

**ІНФОРМАТИКА І ПРОГРАМУВАННЯ**



# ЧИСЛОВІ ТИПИ

Вбудовані числові типи:

`int` - цілі

`float` - дійсні (з плаваючою крапкою)

`complex` - комплексні

У модулях стандартної бібліотеки:

`Decimal` - з фіксованою точністю

`Fraction` - дроби

# Цілі числа

Літерали цілих чисел (тип `int`):

<code>1234</code>	(десяткові)
<code>0o177</code>	(вісімкові)
<code>0x9ff</code>	(шістнадцяткові)
<code>0b101010</code>	(двійкові)

Необмежена точність!

# Дійсні числа (числа з рухомою крапкою)

Літерали дійсних чисел (тип `float`) розпізнаються інтерпретатором за символом десяткової крапки і/або експоненти (символ `e` або `E`):

3.14

2. # те саме, що 2.0, але не 2

.2 # те саме, що 0.2

1.5e-6 # те саме, що 0.0000015

4E6 # те саме, що 4000000.0, але не 4000000

5.2e+6 # те саме, що 5200000

# Комплексні числа

Комплексні числа розпізнаються інтерпретатором за символом уявної одиниці ( $j$  або  $J$ ), який ставиться в кінці:

$3+2j$

$3.-2.5J$

$6j$

Їх можна також створити за допомогою функції `complex()`, аргументами якої є дійсна та уявна частини:

`complex(3, 2)`

Дійсна та уявна частини завжди мають тип `float`. Їх можна отримати через атрибути `real` та `imag`:

`(2+j).real`                    `# 2.0`

`c = 2+j`

`c.imag`                    `# 1.0`

# Математичні оператори

**Table 1-1:** Math Operators from Highest to Lowest Precedence

Operator	Operation	Example	Evaluates to...
**	Exponent	2 ** 3	8
%	Modulus/remainder	22 % 8	6
//	Integer division/floored quotient	22 // 8	2
/	Division	22 / 8	2.75
*	Multiplication	3 * 5	15
-	Subtraction	5 - 2	3
+	Addition	2 + 2	4

```
a = 123
b = 10
print(-a, a + b, a - b, a * b, a / b)
print(b ** a)
print(a // b, a % b)
```



Результатом ділення завжди буде дійсне число:

```
print(123/3)
```

Якщо хоча б одне з чисел у виразі дійсне, то результатом виразу також буде дійсне число:

```
print(123./10)
```

```
print(1.5 + 2.5)
```

Для зміни черговості операцій (групування підвиразів)  
використовуються круглі дужки:

$3*5**2$

$(3*5)**2$

# Функції конвертування типів `int()` і `float()`

Можуть приймати аргументом число або рядок:

```
int(3.95)      # 3 (відтинання дробової частини дійсного числа)
```

```
float(3)       # 3.0 (перетворення цілого числа в дійсне)
```

```
int('3')      # 3
```

```
float('3')    # 3.0
```

```
int('3.95')
```

```
ValueError: invalid literal for int() with base 10: '3.95'
```

```
float('3.95') # 3.95
```

```
float('3,95')
```

```
ValueError: could not convert string to float: '3,95'
```

# Функції переведення в інші системи числення

```
bin(111)          # '0b1101111' (у двійкову)  
oct(0b1101111)   # '0o157' (у вісімкову)  
hex(0o157)        # '0x6f' (у шістнадцяткову)
```

Зауважте, що результатом виконання цих функцій є рядок (тип `str`), а не число (тип `int`)

# Функція заокруглення round()

Приймає першим аргументом число, а другим (необов'язковим) — кількість десяткових знаків, до якої потрібно його заокруглити:

```
round(3.95)          # 4 (заокруглення до найближчого цілого)
round(3.95, 1)       # 4.0
round(1/3, 3)        # 0.333
```

## Інші вбудовані математичні функції

`abs(-2)`            # 2 (повертає модуль числа)

`pow(2, 5)`        # 32 (підносить 2 до 5-ї степені)

## Деякі методи числових типів

```
4.5.as_integer_ratio()      # (9, 2)
```

```
n = 2500
```

```
n.bit_length()             # 12
```

```
2+3j.conjugate()           # (2-3j)
```