Змінні та типи даних

№ уроку: 2 **Курс:** Python Starter

Засоби навчання: PyCharm

Огляд, мета та призначення уроку

Після завершення уроку учні матимуть уявлення про змінні та константи, різні типи даних, арифметичні та логічні операції, скорочену форму запису при виконанні арифметичних операцій, зможуть форматувати та виводити інформацію на екран.

Вивчивши матеріал даного заняття, учень зможе:

- Зберігати значення у змінних та працювати з ними
- Обчислювати у своїх програмах значення арифметичних та логічних виразів
- Використовувати форматоване виведення

Зміст уроку

1. Змінні та константи
2. Арифметичні операції
3. Скорочена форма запису арифметичних операцій під час роботи зі змінними
4. Операції порівняння
5. Логічні операції
6. Форматування та виведення рядків на екран
7. Рядки (str)
8. Байти (bytes і bytearray)
9. Основні методи при роботі з рядками

Резюме

Комп'ютери та його програми працюють з даними. Програма Instagram завантажує дані про нових підписників, комп'ютери Facebook обробляють запити користувачів і віддають потрібні дані у відповідь. Навіть цей документ містить дані, які комп'ютер обробив, щоб відобразити на екрані.

Дані – це важлива частина будь-якої програми. Їх можна зберігати в постійній пам'яті (ПЗП) або в оперативній (ОЗП). У постійній пам'яті зберігаються фотографії, музика, системні файли та інше. В оперативній пам'яті зберігаються дані, що програмам потрібні під час їх роботи: браузеру потрібно зберігати сторінку, а Skype зберігає повідомлення.

Як правило, постійної пам'яті набагато більше, ніж ОЗП. При цьому швидкість роботи ОЗП набагато вища за швидкість ПЗП. Саме через швидкість програм вигідно зберігати дані в ОЗП, щоб швидко при необхідності їх дістати, обробити та відобразити.

Зауважте, що, коли програма завершується, вона втрачає доступ до даних, які вона зберігала в ОЗП. Тому перед завершенням програми, важливі дані необхідно зберегти у файлі на ПЗП.

Змінна - це область пам'яті ОЗП, якій програміст дав ім'я. Надалі у цій області пам'яті можна зберігати значення. В одній програмі може бути десятки тисяч змінних для різних даних.

Типізація

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

У Python будь-які дані характеризуються своїми типами. **Тип даних** визначає множину допустимих значень, які може приймати змінна даного типу та множину допустимих операцій, які можуть застосовуватися до цього типу.

Для людини зрозуміло, що автомобіль призначений для їзди і на ньому не полетиш у космос (поки що).

Для комп'ютера не все так очевидно. Йому явно треба сказати, що це число, а значить з цим числом можна робити математичні операції. "А це рядок" – значить, можемо поєднати рядки, але ніяк не їхній квадрат.
Тобто, тип визначає можливі значення та їх сенс, операції, а також способи зберігання значень типу.

Операція призначення типу **називається типізацією**. Призначення та перевірка типів може здійснюватися заздалегідь (статична типізація), безпосередньо при використанні (динамічна типізація) або поєднувати обидва методи. Типи можуть призначатися "раз і назавжди" (сильна типізація) або дозволяти себе змінювати (слабка типізація).

Статична типізація — прийом, у якому змінна пов'язується з типом під час оголошення і тип може бути змінено пізніше. Приклади статично типізованих мов - C#, Java, C++, Pascal, Haskell, Scala.

Динамічна типізація — прийом, при якому змінна зв'язується з типом у момент присвоювання значення, а не в момент оголошення змінної. Таким чином, у різних ділянках програми одна і та ж змінна може набувати значення різних типів. Приклади мов із динамічною типізацією - Smalltalk, Python, Ruby, PHP, Perl, JavaScript, Lisp, Erlang.
За однією з класифікацій, мови програмування неформально діляться на сильно (строго) і слабо типізовані (англ. strongly and weakly typed), тобто такі, що мають сильну або слабку систему типів. Ці терміни не є однозначно трактованими, і найчастіше використовуються для вказівки на переваги та недоліки конкретної мови. Існують конкретніші поняття, які призводять до називання тих чи інших систем типів «сильними» чи «слабкими».
Система типів називається сильною, якщо вона унеможливлює виникнення непроконтрольованих помилок часу виконання, пов'язаних з узгодженням типів.
Приклади мов із сильною типізацією – Haskell, Scala, Java. Приклади мов із слабкою типізацією – C, JavaScript, PHP.

