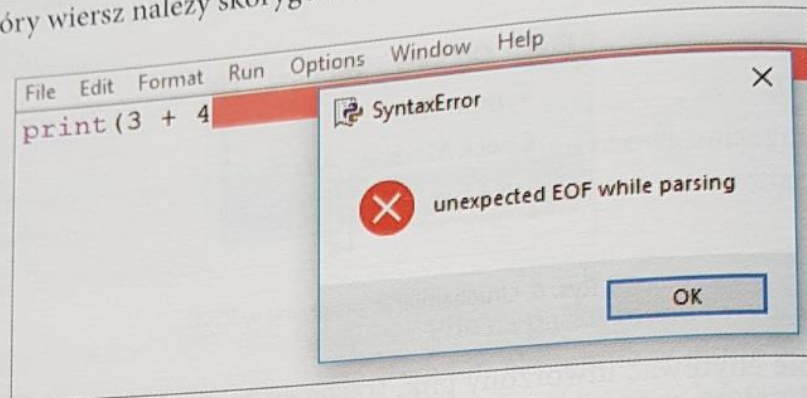
```
print("<ciąg znaków>")
```

```
print(<liczba>)
```

```
print(<zmienna>)
```

Aby dodać pusty wiersz, można skorzystać z polecenia `print("")`.

Jeśli do zapisu wkładnie się błąd składni, pojawi się komunikat **SyntaxError** – kolorowy pasek w IDLE lub szczegółowa informacja o kompilacji w serwisie **ideone.com** wskaże, który wiersz należy skorygować.



edytuj klonuj pobierz

1. `print(3 + 4`

Błąd kompilacji #stdin błąd kompilacji #stdout 0s OKB

stdin  
Standard input is empty

informacja o kompilacji

Traceback (most recent call last):  
 File "/usr/lib/python3.5/py\_compile.py", line 125, in compile  
     \_optimize=optimize)  
 File "<frozen importlib.\_bootstrap\_external>", line 735, in source\_to\_code  
 File "<frozen importlib.\_bootstrap>", line 222, in \_call\_with\_frames\_removed  
 File "./prog.py", line 1  
     print(3 + 4  
           ^  
 SyntaxError: unexpected EOF while parsing

During handling of the above exception, another exception occurred:

Traceback (most recent call last):  
 File "<string>", line 1, in <module>  
 File "/usr/lib/python3.5/py\_compile.py", line 129, in compile  
     raise py\_exc  
 py\_compile.PyCompileError: File "./prog.py", line 1  
     print(3 + 4  
           ^  
 SyntaxError: unexpected EOF while parsing

stdout  
Standard output is empty

Rys. 8. Przykładowy komunikat informujący o braku nawiasu zamykającego w IDLE i w serwisie ideone.com



## DEFINIOWANIE FUNKCJI

Funkcje pozwalają podzielić większy problem na mniejsze, względnie niezależne części, a także przyspieszają i ułatwiają pisanie programu, który staje się dzięki nim bardziej zwięzły i przejrzysty. Definicję funkcji zaczyna się od słowa kluczowego **def**, następnie podaje się nazwę funkcji (bez polskich znaków, spacji) i wstawia nawiasy, w których można umieścić parametr lub parametry lub nie umieszczać niczego (funkcja bez parametru). Wiersz należy zakończyć dwukropkiem, który zagwarantuje, że po naciśnięciu klawisza **Enter** w następnym wierszu powstanie wcięcie. Kolejne wiersze kodu zawierają instrukcje do wykonania. Polecenie **return**, które przekazuje szukaną wartość, kończy działanie funkcji.

### ZAPAMIĘTAJ!

```
def <nazwa funkcji>(<parametr>):
    <instrukcje>
```

W Pythonie polecenia grupowane są w bloki, które uzyskuje się przez **indentację**, tj. wcięcia w wierszach. Jeśli kilka wierszy stanowi jeden blok, to wszystkie muszą być poprzedzone jednakowymi wcięciami.

```
1. def pole():
2.     a = 5
3.     return a ** 2
```

→ 25

Rys. 9. Funkcja `pole` i jej wynik

## Ćwiczenie 4. Przeliczanie prędkości

Samolot z Warszawy do Rzymu pokonuje dystans 1320 km. Przeanalizuj dane w tabeli i zdefiniuj funkcję `jak_szybko(t)`, której parametrem jest czas przelotu samolotu w godzinach, a wynikiem średnia prędkość tego samolotu.

