

Основні поняття мови Python

Згадайте, з чого ви починали вивчення української та іноземної мови – з алфавіту. Потім ви вивчали правила запису слів – синтаксис мови, і значення нових слів – семантику мови.

Основними складовими будь-якої мови програмування є алфавіт, синтаксис і семантика.

Отже, і вивчення мови Python ми теж розпочнемо з алфавіту.

Алфавіт мови Python

У мові Python при створенні програм можуть використовуватися такі символи:

- літери латинського алфавіту A..Z, a..z; цифри 0..9;
- знаки арифметичних операцій, спеціальні символи: + – * / \ ^ = < > () . , : ; ' # _;
- комбінації символів: <=, >=, <>, =, !=, **;
- службові слова, що мають фіксований для Python зміст, наприклад: **and, elif, if, print, as, else, import** тощо.

Синтаксис мови – сукупність правил побудови команд мови програмування.

Семантика мови – сукупність правил виконання комп'ютером команд, записаних мовою програмування.

З синтаксисом та семантикою команд Python ви будете ознайомлюватися по мірі вивчення мови програмування.

Величини в мові Python

Окремий інформаційний об'єкт (число, символ, рядок та ін.) називають **величиною**.

Основними характеристиками величин є назва, вид, тип і значення

Вид величини визначає спосіб використання величини в програмі. Величина може бути константою (тобто постійною) або змінною.

Константи — це величини, значення яких не можуть змінюватися в ході виконання програми. Прикладом константи може бути число (5, 1.23) або рядок: "Це рядок!". *Змінні* — величини, значення яких можуть змінюватися в ході виконання програми.

Змінна – це просто іменована частина пам'яті твого комп'ютера, де ти тримаєш певну інформацію. На відміну від констант, потрібен якийсь спосіб для отримання доступу до змінних, і саме тому змінним дають імена (ідентифікатори).

Існують певні правила, яким потрібно слідувати при іменуванні змінних:

1) Першим символом імені має бути літера чи знак нижнього підкреслювання '_'.

2) Решта імені може складатися з літер, чисел або знаків нижнього підкреслювання. Не можна використовувати спеціальні символи, такі, як /, # або @.

3) Не можна використовувати пробіли, замість пробілу можна застосувати нижнє підкреслення.

3) Імена змінних чутливі до регістру символів. Наприклад, myname і myName – це різні змінні.

4) Не можна називати змінні іменами команд, наприклад, print.

Приклад 1. Правильними ідентифікаторами є i, __my_name, name_23, a1, b2.

Приклад неправильних імен: 2things, this is spaced out, my-name.

Коментарі в програмі

Коментар — це текст, призначений для читання людиною, а не комп'ютером. Коментар — це підказка для нас, яку дію виконує програма. Вставляючи коментарі в текст програми, ми спрощуємо собі та іншим її читання і розуміння.

Щоб комп'ютер відрізняв команди від коментарів, у мові Python перед текстом коментаря ставиться знак '#'. Редактор IDLE виділяє коментарі червоним кольором, нагадуючи про те, що Python проігнорує ці фрагменти коду.

Приклад 24.2. Коментар пояснює призначення наступної команди:

```
# Запит імені користувача  
s = input('Як тебе звати? ')
```

Типи величин в мові Python

Змінні зручно представити у вигляді "поштових скриньок" (комірок пам'яті комп'ютера), на які навішені ярлики з їх іменами. Для різних величин створюються «скриньки» різного розміру, який залежить від типу величини.

Тип величини визначається обсягом пам'яті, необхідним для її збереження, множиною припустимих значень величини, та операціями, які можна над нею виконувати.

Основними типами величин є числа і рядки.

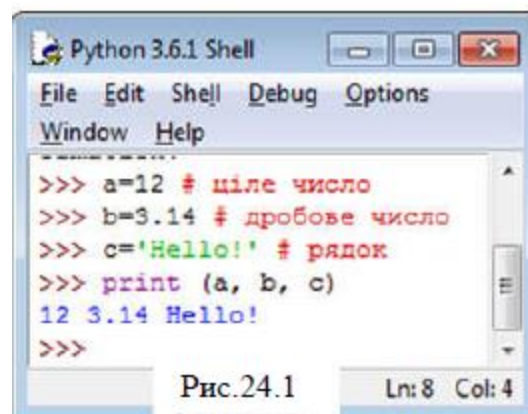
В Python є два типи числових даних: цілі числа (int), тобто числа без дробової частини, і дійсні (float) — дробові числа з десятковою крапкою.

