

Лекція 2

Тема: Складові алгоритмічної мови. Елементи мови Python.



Лекція 2

Тема:Складові алгоритмічної мови. Елементи мови Python.

- 1) Склад алгоритмічної мови.
- 2) Алфавіт мови.
- 3) Лексеми.
- 4) Вирази, операнди, змінні і константи.
- 5) Типи даних.

1.Склад алгоритмічної мови

Будь-яка мова може бути визначена такими основними компонентами: символами, словами, словосполученнями та реченнями. В алгоритмічній мові їм відповідають: *символи, лексеми, вирази та оператори.*

Оператори мови складають *програму.*

Символи – це основні неподільні елементи мови. Представляють собою знаки, які складають усі конструкції мови.



2. Алфавіт мови.

Символи утворюють *алфавіт* мови. До алфавіту мови python входять символи, визначені в таблиці кодування ASCII, а саме:

1. Великі та малі латинські літери (розрізняються компілятором);
2. Цифри 0-9;
3. Знаки пунктуації;
4. Деякі інші символи.



Таблиця 2.1. Знаки пунктуації

Символ	Назва	Символ	Назва
,	кома)	права кругла дужка
.	крапка	(ліва кругла дужка
;	крапка з комою	{	права фігурна дужка
:	двокрапка	}	ліва фігурна дужка
?	знак питання	<	менше
'	апостроф	>	більше
!	знак оклику]	права квадратна дужка
	вертикальна риса	[ліва квадратна дужка
/	дробова риса (слеш)	#	решітка
\	зворотна риса (зворотний слеш)	%	процент
~	тільда	&	амперсанд
*	зірочка	^	виключне «або»
+	плюс	=	дорівнює
-	мінус	"	лапки
_	знак підкреслення		

3. Лексеми

Лексема – мінімальна одиниця мови, яка має власний зміст та обробляється транслятором. Лексеми формують вирази.

Програма розглядається як послідовність виразів і лексем.

Усі лексеми в Python поділяються на п'ять основних груп:

1. ідентифікатори та ключові слова (NAME);
2. літерали (NUMBER, STRING тощо);
3. символи операцій (OP);
4. роздільники (NEWLINE, INDENT, ENDMARKER ін.);
5. коментарі (COMMENT).

Lexeme?



Ідентифікатори – це просто імена (наприклад, змінні, функції або класи).

У Python ідентифікатори чутливі до регістру символів, повинні починатися з літери будь-якого алфавіту в Юнікодi або символу підкреслення (`_`), після чого можуть слідувати будь-які літери, цифри або символи підкреслення. При цьому з цифри імена не повинні починатися, інакше інтерпретатор не зможе відрізнити їх від чисел.

У той самий час комбінації символів `15var_1`, `$Var2` чи `var 3` неможливо використовувати як ідентифікаторів, так як вони починаються чи містять неприпустимі символи (третє ім'я містить прогалину). Також слід пам'ятати про регістр символів, що використовуються, оскільки, наприклад, ідентифікатори `var_1`, `Var_1` і `VAR_1` позначають імена абсолютно різних змінних або функцій, так як регістр символів вони збігається.

До роздільників в Python можна віднести:

- ❑ Відступи та крапка з комою в Python;
- ❑ Вкладені інструкції одного рівня мають оформлятися однаковими відступами від лівого краю. Саме за величиною відступів інтерпретатор визначає, де закінчується черговий блок інструкцій та починається новий;
- ❑ Як додаткові аргументи функції можна передавати іменовані рядкові аргументи **sep** і **end**.

Ще однією важливою частиною синтаксису будь-якої мови програмування є *коментарі*.

У Python використовуються лише однорядкові коментарі, які починаються із символу #.

Коментарі йдуть тільки до кінця рядка і весь код після символу # інтерпретатором ігнорується.

При цьому прийнято записувати коментарі на окремому рядку до інструкції, а не вставляти їх після того ж рядка).



2. Вирази, операнди, змінні і константи.

Вираз задає правило обчислення деякого значення і побудований на основі множини символів операцій, ключових слів та **операндів**. Порядок обчислення виразів визначається пріоритетами операцій, дужками і правилами асоціативності.

Операндами у виразі можуть бути як літерали (або літеральні константи), так і змінні.

Змінна – це іменована область пам'яті, до якої є доступ із програми.

