Python Basics

Python Course · Module 2

ElnurBDa

Variables

A variable stores a **reference** to a value.

Created by assignment:

```
name = "Alice"
age = 20
height = 1.75
```

Rules:

- Start with a letter or _
- Letters, digits, underscores allowed
- Case-sensitive: count ≠ Count

Data Types

▒▒▒ int

Whole numbers, positive or negative.
Example:

```
age = 25
```

▒▒▒ float

Decimal numbers.

```
pi = 3.14
```

▒▒▒ str

Text data in quotes.

```
city = "Baku"
```

▒▒▒ bool

Logical values: True or False.

```
is_student = True
```

▒▒▒ NoneType

Represents “no value”.

```
result = None
```

Input & Output

▒▒▒ Output (stdout)

```
name = "ElnurBDa"
print("Hello")
print("User:", name)
print(f"The F String is used by {name}")
```

```
──────────────── [finished] ────────────────
```

```
Hello
User: ElnurBDa
The F String is used by ElnurBDa
```

▒▒▒ Input (stdin)

```
name = input("Enter your name: ")
```

`input()` **always** returns `str`.

Convert for numeric use:

```
age = int(input("Enter age: "))
```

Arithmetic Operators

Operator Summary

```
+   addition
-   subtraction
*   multiplication
/   division (float)
//  floor division (drops decimals)
%   remainder
**  exponent
```

Examples

```
10 + 3   # 13
10 - 3   # 7
10 * 3   # 30
10 / 3   # 3.333...
10 // 3  # 3
10 % 3   # 1
2 ** 3   # 8
```

Comparison Operators

```
x == y   # equal
x != y   # not equal
x > y
x < y
x >= y
x <= y
```

Example:

```
score = 85
score >= 90   # False
score < 100   # True
```

Logical Operators

```
and   # both conditions True
or    # at least one True
not   # reverses boolean
```

Example:

```
age = 20
has_id = True

age >= 18 and has_id   # True
age < 18 or has_id     # True
not has_id             # False
```

Assignment Operators

Shorthand updates:

```
x = 5
x += 2   # 7
x -= 1   # 6
x *= 3   # 18
x /= 2   # 9.0
```

Used often in loops and accumulations.

Type Casting (Conversion)

Convert between types:

```
int("10")      # 10
float("3.14")  # 3.14
str(123)       # "123"
bool("hi")     # True
bool("")       # False
```

Used for:

- numeric input
- formatting
- arithmetic

Example

```python
name = input("Enter your name: ")
age = int(input("Enter your age: "))

next_age = age + 1
is_adult = age >= 18
message = f"{name}, next year you will be {next_age}."

print(message)
print("Adult status:", is_adult)
```

Mini Task

Create a script that:

1. Asks for two numbers
2. Converts them to integers
3. Calculates:
   - sum
   - difference
   - product
4. Prints results