Python – мова із сильною динамічною типізацією.
Константа в програмуванні — спосіб адресації даних, зміна яких програмою (що розглядається) не передбачається або забороняється. Використання іменованих констант - прийом, що підвищує надійність і безпомилковість програм, дозволяючи уникати використання «магічних чисел».
Python не має окремого механізму для оголошення констант. Прийнято називати їх ідентифікаторами, написаними великими літерами (MY_CONSTANT), тоді як змінні Python оголошуються з іменами з малих літер (my_variable). Це не більше, ніж угода між програмістами.

Ідентифікатор – це ім'я, за яким можна звертатися до будь-якого об'єкта. Він може складатися з великих і маленьких букв, цифр, знаків підкреслення, не повинен починатися з цифри і не може збігатися із зарезервованими ключовими словами: False, class, finally, is, return, None, continue, for, lambda, try, True, def, from, nonlocal, while, and, del, global, not, with, as, elif, if, or, yield, assert, else, import, pass, break, except, in, raise.

Ключові слова – це програмні терміни, які використовуються у мові програмування для яких-небудь дій. Ключові слова чутливі до регістру і їх не можна використовувати як назви для змінних.

Зауважте, що Python чутливий до регістру: identifier та Identifier — це два різних імені.

Типи даних

CyberBionic systematics
CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- **None** – спеціальне значення типу NoneType, яке використовується, щоб вказати на відсутність значення.

Числа - базовий тип, що представляє собою незмінну послідовність цифр. Цьому типу притаманні всі математичні операції.

- **int** – тип даних, який використовується для цілих чисел. Приклади: 3, 0, -1, 256123462346523235252 (десяткова система числення), 0b11101 (двійкова система числення), 0o730 (вісімкова система числення), 0x7de (шістнадцяткова система числення).
- **bool** – логічні значення. Логічний тип використовується для перевірки істинності якоїсь умови. Якщо умова вірна, то повертається True, інакше — False.
- **float** – дійсні числа подвійної точності. Приклади: 0.3215235, 125256.73, -2.3e8 (експонентна форма запису).
- **complex** – комплексні числа. Приклади: 2+3j, 0.7 - 0.1j
- **str** – базовий тип, що представляє собою незмінну послідовність символів; str від "string" - "рядок".

Крім вищезгаданих типів даних є ще безліч інших типів даних, які будуть розглянуті в курсах Python Essential і Python Advanced, а також є можливість створювати свої типи даних. Будь-яка сутність у Python є об'єктом з певним типом.

Змінні в Python – це посилання, імена, які прив'язуються до об'єктів. Можна прив'язати кілька різних імен до одного об'єкта, і тоді при зміні об'єкта по одному імені всі ці зміни будуть доступні і по іншим.
Об'єкти бувають змінювані (mutable) і незмінювані (immutable). Об'єкти розглянутих вище типів є незмінюваними.
Операції з числами:

Код	Значення
x + y	сума
x – y	різниця
x * y	добуток
x / y	частка
x // y	операція цілочисельного ділення
x % y	остача від ділення
-x	число, протилежне x
+x	x
abs(x)	модуль числа x
int(x)	перетворити x в ціле число
float(x)	перетворити x в комплексне число
complex(re, im)	створити комплексне число re + im*i
c.conjugate()	число, спряжене комплексному числу c
x ** y, pow(x, y)	x в степені y
round(x), round(x, n)	округлити дійсне число x (до n цифр після коми, якщо n вказано)

Арифметичні операції мають пріоритети (як і в математиці). Щоб змінити порядок обчислень, слід використовувати круглі дужки.

Арифметичні операції, першим операндом яких є та ж змінна, що і результат, можна записувати в скороченій формі, наприклад, x += y замість x = x + y.
Для прискорення процесу розроблення під час роботи зі змінними використовують скорочену форму запису:

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

my_variable = my_variable + 4 эквивалентно: my_variable += 4
my_variable = my_variable - 4 эквивалентно: my_variable -= 4
my_variable = my_variable * 4 эквивалентно: my_variable *= 4
my_variable = my_variable / 4 эквивалентно: my_variable /= 4

Якщо імпортувати модуль math:
import math

то можна також використовувати ряд його функцій і констант. Серед них:

