

Klasa VIII – materiał do sprawdzianu:

Na zajęciach z podstaw programowania wielokrotnie podkreślaliśmy zalety języka **Python**, mówiliśmy, czym jest środowisko programowania, a nasze ćwiczenia z kodowania realizowaliśmy w środowisku **Thonny**. Całość niezbędnych informacji na ten temat znajdziecie poniżej, w dziale „**Uzupełnienie materiałów znajdujących się w podręczniku**”. Materiał wyróżniony **kolorem zielonym** nie jest obowiązkowy.

I. Uzupełnienie materiałów znajdujących się w podręczniku:

Język Python:

Python to wszechstronny, **interpretowany** język programowania, który cechuje się prostą składnią i czytelnością kodu, co czyni go idealnym zarówno dla początkujących, jak i zaawansowanych programistów. Został stworzony przez **Guido van Rossuma w 1991 roku** i od tego czasu zdobył ogromną popularność.

Python znajduje zastosowanie w wielu dziedzinach, takich jak:

- **Sztuczna inteligencja i uczenie maszynowe,**
- **Tworzenie stron internetowych,**
- **Automatyzacja zadań,**
- **Aplikacje desktopowe i gry.**

Jego zalety to m.in. duża społeczność, otwartość na różne platformy (kod Pythona możemy uruchamiać na różnym sprzęcie i różnych systemach operacyjnych), możliwość integracji z innymi językami oraz ciągły rozwój. Python jest obecnie jednym z najpopularniejszych języków programowania na świecie.

Język interpretowany:

Język interpretowany to taki język programowania, w którym pisany przez programistę kod źródłowy jest wykonywany bezpośrednio przez specjalny program zwany **interpreterem**, linia po linii. Dzięki temu programy w językach interpretowanych, takich jak Python czy JavaScript, są łatwiejsze do uruchomienia i wyszukiwania błędów, ale ich działanie może być wolniejsze niż w przypadku języków kompilowanych, takich jak np. C++.

Środowisko programowania:

Środowisko programowania to zestaw programów, które umożliwiają tworzenie, testowanie, usuwanie błędów i zarządzanie tworzonym przez programistę kodem źródłowym programu. W skład środowiska programowania wchodzi zazwyczaj:

1. **Edytor kodu** – narzędzie do pisania kodu, często oferujące funkcje takie jak kolorowanie składni, podpowiedzi kodu czy automatyczne uzupełnianie.
2. **Kompilator lub interpreter** – narzędzia tłumaczące kod źródłowy na język zrozumiały dla komputera. **Interpreter** działa "na żywo" (np. w Pythonie), a **kompilator** tworzy pliki wykonywalne (np. w C++) które następnie można uruchamiać.

Na naszych zajęciach używaliśmy środowiska programowania **Thonny**:

Thonny to łatwe w obsłudze środowisko programistyczne (IDE) zaprojektowane z myślą o nauce programowania w języku Python. Jest szczególnie popularne wśród początkujących dzięki swojej prostocie i intuicyjnemu interfejsowi.

Główne cechy Thonny:

1. **Prosty interfejs** – minimalizm pozwala skupić się na kodzie, a nie na skomplikowanych funkcjach.
2. **Wbudowany Python** – Thonny instaluje własną wersję Pythona, dzięki czemu nie wymaga dodatkowej konfiguracji.
3. **Debugowanie (wyszukiwanie błędów) krok po kroku** – umożliwia śledzenie wykonywania programu linia po linii, co ułatwia zrozumienie działania kodu.
4. **Wsparcie dla mikrokontrolerów** – **Thonny** obsługuje platformy takie jak **MicroPython** i **Raspberry Pi Pico**, co czyni go świetnym narzędziem do nauki programowania sprzętu.

Thonny jest idealnym wyborem dla początkujących programistów oraz osób uczących się Pythona w środowisku edukacyjnym.

II. Materiał znajdujący się w podręczniku:

- **rozdział 2.1** (Rysuj z żółwiem): co to jest **składnia**, czym są **moduły** w Pythonie, jak w kodzie zaznaczamy **komentarze**, **instrukcje grafiki żółwia** (tabela, str. 52), **polecenia potrzebne do rysowania kolorami** (str. 54). Umiejętności praktyczne: napisanie kodu rysującego pusty i zamalowany kwadrat (z użyciem instrukcji iteracyjnej **for ... in range ...**) - kody na str. 55.
- **rozdział 2.2** (Fantazyjne posadzki): **definiowanie funkcji bez parametru** (str. 58), **definiowanie funkcji z parametrem** (str. 60). Umiejętności praktyczne: napisanie kodów zawierających przykłady definiowania funkcji bez parametru (str. 58 i 59) i z parametrem (str. 61 - kod z rys. 5, **kod z rys. 6 – nieobowiązkowy**)
- **rozdział 2.4** (Operacje arytmetyczne i zmienne): **operatory arytmetyczne** (str. 71), **pojęcie i typy zmiennych** (str. 72)
- **rozdział 2.5** (Pętle i warunki): składnia instrukcji iteracyjnej (pętli) **while**, str. 77, instrukcja warunkowa **if ... elif ... else ...** str. 78, użycie znaku równości, str. 79. Umiejętności praktyczne: napisanie kodów realizujących sumowanie cyfr w liczbie (pętla **while** – kod ze str. 78), badanie, czy liczba jest dodatnia czy ujemna (instrukcja **if ...** kod ze str. 79)

Sprawdzian obejmuje wyłącznie materiał wymieniony powyżej. Składa się z części teoretycznej, mającej formę testu oraz części praktycznej, w której wymagane jest napisanie kodu (kodów) w języku Python. Wszystkie kody wymagane na sprawdzianie zostały wymienione powyżej - opisane jako „umiejętności praktyczne”.