

Python Number Methods Explained in Layman Terms

Python provides several built-in methods to work with numbers, including integers, floats, and complex numbers. Below is a clear explanation of each important number method with simple examples.

1. `abs(x)`

Returns the absolute (positive) value of the number.

Examples:

```
abs(-10)    # 10
abs(5)      # 5
abs(-3.7)   # 3.7
```

2. `round(x, n)`

Rounds the number `x` to `n` decimal places. If `n` is not given, it rounds to the nearest whole number.

Examples:

```
round(3.14159)    # 3
round(3.14159, 2) # 3.14
round(5.6789, 1)  # 5.7
```

3. `pow(x, y)`

Returns the result of `x` raised to the power `y` (x^y).

Examples:

```
pow(2, 3)    # 8
pow(5, 2)    # 25
pow(9, 0.5)  # 3.0 (square root)
```

4. divmod(a, b)

Returns a tuple (quotient, remainder) when a is divided by b.

Examples:

```
divmod(10, 3)    # (3, 1)
divmod(22, 5)    # (4, 2)
divmod(7, 7)     # (1, 0)
```

5. int(x)

Converts a number or string to an integer.

Examples:

```
int(4.7)         # 4
int("10")        # 10
int("100")       # 100
```

6. float(x)

Converts a number or string to a float (decimal number).

Examples:

```
float(10)        # 10.0
float("3.14")    # 3.14
float("7")       # 7.0
```

7. complex(x, y)

Creates a complex number $x + yj$.

Examples:

```
complex(2, 3)    # (2+3j)
complex(0, 4)    # 4j
complex(5)       # (5+0j)
```

8. bin(x)

Converts an integer to a binary string.

Examples:

```
bin(4)           # '0b100'
bin(10)          # '0b1010'
bin(0)           # '0b0'
```

9. oct(x)

Converts an integer to an octal string.

Examples:

```
oct(8)           # '0o10'
oct(64)          # '0o100'
oct(0)           # '0o0'
```

10. hex(x)

Converts an integer to a hexadecimal string.

Examples:

```
hex(255)         # '0xff'
hex(16)          # '0x10'
hex(0)           # '0x0'
```

11. isinstance(x, type)

Checks if `x` is of a specified number type (int, float, or complex).

Examples:

```
isinstance(10, int)          # True
isinstance(3.14, float)     # True
isinstance(2 + 3j, complex) # True
```

12. type(x)

Returns the type of the number.

Examples:

```
type(5)          # <class 'int'>
type(4.5)        # <class 'float'>
type(1 + 2j)     # <class 'complex'>
```

Summary Table

Method	Purpose
<code>abs(x)</code>	Absolute value
<code>round(x, n)</code>	Round to n decimal places
<code>pow(x, y)</code>	Raise x to power y
<code>divmod(a, b)</code>	Returns quotient and remainder
<code>int(x)</code>	Convert to integer
<code>float(x)</code>	Convert to float
<code>complex(x, y)</code>	Create a complex number
<code>bin(x)</code>	Convert to binary string
<code>oct(x)</code>	Convert to octal string
<code>hex(x)</code>	Convert to hexadecimal string
<code>isinstance(x, type)</code>	Check number type

Method	Purpose
<code>type(x)</code>	Get the type of the value

These number methods are very useful for basic operations, conversions, and type checks in Python programming.