Код	Значення
math.trunc(x)	відкинути дробову частину дійсного числа x, повертає ціле число
math.floor(x)	найбільше ціле число (ціле в математичному сенсі, а не як тип даних), яке не перевищує дане дійсне число
math.ceil(x)	найменше ціле число, більше або дорівнює даному дійсному
math.pi, math.e	константи π, e
math.sin, math.cos и т.д.	математичні функції

Повний список математичних функцій можна переглянути в документації або набравши в консолі інтерпретатора:

import math
dir(math)

Логічні операції:

Код	Значення
x or y	якщо x – хибне, то y, інакше x
x and y	якщо x – хибне, то x, інакше y
not x	якщо x – хибне, то True, інакше False

Операції порівняння:

Код	Значення
x < y	x строго менше y
x > y	x строго більше y
x <= y	x менше або дорівнює y
x >= y	x більше або дорівнює y
x == y	x дорівнює y
x != y	x не дорівнює y
x is y	x та y – це один і той самий об’єкт
x is not y	x та y не є одним і тим самим об’єктом в пам’яті

Можна також використовувати подвійні порівняння, наприклад: -2 <= x < 3, a < b < c. Деякі операції з рядками:

Код	Значення
s1 + s2	конкатенація (об’єднання) рядків
s % x, s % (x1, x2, …, xn)	форматування рядка у стилі C, s – форматний рядок, x1…xn – значення, https://docs.python.org/3/library/stdtypes.html#printf-style-string-formatting
s.format(args)	форматування рядка у стилі C#, https://docs.python.org/3/library/stdtypes.html#str.format

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Для виводу значень на екран служить функція print:
print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)

де:

* objects - це об'єкти, які необхідно вивести (символ * в оголошенні функції означає, що при виклику функції всі вони передаються як окремі параметри функції, а інтерпретатор Python об'єднує їх у список і передає функції як один аргумент)
* sep - роздільник, який вставляється між виведенням окремих об'єктів (за замовчуванням пробіл)
* end – рядок, який необхідно вивести після всіх об'єктів (за замовчуванням – символ нового рядка)
* file – файл, до якого необхідно вивести дані. В рамках цього курсу ми не будемо використовувати цей параметр, оскільки файли розглядаються в курсі Python Essential
* flush – чи потрібно відразу після виведення скинути вміст буфера у файл. Якщо ви виводите інформацію на одному рядку через тривалі проміжки часу, і вона не з'являється на екрані, доки ви не виведете символ нового рядка, додайте параметр flush=True

Жоден із цих параметрів не є обов'язковим.
Для введення даних із клавіатури можна використовувати функцію input: input(prompt) або input()
Якщо параметр prompt заданий, то він виводиться, як пояснюючий текст, запрошення до введення.
Функція призупиняє виконання програми, доки користувач не введе рядок тексту, зчитує його та повертає.
Зверніть увагу, що вона повертає саме рядок, тобто значення типу str, тому якщо необхідно ввести число, потрібно скористатися однією з функцій, які були розглянуті вище, щоб сконструювати число з його текстового представлення.

Рядок (str) – це впорядкована незмінна послідовність символів Юнікоду.
Літерали рядків створюються з використанням лапок або апострофів, при цьому важливо, щоб з обох кінців літералу використовувалися лапки одного і того ж самого типу. Також можна використовувати рядки, які починаються та закінчуються трьома символами лапки.
Важливим для рядкового типу є поняття кодування символів, що, зокрема, впливає на правила порівняння рядків. Python за замовчуванням зберігає рядки в кодуванні UTF-8.

Послідовність	Значення
\\	Зворотній слеш ()
\'	Апостроф (')
"\""""	"Лапка ("")"
\n	Символ «Переведення рядка»
\t	Символ «Горизонтальна табуляція»
\a	Дзвінок
\b	Забій
\f	Переведення сторінки
\r	Повернення каретки
\v	Вертикальна табуляція

Якщо у рядку необхідно використовувати спеціальні символи (наприклад, перенесення або однойменні лапки), можна скористатися механізмом екранування символів, для чого використовується спеціальний символ \. У таблиці наведено деякі з екранованих послідовностей:

Спеціальні символи

Повторимо операції вище для рядків:
string = "Lorem ipsum dolor sit amet, consectetur adipisicing elit." # Ітерування
for character in string: print(character)

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

```
# Отримання доступу до елементів за допомогою цілочисельних ключів (індексація)
print(string[0])
print(string[2]) print(string[-1])

# Довжина послідовності print('Length:', len(string))
```

В Python рядок — це об'єкт, що складається з декількох елементів-символів.

