

Практична робота №1

Основи роботи з даними у Python

Хід роботи

1. Відкрити у браузері будь-який онлайн сервіс для програмування на Пайтон
2. Набрати, проаналізувати код та запустити його на виконання:

Змінні у Python

У Python змінні визначаються простим присвоєнням значень. Не потрібно оголошувати тип змінної, оскільки Python є мовою з динамічною типізацією, де типи даних визначаються автоматично під час виконання програми.

Приклад коду:

```
x = 10 # Ціле число
y = "Hello, Python!" # Рядок
z = 4.6 # Дробове число
```

Правила найменування змінних

Змінні можуть починатися з літери або знаку підкреслення (`_`), але не з цифри, та можуть містити алфавітні символи, цифри та підкреслення.

Приклад коду:

```
my_variable = "Valid"
_my_variable = "Also valid"
1_variable = "Invalid and will cause a syntax error"
```

Динамічна типізація

Тип змінної в Python може змінюватися на льоту, що надає гнучкість, але також вимагає обережності, щоб уникнути помилок типізації.

Приклад коду:

```
x = 42
x = "Now I'm a string"
```

Функції перетворення типів

Python дозволяє явно перетворювати типи даних за допомогою вбудованих функцій.

Приклад коду:

```
number_string = "123"
number = int(number_string) # Конвертація рядка в ціле число
print(float(number)) # Конвертація цілого числа в дробове
```

Помилки при роботі зі змінними

Неправильне використання змінних може призвести до різних помилок, таких як `NameError`, `TypeError`, `IndexError`.

Приклад коду:

```
# NameError
print(unknown_var)

# TypeError
'6' + 7 # намагаємося додати рядок до числа

# IndexError
list = [1, 2, 3]
print(list[3]) # індекс за межами діапазону
```

Коментарі

Завжди важливо документувати код за допомогою коментарів для кращого розуміння та легшої підтримки.

Приклад коду:

```
# Це однорядковий коментар
```

```
print("Hello, Python!") # Це коментар в кінці рядка коду
```

```
'''
```

```
Це багаторядковий
```

```
коментар
```

```
'''
```

3. Прийміть подяку від викладача.

Висновки

Замість висновків, ви усі молодці!