Кожна змінна Python має певний тип, який характеризує розмір і розташування цієї області пам'яті, діапазон значень, які вона може зберігати, і набір операцій, застосовних до цієї змінної. Змінна, як і літерал, зберігає своє значення в деякій області пам'яті. Але, на відміну від літерала, до неї можна звернутися за адресою в пам'яті.

Оголошення змінних в програмі може знаходитись:

- на зовнішньому рівні, поза усіх функцій, це глобальні змінні;
- в середині функції – локальні змінні;
- у визначенні параметрів функції – формальні параметри.

Назва змінної в Python повинна починатися з алфавітного символу або символу підкреслення і може містити алфавітно-цифрові символи і знак підкреслення. Крім того, назва змінної не повинна співпадати з назвою ключових слів мови Python.

Ключові слова мови Python :

1	False	await	else	import	pass
2	None	break	except	in	raise
3	True	class	finally	is	return
4	and	continue	for	lambda	try
5	as	def	from	nonlocal	while
6	assert	del	global	not	with
7	async	elif	if	or	yield

У пайтоні застосовується два типи найменування змінних: **camel case** та **underscore notation**.

Camel case має на увазі, що кожне нове підслівне в найменуванні змінної починається з великої літери.

```
1 | userName = "Tom"
```

Underscore notation передбачає, що підслів в найменуванні змінної поділяються знаком підкреслення.

```
1 | user_name = "Tom"
```

Константа в Python – це особливий тип змінної, значення якої не можна змінити. У Python константи зазвичай оголошуються/визначаються в модулі (окремий файл, що містить змінні, функції та інше, який імпортується в основний файл).

```
1 # определяем константы
2 PI = 3.14
3 GRAVITY = 9.8
```

Літерали в Python – це уявлення фіксованих значень у програмі. Це можуть бути числа, символи, рядки тощо, наприклад, 'Hello, World!', 12, 23.0, 'C'.

4. Типи даних.

Змінна зберігає дані одного з типів даних. У Python існує безліч різних типів даних.

Базові типи:

- **bool;**
- **int;**
- **float;**
- **complex;**
- **str.**

Тип **bool** представляє два логічні значення: *True* (вірно, істина) або *False* (невірно, брехня). Значення True служить у тому, щоб показати, що щось істинно. Тоді як значення False, навпаки, показує, що щось хибне.

Тип **int** є цілим числом, наприклад, 1, 4, 8, 50.

За замовчуванням стандартні числа розцінюються як числа у десятковій системі. Але Python також підтримує числа у двійковій, вісімковій та шістнадцятковій системах.

Тип **float** є число з плаваючою точкою, наприклад, 1.2 або 34.76. У якості роздільника цілої та дробової частин використовується крапка.

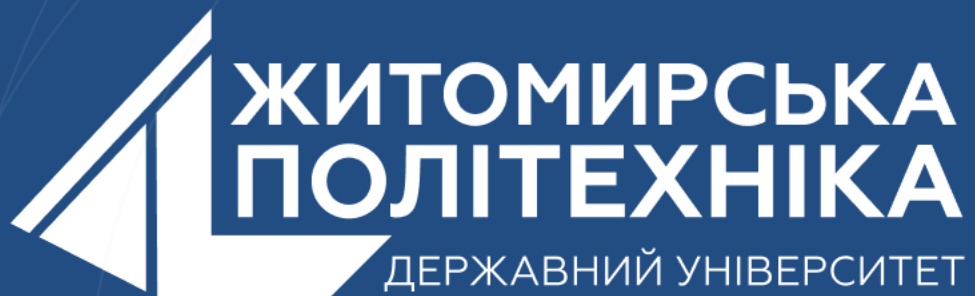
Число float може мати лише 18 значущих символів. Так, у цьому випадку використовуються лише два символи – 3.9. І якщо число занадто велике чи занадто мало, ми можемо записувати число у подібній нотації, використовуючи експоненту. Число після експоненти вказує ступінь числа 10, яке треба помножити основне число - 3.9.

Тип **complex** представляє комплексні числа у форматі `речова_частина+уявна_частинаj` - після уявної частини вказується суфікс `j`.

Тип **str** представляє рядки. Рядок представляє послідовність символів, укладену в одинарні або подвійні лапки, наприклад `"hello"` та `"hello"`. У Python 3.x рядки представляють набір символів кодування Unicode.



   @ZTUEDUUA



- Розвиваємо лідерів
- Створюємо інновації
- Змінюємо світ на краще