Wywołanie funkcji	Wynik
<code>jak_szybko(2.5)</code>	<b>528.0</b>
<code>jak_szybko(3)</code>	<b>440.0</b>

- Aby wyznaczyć średnią prędkość, należy podzielić drogę przez czas.
- Zdefiniuj funkcję `jak_szybko(t)`, np.

```
1. def jak_szybko(t):
2.     s = 1320
3.     return s / t
```

- Sprawdź działanie skryptu z różnymi danymi.
- W IDLE wybierz z menu Run → Run Module, kliknij OK, a następnie w oknie Shell napisz wywołanie funkcji z odpowiednim parametrem w nawiasie.

```
>>> jak_szybko(2.5)
528.0
>>> jak_szybko(3)
440.0
```

- W serwisie **ideone.com** pod definicją funkcji dodaj polecenie **print** i umieść wywołanie funkcji z odpowiednim parametrem w nawiasie.

```
</> wprowadź kod źródłowy albo wstaw wzorec lub przykład shortcuts
def jak_szybko(t):
    s = 1320
    return s / t

print(jak_szybko(2.5))
print(jak_szybko(3))
```

Uruchom program (Ctrl+Enter)

Python 3 | stdin | stdout | clear | więcej opcji | **Uruchom**



dane wejściowe | Wyjście | clear the output | kolorowanie składni

Sukces #stdin #stdout 0s 15232KB

528

440

### Ćwiczenie 5. Przeliczanie jednostek

Mila lądowa, jednostka długości stosowana w krajach anglosaskich, równa jest 1,609 km. Przeanalizuj dane w tabeli i zdefiniuj funkcję **zamiana(m)**, której parametrem jest odległość w milach lądowych, a wynikiem ta sama odległość w kilometrach.

Wywołanie funkcji	Wynik
<b>zamiana(10)</b>	<b>16.09</b>
<b>zamiana(12)</b>	<b>19.308</b>

- Aby zamienić mile na kilometry, należy pomnożyć odległość w milach przez 1,609.



- Zdefiniuj funkcję `zamiana(m)`, np.

```
1. def zamiana(m):
2.     return m * 1.609
```

- Sprawdź działanie skryptu z różnymi danymi.

### Ćwiczenie 6. O której godzinie pociąg dotrze do celu

Nocny pociąg z Krakowa do Paryża wyjeżdża ze stacji o godzinie 21. Przeanalizuj dane w tabeli i zdefiniuj funkcję `godzina(n)`, której parametrem jest czas przejazdu pociągu w godzinach, a wynikiem godzina dojazdu do celu.

Wywołanie funkcji	Wynik
<code>godzina(14)</code>	11
<code>godzina(20)</code>	17

- $21 + 14 = 35$ ,  $21 + 20 = 41$ . Doba trwa 24 godziny, dlatego aby obliczyć czas przyjazdu pociągu, należy wyznaczyć resztę z dzielenia wyniku dodawania przez 24.
- Zdefiniuj funkcję `godzina(n)` – wykorzystaj do obliczeń resztę z dzielenia całkowitego.

```
1. def godzina(n):
2.     return (21 + n) % 24
```

- Sprawdź działanie skryptu z różnymi danymi. Jakich danych warto użyć?

#### ZAPAMIĘTAJ!

Ważną umiejętnością programistyczną jest zapis w języku formalnym tego, co można wyliczyć bez komputera. Podczas rozwiązywania zadań staraj się najpierw zrozumieć problem, przeanalizuj go na kilku przykładach, potem zaproponuj algorytm i zapisz go w języku programowania. Nie wolno zapomnieć o testowaniu. Wypracowanie takiej metody postępowania ułatwia rozwiązanie różnych problemów, nie tylko tych wywodzących się z informatyki.



### Ćwiczenie 7. Która godzina będzie w Nowym Jorku

Sprawdź, jaka strefa czasowa obowiązuje w Nowym Jorku, a następnie przeanalizuj dane w tabeli i zdefiniuj funkcję `godz_nj(g)`, której parametrem jest pełna godzina w Warszawie, a wynikiem aktualna pełna godzina w Nowym Jorku.

Wywołanie funkcji	Wynik
<code>godz_nj(18)</code>	12
<code>godz_nj(2)</code>	20

## PODSUMOWANIE

- Python to język interpretowany o przejrzystej składni, za pomocą którego można pisać mniej i bardziej skomplikowane programy. Jego skrypty są analizowane przez interpreter, a następnie wykonywane przez komputer.
- Podstawowe operatory arytmetyczne i porównania są zbliżone do używanych w języku naturalnym, dlatego korzystanie z nich szybko staje się intuicyjne.
- Polecenia grupowane są w bloki, które uzyskuje się przez wcięcia w wierszach (indentację).
- Język Python rozróżnia małe i wielkie litery.
- Rozwiązywany problem warto podzielić na mniejsze problemy, które można niezależnie rozwiązać przez definiowanie funkcji. Gdy problem jest prosty, wystarczy jedna funkcja.
- Definicję funkcji zaczyna się od słowa kluczowego **def**, następnie podaje się nazwę funkcji i wstawia nawiasy, w których można umieścić parametr. Deklarację należy zakończyć dwukropkiem. Kolejne wiersze kodu zawierają instrukcje do wykonania.
- Funkcje mogą przekazywać wynik (polecenie **return**) lub powodować określony skutek na ekranie (instrukcja **print**).
- Ważną umiejętnością programistyczną jest zapis w języku formalnym tego, co można wyliczyć bez komputera.