Цілі числа потрібні для рахунку (перший, другий, третій...). Кількість учнів у класі, вік людини, кількість предметів ми зазвичай указуємо за допомогою цілих чисел. Числа з плаваючою крапкою, або десяткові дробі, потрібні, коли ми хочемо указати частину чого-небудь, наприклад, 3.5 м, 1.25 грн. Звісно, у програмі ми не будемо указувати одиниці вимірювання (метри, гривні), лише число з дробовою частиною.

В якості роздільника між цілою і дробовою частиною числа використовуйте крапку.

Рядок — це взята в одинарні лапки послідовність будь-яких символів — цифр, літер, розділових знаків. У змінних рядкового типу ми зберігатимемо фрагменти тексту.

Приклад 24.3. Типи величин в мові Python (рис.24.1).



Приклад 24.4. Щоб дізнатися тип величини, можна у вікні консолі виконати команду `type`.

Дізнаємось тип величин `a=24`, `b=3.14`, `c='a book'` (рис.24.2).

```
>>> type (24)
<class 'int'>
>>> type (3.14)
<class 'float'>
>>> type ('a book')
<class 'str'>
>>>
```

Рис.24.2

Оператори Python

Математичні символи, такі як `+` (плюс) і `-` (мінус), називаються операторами, так як вони оперують (або виконують обчислення) числами в рівнянні. У мові Python використовується більшість операторів, якими ви користуєтесь на уроках математики, в тому числі `+`, `-`, дужки

`()`. Однак деякі оператори відрізняються від використовуваних в школі: так, оператор множення представлений зірочкою `*`, а оператор ділення – косою рискою `/`.

Математичні оператори Python			
Операція	Символ оператора	Приклад	Результат
Додавання	<code>+</code>	<code>Res = 15+3</code>	<code>Res = 18</code>
Віднімання	<code>-</code>	<code>A = Res - 10</code>	<code>A = 8</code>
Множення	<code>*</code>	<code>A = A*2</code>	<code>A = 16</code>
Ділення	<code>/</code>	<code>Res = 5 / 2</code>	<code>Res = 2.5</code>
Обчислення неповної частки від ділення	<code>//</code>	<code>Res = 5 // 2</code>	<code>Res = 2</code>
Обчислення остачі	<code>%</code>	<code>Res = 5 % 2</code>	<code>Res = 1</code>
Піднесення до степеня	<code>**</code>	<code>A = 4**2</code>	<code>A = 16</code>

Приклад 24.5. Виконаємо у вікні консолі обчислення виразів:

```
>>> 5/2
2.5
>>> 5//2
2
>>> 5%2
1
>>> 5**2
25
```

Приклад 24.6. Дужки указують Python, яку частину виразу обчислювати в першу чергу, тобто регулюють порядок дій у виразі:

```
>>> 3*(6+2)
24
>>> 3*6+2
20
>>> 6/3+2
4.0
>>> 6/(3+2)
1.2
```

Випадкові числа

Випадкові числа часто застосовують у програмуванні при створенні ігрових або тестових програм тощо. Щоб отримати випадкове число, необхідно за допомогою команди `import` завантажити в Python функцію `randint`. Функція `randint(x1,x2)` вибирає ціле випадкове число в діапазоні від `x1` до `x2`.

Приклад 24.7. Отримання випадкового числа в діапазоні від 1 до 10.

Синтаксичні помилки

Якщо Python не може зрозуміти введену вами команду, то він може вивести у відповідь повідомлення про помилку з текстом `"SyntaxError"`. Це означає, що виникла проблема з тим, яким чином ви попросили комп'ютер виконати якесь завдання, тобто з синтаксисом.

Місце помилки у вікні консолі помічається червоним кольором. Після команди, що містить помилку, виводиться повідомлення про помилку. Уважно читайте це повідомлення – це допоможе зрозуміти, в чому помилка, і виправити її.

Приклад 24.8. У виразі допущено помилку: надруковано зайву дужку.

```
>>> (x+5) ]  
SyntaxError: invalid syntax
```

Питання для самоперевірки:

1. Назвіть основні характеристики величини.
2. Назвіть і охарактеризуйте види величин.
3. Які імена недопустимі в якості ідентифікаторів і чому?
а) *suma*; б) *w1*; в) *primer 1*; г) *(sum)*; д) *a-4*; е) *if*; ж) *8a*.
4. Дайте визначення типу величини.
5. Обчисліть значення виразів:
а) $7/2$ б) $7 // 2$ в) $7 \% 2$
г) $123 // 100$ д) $123 \% 10$ е) $(123 // 10) \% 10$
6. Як отримати випадкове число в діапазоні від 1 до 100?