"Сирі" рядки – пригнічують екранування

Якщо перед відкриваючою лапкою стоїть символ 'r' (у будь-якому регістрі), то механізм екранування відключається.

```
S = r'C:\newt.txt'
```

Але, попри значення, "сирий" рядок не може закінчуватися символом зворотного слеша. Шляхи вирішення:

```
S = r'\n\n\\'[:-1]
S = r'\n\n' + '\\'
S = '\\n\\n'
```

Закріплення матеріалу

- Що таке змінна?
- Що таке тип даних?
- У чому різниця між статичною та динамічною типізацією?
- У чому різниця між слабкою та сильною (строгою) типізацією?
- Охарактеризуйте типізацію у мові Python.
- Що таке ідентифікатор?
- Які основні числові типи даних є в Python?
- Що таке None?
- Які два способи форматування рядків є у Python?
- Які аргументи може приймати функція print?
- Яким буде результат виразу 3 != 4 and not ("test" != "test" or "Python" == "Python")?

Додаткове завдання

Завдання 1

Напишіть програму, яка запитує користувача радіус кола і виводить його площу. Формула площі кола: S= πr^2.

Завдання 2

Напишіть програму, використовуючи знання про змінні типу цілих чисел, щоб підрахувати, за який час транспортний засіб доїде з пункту A в пункт B, якщо відомі такі дані: відстань від A до B = 700 км, швидкість автомобіля буде постійною та дорівнює 90 км/год.

Використовуйте формулу: Час = відстань / швидкість. Створіть змінні length, куди запишіть відстань, velocity, куди запишіть швидкість та змінну time, куди запишіть результат. Після цього використайте метод print(), щоб вивести в консоль результат.

Завдання 3

Створіть змінну name, age, в якій зберігатиметься ім'я та вік користувача, введене з клавіатури. Виведіть у консоль “My name is <name> and I am <age>”, але зробіть рішення таким, щоб використовувалася конкатенація рядків (через +) та приведення типу числа до рядка (str(...)). Тобто, щоб якщо замінити змінні name і age, то в методі print() нічого не потрібно було би змінювати. Наприклад, a = 15, print(“Value =” + str(a)) і буде “Value = 15”.

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Самостійна діяльність учня

Завдання 1
Напишіть програму, яка запитує у користувача два слова і виводить їх розділеними комою.

Завдання 2
Напишіть програму, яка запитує три цілі числа a, b і x та друкує їх добуток.

Завдання 3
Напишіть програму, яка розв'язує квадратне рівняння $ax^2 + bx + c = 0$ за формулами $x_{1,2} = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$. Значення a, b та c вводяться з клавіатури. Для знаходження кореня використовуйте оператор зведення в ступінь, а не функцію math.sqrt, щоб отримати комплексні числа у випадку, якщо вираз під коренем негативний.

Завдання 4
Напишіть програму, в якій користувач вводить фразу з клавіатури, яка складається з 10 символів. На екрані виведіть суму ASCII-кодів символів введеного рядка.

Завдання 5
Напишіть програму, яка вводить з клавіатури текст і виводить його в оберненому порядку.

Рекомендовані ресурси

Документація з Python про стандартні типи даних та операції з ними
https://docs.python.org/3/reference/datamodel.html#the-standard-type-hierarchy
https://docs.python.org/3/library/stdtypes.html

Документація Python про операції над рядками та їх форматування
https://docs.python.org/3/library/stdtypes.html#text-sequence-type-str
https://docs.python.org/3/library/stdtypes.html#printf-style-string-formatting
https://docs.python.org/3/library/stdtypes.html#str.format
https://docs.python.org/3/library/string.html#formatstrings

Статті у Вікіпедії про ключові поняття, розглянуті на цьому уроці
https://ru.wikipedia.org/wiki/Переменная_(программирование)
https://ru.wikipedia.org/wiki/Константа_(программирование)
https://uk.wikipedia.org/wiki/Тип_даних
https://ru.wikipedia.org/wiki/Статическая_типизация
https://ru.wikipedia.org/wiki/Динамическая_типизация
https://ru.wikipedia.org/wiki/Сильная_и_слабая_типизация
https://uk.wikipedia.org/wiki/Цілі_числа
https://uk.wikipedia.org/wiki/Дійсне_число
https://uk.wikipedia.org/wiki/Комплексне_число
https://ru.wikipedia.org/wiki/Логический_тип

CyberBionic systematics
CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com