



Python Input: Getting User Data

In Python, the `input()` function allows you to take data from the user. This lets you create interactive programs that respond to user input.

Taking User Input

The Input Function

The basic syntax is `var = input("Prompt Message")`. This prompts the user with the message and stores the entered value in the variable `var`. The input is always read as a string, even if the user enters numbers.

Example

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

This code will prompt the user to enter their name. The input will be stored in the variable `name`.

string → integer → : Float

Converting Input Data Types

Integers

To convert a string input to an integer, use `int(var)`. For example, `age = int(input("Enter your age: "))`.

Floats

Use `float(var)` to convert a string to a decimal number. For example, `height = float(input("Enter your height: "))`.

Strings

Use `str(var)` to convert a value to a string. For example, `age = 25; age_str = str(age)`.

Evaluating Expressions with eval()

The eval() Function

The `eval()` function evaluates a string as a Python expression, allowing you to perform calculations directly from user input. It's useful for simple math operations.

Example

```
expr = input("Enter a math expression (e.g., 3 + 5): ")  
result = eval(expr)
```

This code takes a math expression from the user and then calculates the result using `eval()`.



Important Note: eval() Security

Be cautious when using `eval()`, as it can execute any valid Python code, potentially leading to security vulnerabilities. Only use it with trusted input or in controlled environments.