R Basics: Input and Output Functions

1. Input: Using `readline()` for User Input

In R, you can use the `readline()` function to get input from the user. This function prompts the user to enter a value, which is then read as a string.

Syntax:

variable <- readline(prompt = "Your prompt here: ")

- `prompt`: This is the message that will be displayed to the user, asking them for input.

Example:

Prompt the user to enter their name

name <- readline(prompt = "Enter your name: ")

Prompt the user to enter their age

age <- readline(prompt = "Enter your age: ")

Converting Input:

- Since `readline()` returns the input as a string, you may need to convert it to other data types if necessary (e.g., numeric, integer).

Example:

Convert the age input to numeric

```
age <- as.numeric(readline(prompt = "Enter your age: "))
```

Key Points:

- Always remember to convert the input to the appropriate data type if you're going to use it in calculations or comparisons.

- Be cautious about non-numeric input when converting using `as.numeric()` or `as.integer()`.

2. Output: Using `print()` and `cat()` for Displaying Values

In R, you can display output to the console using the `print()` and `cat()` functions. Both are used to show results or messages, but they have different use cases.

Using `print()`

- The `print()` function is the most basic way to output values in R. It automatically formats the output and is typically used for displaying variables, numbers, or any R objects.

Syntax:

```
print(value)
```

Example:

Print a simple message

print("Hello, World!")

Print a numeric value

x <- 42

print(x)

Print the result of an expression

sum <- 5 + 3

print(sum)

Key Points:

- `print()` is straightforward and automatically adds a newline after the output.

- It's ideal for quickly displaying the contents of variables or the results of calculations.

put formats.

- It’s useful for constructing sentences or combining text with variables.

- You need to manually include `\n` for newlines if you want the output to be on separate lines.

Example Combining Input and Output

Here’s a simple R script that uses both input and output functions:

```
 Prompt the user for their name and age
name <- readline(prompt = "Enter your name: ")
age <- as.numeric(readline(prompt = "Enter your age: "))

 Output a greeting message using print()
print(paste("Hello,", name))

 Output a message about the user's age using cat()
cat("You are", age, "years old.\n")
```

Explanation:

- The user is first asked to input their name and age.
- `print()` is used to greet the user by their name.
- `cat()` is used to display their age in a sentence, ensuring the output is formatted as desired.